Abstract

Using Sentence Context and Implicit Contrast to Learn Sensor-Grounded Meanings for Relational and Deictic Words: The TWIG System

Kevin Gold

2008

This thesis describes a novel system that allows a robot to infer the meanings of new words from their usage in context. TWIG (Transportable Word Intension Generator) can parse simple sentences, determine the reference of any unknown words to objects or people in the environment through sentence context, and can determine over time what the meanings of the new words are by building "definition trees" that imply the word meanings from their structure. The system was originally built to learn pronouns, a word category that has previously been unmodeled in the robotic word learning literature, but is general enough to learn some other word categories, including prepositions and transitive verbs. The system was implemented on a physical robot equipped with face detectors, simple vision systems, and a sensor network for object localization. TWIG succeeded in learning that "I" and "you" refer to the speaker and addressee; that "he" must refer to a person that is neither of these; that "this" and "that" must refer to proximal or distal non-person objects; that "above" and "below" are prepositions that refer to relative height; and that "am" and "are" refer to the identity relation. The system can be used for sentence production as well as comprehension, and was found to produce more correct sentences and fewer incorrect sentences about its environment than similar systems that lacked the system's extension inference and definition tree capabilities. The work contains several new approaches in the area of robotic word learning, and can also be interpreted as a computational model of how human infants use contrast to learn word meaning.

Using Sentence Context and Implicit Contrast to Learn Sensor-Grounded Meanings for Relational and Deictic Words: The TWIG System

A Dissertation
Presented to the Faculty of the Graduate School
of
Yale University
in Candidacy for the Degree of
Doctor of Philosophy

by
Kevin Gold

Dissertation Director: Brian Scassellati

May 2008

Contents

Acknowledgments

I would like to thank my advisor, Brian Scassellati, for allowing me to venture off and explore a topic entirely different from what we had originally thought I would do. He has been encouraging from beginning to end, and has taught me quite a bit about how to effectively communicate scientific ideas. This thesis has also benefited greatly from the insightful comments and questions of Drew McDermott and Dana Angluin, and I thank them for their guidance. I would like to thank Chen Yu for agreeing to be a reader as well; I am a big fan of Chen's research, and I'm pleased and flattered that he considers this thesis worth reading.

Almost none of the underlying robotic architecture of Nico was my doing, but was the result of the hard work of my advisor Brian Scassellati (design, code libraries), Matthew Herberg (vision system and libraries), Ganghua Sun (physical structure and design, motors, motor control, vision), Andrew Lovett (porter communication system, vision processing), Marek Doniec (vision, TCP/IP-based porter communication system, motor control), Chris Crick (Cricket sensor location system), and Justin Hart (vision, motor control). When it comes to the robot hardware and its interfaces, I stand on the shoulders of giants, and I am extraordinarily grateful for their contributions.

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 words 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 also supported in part by a grant of computer software from QNX Software Systems Ltd.

I would also like to thank Eli Kim, Marek Doniec, Chris Crick, Emily Dernier, Justin Hart, and Justin Kosslyn for their generously volunteering to help test the word learning system at various stages.

My heartfelt thanks go to Carie Cardamone for her invaluable support and assistance during the writing of this thesis – particularly during those last few hectic days! I would also like to thank Sarah Oelker for her support and encouragement over the years, as well as Fred Shic, Stephanie Tellex, Eli Kim, Max and Lynn Saltonstall, Julie Bowring, Mary Beth Willard, Genevieve Tauxe, Veronique Greenwood, and Alisa Beer. They've all helped make living in New Haven much more bearable ... even, dare I say, pleasant?

Thank you also to my parents and grandparents, who have always been very supportive of my pursuit of higher education.

Finally, I'd like to thank all the friends who have not received the attention from me that they deserve as I've pursued this research. Thanks for being there for me, everyone.

Chapter 1

Introduction

Learning a new language from observation alone is hard. Most conversations take place without helpful pointing gestures at what the speaker is talking about. Following a speaker's gaze, when it can be done at all, can be misleading or uninformative if the speaker is looking at the person he or she is addressing, or at something unrelated to the content of the speech. Even if the speaker were to point at, say, an empty chair, this hardly narrows down the range of semantic possibilities: the speaker could be saying, "Sit down," or "I got this at Target for ten bucks," or "We need one more for Bridge – care to join us?" A listener who knows no words of a language will usually be at a complete loss to decipher the utterances of adult speakers addressing each other.

On the other hand, a listener who knows some words of a language has a great advantage in learning new words, because that listener can infer some of the meaning of a new word from context. If the speaker says, "This chair is made of *mahogany*," an English-speaker might infer that mahogany is a kind of wood. If the speaker says, "The *Caliph* will sit there," the listener could wait to see what kind of person eventually sits in the chair. In both cases, the learner avoids the naive assumption that the new word is a name for the chair itself.

This thesis is about how a robot can take advantage of this kind of reasoning to learn new words from the people around it. Using the methods I will describe, a robot can watch two nearby people

have a simple conversation, pick out a sentence in which it understands all but one word, and then reason about what the new word means, based on its experience. The robot is then able to use the new word in understanding new sentences, generating its own sentences, and helping it to learn other words. To borrow an analogy from inductive proofs, my concern here is not with the "base case" of learning first words, but the "inductive step" of moving from a vocabulary of n words to a vocabulary of $n+1$ words. Previous robotic work has focused primarily on that first step of learning first words (see Chapter 2), and so has not explored the power of using existing word meanings and linguistic knowledge to learn new meanings. The work presented here shows how to expand a vocabulary, given a little knowledge to start.

My work on this problem began with a thought experiment. How would a robot learn the meanings of the words "I" and "you"? When I began this project, robotic word learning research had been mostly about finding repeated patterns of audio and matching those patterns to visual images (Roy and Pentland, 2002; Yu and Ballard, 2004). A robot using this kind of algorithm would fail to learn either "I" or "you," because "I" would always appear to refer to a human of some kind, and "you" would not look like anything at all if the speaker was looking at the robot and there was no mirror nearby. I was drawn to this puzzle, as well as the compelling idea of a robot learning to use "I" to refer to itself.

But the prize was more than just learning these two words. "I" and "you" seemed to point to a whole realm of language that was unavailable to the previous word learning systems. Even the youngest language learners often use words that would be inaccessible to a system that assumed all words can be defined by association with images: a child's first fifty words can include the words "more," "that," and "bye" (Nelson, 1973). By focusing on simple, universal words that were explicitly not visually defined, I hoped to uncover how other such words might be learned by a robot, and perhaps learn something of the mental scaffolding that underlies human language and thought.

I turned to the developmental psychology literature to find out what was known about how human children learned the words "I" and "you." The work of psychologist Yuriko Oshima-Takane suggested

that children do not learn these words through one-on-one teaching, but through observing other people talk to each other (Oshima-Takane et al., 1996, 1999; Oshima-Takane, 1988). This stands to reason; after all, if a child only learns words in one-on-one conversation, she will only ever hear the word "you" in reference to herself. How, then, would the child ever learn that "you" can refer to somebody else? The child would need to hear the word used in reference to somebody else, and that meant learning from conversations in which she was not taking part.

But learning words from speakers that do not intend to teach is hard – perhaps the hardest part about learning "I" and "you." I mentioned some of the reasons for this earlier: speakers cannot be expected to be helpful with their gaze and pointing, and they will generally speak in complete sentences. They will also tend to only use language correctly; there is no explicit "negative evidence" for the learner to decide when it would be incorrect to use a particular word.

Thus, by attacking the problem of "I" and "you," I had stumbled upon one of the great mysteries of child language acquisition: how young children manage to learn language even when they receive no explicit instruction. There are, in fact, some communities that consider teaching a child language as absurd a task as teaching the child to crawl; many adults in these communities do not even deign to speak to children directly until they already understand some language (Heath, 1983). Moreover, several researchers have argued that children do not pay attention to grammatical correction (Braine, 1971; Morgan et al., 1995), though this may not hold for semantic correction (Brown and Hanlon, 1970). In any case, it is certainly true that under many circumstances, children do not need to be told the meanings of words, but can infer meaning from context (Bloom, 2000; Brown, 1957)

A robot that could learn words passively, by observing humans talk to each other, would be considerably more useful than one that required explicit instruction. Humans have better things to do than teach a robot new words. Moreover, they may not know which words the robot does not already know, or may not even realize that a robot has the ability to learn new words at all. A robot that is able to learn without explicit instruction will be generally better able to adapt than one that does not have the same capacity. One resident of a working class mill town put it well when she told

a psychologist her philosophy on child word learning:

> He's got to learn to know about this world; there's no one who can tell him ... White folks hear their kids say something, they say it back to them, they ask them again and again about things, like they're supposed to be born knowing. You think I can tell [my grandson] all that he's got to know to get along? He's just got to be keen, keep his eyes open ... There's no use me telling him, "Learn this, learn that. What's this? What's that?" (qtd. in Heath, 1983)

Indeed, neither the robot designer nor the naive user can be expected to have the time or thoroughness to tell the robot "all he's got to know to get along." If the ability to learn new words at runtime is to be useful, it can't require a teacher's explicit instruction for every single new word. Not even human children receive such extensive instruction.

The method I eventually settled upon for determining the reference of new words was using sentence context. By parsing the understood words of a sentence, and grounding those words in the robot's environment, the robot could tell what the new word was referring to. For example, if the robot heard "*I* got the ball," and it knew what "got the ball" meant, it could look to see who was holding the ball, and assume "I" referred to that person. The details of this system are to be found in Chapter 6.

But this is only half the problem: finding the *extension* of the word, or what it refers to in a particular sentence. Over time, the system must learn the *intension* of a word, or what it means in a general sense without a particular referent in mind. In the case of nouns and pronouns, the intension of a word includes all the facts that the word implies about its referent: a "ball" must be round, "you" must be the person spoken to, and so on. For transitive verbs and prepositions, the intension is the information that the word implies about the relation between the two noun phrases it joins: for instance, "A chases B" implies A is behind B, and that both are moving quickly.

Ulimately, word meanings should be *grounded* in facts that the robot can infer from its sensors. This connection to the real world is what makes the robot more than a simple repository of facts it

has been told. It allows the robot to communicate what it senses using the words it has learned, and augment its sensory knowledge with factual knowledge. (For more arguments on the need to ground language and facts in experience, see Harnad (1990) and Roy (2005).)

There are several difficulties in learning the meaning of a word from examples in context alone. Sensor noise is a constant problem, rendering useless any technique that expects the data set to be fully accurate. Some words require that certain properties are *not* true of their referents; for example, "he" implies that the referent is not the speaker. Allowing such properties to be a part of a word's meaning can open up a Pandora's Box of possible meanings, because there are a huge number of facts that are not true at any given time, and some will consistently hold across all examples of a word. There are also issues of deciding how complex to make a definition, and how far it should generalize from only a few examples. Finally, there is the issue of *deixis*: some word meanings, such as "I," depend on who is speaking, and the robot must generalize to its own case using only examples from other people. I shall describe in Chapter 7 an elegant system that solves these problems, using a novel variant on decision trees in which the meanings are stored on the paths to the root.

The list of words that these methods have learned through observation is growing all the time, but has included the deictic pronouns "I," "you," "this," and "that"; the linking verbs "am" and "are"; and the prepositions "above" and "below." Each word is defined in terms of basic sensory predicates that the robot can verify for itself. Unlike previous word-learning systems, once the robot has learned a word, it can use the word in full sentences, even understanding sentences composed entirely of words that it did not know when the learning began. The robot can answer questions about its environment that contain the new words, or answer a question using a new word it has learned. In fact, the same methods that allow the robot to infer new extensions can also allow the robot to infer the referents of ambiguous pronouns such as "it," or to understand what is being asked in a question. In short, this thesis represents a significant step forward in increasing robots' ability to communicate.

Chapter 2 will review previous approaches to robotic word learning, thus giving the reader a

better idea of the aims, scope, and contributions of this thesis. Chapter 3 will review some of the psychological and linguistic principles that are embodied in the present work, including a brief overview of formal semantics and some relevant facts about childrens' word learning. Chapter 4 will discuss the sensory systems of the robot I used for my experiments. Chapter 5 will summarize the experiments I performed leading up to the present system, which were geared toward learning the words "I" and "you" as particularly tricky examples; this work previously appeared at the First Annual ACM Conference on Human-Robot Interaction (2006), the 5th IEEE International Conference on Development and Learning (2006), and the Sixth International Conference on Epigenetic Robotics (2006). Chapter 6 will describe the extension inference system that grew out of the earlier experiments, which was presented at the annual meeting of the Association for the Advancement of Artificial Intelligence (AAAI) in 2007. Chapter 7 will present my "Definition Trees" method for learning word intensions, which first appeared at the 2007 International Conference on Development and Learning. Chapter 8 will summarize the contributions of the system and its implications for other work.

Chapter 2

A Review of Robotic Word Learning

2.1 Overview

> We may hope that machines will eventually compete with men in all purely intellectual fields. But which are the best ones to start with? Even this is a difficult decision. Many people think that a very abstract activity like the playing of chess would be best. It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English. This process could follow the normal teaching of a child. Things would be pointed out and named, etc. Again I do not know what the right answer is, but I think both approaches should be tried. (Turing, 1950)

As this quote from Turing's "Computing Machinery and Intelligence" shows, the idea of teaching a robot words for the things it sees has been around even longer than the field of artificial intelligence has had a name. (The Dartmouth Artificial Intelligence Conference would would not occur until 1956.) Turing thought that by providing a robot with a sensory system and a capacity for language, a robot might then be able to obtain an education in a manner similar to a human child. Like much else in this famous essay that introduced Turing's imitation game, the idea was far ahead of its time

in terms of the available technology. Nevertheless, Turing recognized that if the intelligence's pursuits were not to be limited to "a very abstract activity like the playing of chess," it would probably need some sensory connection to the real world, as well as a means by which humans could communicate facts about that world to the robot.

Turing made his observations so briefly that he is not usually given credit for his argument that robots should be taught English using "the best sense organs money can buy." Rather, the more common citation in recent years is Stevan Harnad's "The Symbol Grounding Problem." Harnad likened the state of most artificial intelligence systems to somebody trying to learn Chinese from a Chinese/Chinese dictionary. In most reasoning systems, "meanings" for symbols merely pointed to other symbols, with nothing necessarily anchoring the whole system in reality. Harnad asked, "How can you ever get off the symbol/symbol merry-go-round?" He concluded that meaning had to be built "bottom up," from sensors, or else the meanings of the symbols would be simply "parasitic on the meanings in the head of the interpreter" (Harnad, 1990).

Harnad's analysis was probably influenced by Searle's Chinese Room argument. Searle likened artificial intelligence programs to a person locked in a room, forced to respond to Chinese phrases by looking up the replies in a giant book of protocol. Artificial intelligences cannot be said to understand their content any more than the person locked in the room can be said to understand Chinese, said Searle (1980a). Arguably, if the locked room aspect were taken away, and the Chinese characters presented along with the real situations that engendered them, then the person locked in the Chinese room could be able to "understand Chinese" after all. This is the heart of the "robot reply" to the Chinese room argument: that giving the artificial intelligence sensors, or having the symbols be caused by the external environment "in the right way" could allow the intelligence to understand its symbols after all (Fodor, 1980). This answer doesn't satisfy Searle, who argues that this semantics isn't the kind of semantics he's looking for; he wants the understanding to be grounded in subjective experience, or "qualia," which he believes can only be generated by brains (Searle, 1980b). Nevertheless, the Chinese Room analogy is interesting in that it can be seen as an

argument against artificial intelligence from semantics, and therefore suggests the importance of the "epistemology" of an A.I.'s semantics – i.e., how it came to know the meanings of its words and symbols.

Curiously, however, the recent work that has attempted to ground words in perception has come mostly from the cognitive science community, with an emphasis on cognitive modeling over artificial intelligence; this includes the work of Regier (1996), Bailey (1997), Yu and Ballard (2004), de Marcken (1996), and Roy and Pentland (2002). The lack of roboticists working on the problem might be the legacy of Rodney Brooks, who argued forcefully that the robotics community ought to give up implementing logic and reasoning entirely, and focus instead on low-level reflexes and instincts (Brooks, 1990). It may also be attributable to the focus of the natural language processing community on text in the absence of perception, since this tends to be where most practical applications lie (e.g., web crawling, document summarization, and automated telephony). Furthermore, roboticists interested in language acquisition have tended to focus on the most obvious words: concrete nouns. This makes the problem seem to be unsolvable without good computer vision, which itself is a subcommunity with plenty of open problems.

Below, I shall briefly summarize the most influential pieces of robotic language acquisition research leading to the present work. Most of them have appeared within roughly the last ten years, and most of them were framed in some way as models for child language acquisition. My own motivation is somewhat more complicated; I believe that human word learning is worth trying to "reverse engineer," but that clarity, versatility, logical coherence, and accuracy are more interesting goals than biological and psychological plausibility, which are often in the eye of the beholder. Nevertheless, because they learn sensor-grounded meanings for words in an unsupervised fashion, the systems I will describe below are the most similar to the present work in their goals and methods.

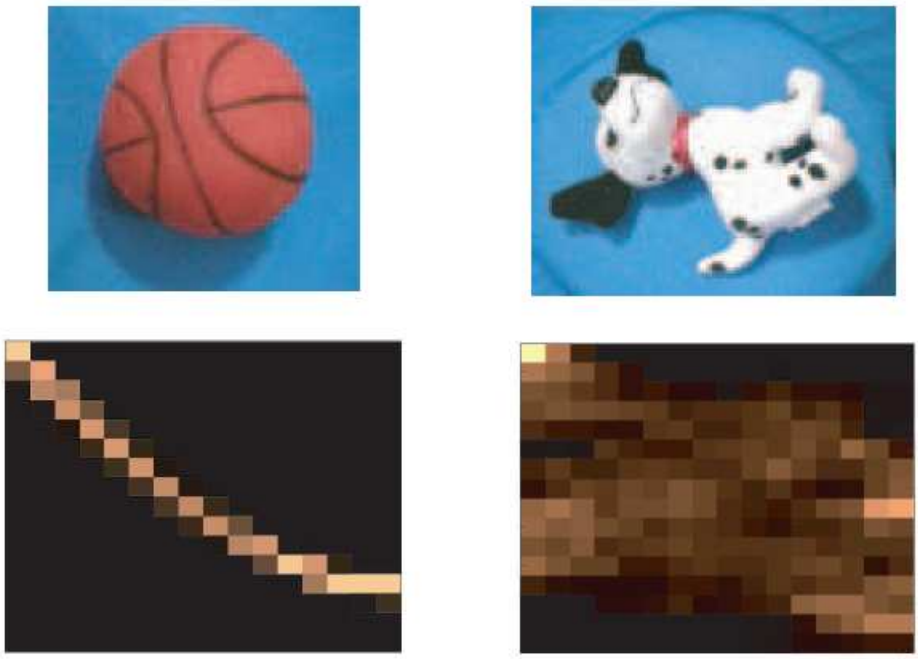

Figure 2.1: Sample visual inputs to Roy and Pentland's CELL system (top), and the histogram of pairwise distances between points on the objects' perimeters (bottom), which CELL used to visually characterize them (Roy and Pentland, 2002).

2.2 The CELL System

Deb Roy's CELL system (Roy and Pentland, 2002) was one of the first systems to use real auditory input with real images for word discovery and learning. Though there were word learning systems that came before it, no previous system combined word discovery from real infant-directed audio with vision.

The inputs to the CELL system were audio recordings of real mothers talking to their children about various toys, combined with static images of those toys against plain backgrounds (Figure 2.1). The desired outputs were phonetic transcriptions of words for toys, paired with histograms of pairwise point distances that represented the shape of each toy.

To do this, CELL passed the audio information to a recurrent neural network that had been trained to output a vector of phoneme probabilities for each 10ms time window. The visual information was changed into a representation of object shape by subtracting out the plain background

from the image, then calculating the relative distance between every pair of points, as well as the relative angle between the line connecting the points and the object edge at one of the points. This information was then stored in a histogram for each view of each object; histograms for multiple views were stored in lieu of a true 3D representation.

Roy's algorithm handled the noisy, variable nature of his data by defining distance metrics between audio segments, and between video segments. For the audio, this was done by creating a Hidden Markov Model for each audio segment, in which the production probabilities were the probabilities output by the RNN and the transition probabilities were trained from the TIMIT audio database (Garofalo, 1988); the distance between audio segments was then a function of the probabilities that each segment's HMM generated the observed output of the other. For the shape histograms, a simple chi-square statistic on the four most-similar views for two objects served as the distance metric. The audio distances were used to find repeated words in the system's "short term memory," which qualified an audio-visual pair for entry into "long term memory." Then both kinds of distances were used to consolidate similar audio-visual pairs in long-term memory.

The CELL system's greatest strength was its clever distance metrics, which allowed it to compare messy real data for word discovery. Going from idealized representations to real input is a huge step, and the distance metrics were the key to making this happen. On the other hand, CELL was still a long way from learning language in a real environment. Assuming that all words refer to the shape of an object, and that the object in question is visible and obvious, is a long way from being able to learn language in general. It isn't clear, for example, how the distance metric idea would work in a domain where shape is sometimes irrelevant; objects may be quite distant along some dimension, while still close in the dimension relevant to the word at hand. (Consider being presented with Big Bird and a sunflower as examples of the word "yellow." What should the distance between their visual representations be?) Also, the semantics of the system were entirely based on recognition, rather than production, and it is difficult to see how shape histograms could be integrated into a full sentence compositional semantics.

2.3 Yu's Gaze-Tracking Word Learner

Real word learning does not occur in a blue-background vacuum, and Chen Yu's word learning system (Yu and Ballard, 2004) used gaze direction to determine what the speaker was talking about. The system used a head-mounted eyetracking system to determine the speaker's head pose and gaze fixation points; a Hidden Markov Model with these inputs decided whether the observer was fixating or in the midst of a saccade to a new target. An image segmentation algorithm (Wang and Siskind, 2003) was combined with a color similarity measure to extract the observed object from the rest of the scene.

The visual features used in Yu's word learner were considerably more expressive than the CELL system's simple shape histograms. Color, shape, and texture were extracted from the objects observed. In addition, for hand movements of the speaker, the system extracted three-dimensional translation and rotation speeds.

Similarity between words was determined using "edit distance" calculations in which the distance between phonemes was based on the similarity of their articulatory features. Rather than requiring that words be repeated within a short amount of time, the system put all utterance fragments into bins based on their visual context, and searched for similarity only within the same bin. (Unlike CELL, Yu's system does not appear to have needed a short-term memory buffer, probably because the phoneme representation was more efficient than storing Roy's probability vectors at 100 Hz.) Each bin was the result of a clustering algorithm applied to the visual data; these clusters were considered the possible meanings of the words, and each word could be binned with as many "meanings" as applicable at the time it was uttered. A final step performed expectation maximization to calculate the most likely conditional probabilities of "Pr(meaning | word)" for each word.

Yu's system included several advances over Roy's: a wider variety of visual features, a method for determining the referent of a word from a complicated visual scene, a reduced reliance on repetition within a short time frame, and better accuracy overall. The ability to pick out referents through gaze tracking was an especially clever addition.

While Roy's method provided a distance metric such that objects within a certain perceptual distance of each other could be judged the same, Yu's method clustered the existing examples together. This unfortunately meant that the system's meanings lacked an easily expressible, compact form that could be used in a compositional semantics. Another possible criticism is that the visual prototypes themselves were difficult to interpret, being the result of clustering applied to principal component analysis applied to large-dimensional spaces. It is unclear how well the clusters fit the "necessarily true" aspects of the words in question, as opposed to facts incidental to the experimental setup.

Yu's recent work (Yu, 2006) has shown that information about the "syntactic class" of a word, corresponding to the kinds of words that have been observed to be substituted for that word, can help in mapping words to objects. The work represents an interesting bridge between automated grammar learning (the system uses the ADIOS grammar learner (Solan et al., 2005)) and semantics learning. Yu's work along these lines was tested only in simulation, with the environmental context represented as unordered lists of abstract symbols (similar to de Marcken (1996); see below), and thus could not capture the relational information, compositional semantics, or conjunctions and numerical values used by TWIG; on the other hand, this system learned some grammatical information online, which TWIG does not do.

2.4 Regier's Preposition Learning

We turn now from methods implemented on real robots to cognitive models implemented in simulation, one of the most influential of which was Terry Regier's preposition-learning system (Regier, 1996). Regier's interest was in how word meanings can be learned from only "positive examples," when so many neural networks require "negative feedback" for learning categories. (Children are thought to learn language primarily from correct examples; see chapter 3 for more details and citations.) His answer was essentially that situations labeled with other words count as implicit, "weak"

negative examples for the target word. For example, if a situation was labeled as an example of "near," it would also update the representation for "outside" as a weak negative example. This translated into mathematical terms as a reduction in the step-size Δ during back-propagation of error through the neural network.

Though Regier's work was done in simulation, it was a much better simulation than most. Images were processed with artificial neurons performing the processing of center-surround and more complex cells, modeling actual human visual processing. Visually, then, it was quite close to the goal of grounding language in perception, at least for the case of prepositions when the "trajector" and "landmark" are known. (In "A is near B," A is the trajector and B is the landmark.) Still, the shapes involved were largely simple line drawings, making Yu's and Roy's systems slightly more convincing demonstrations of practical visual grounding.

The model was also notable for modeling the prepositions of several different languages, including Russian, German, and Mixtec, which can contain very different prepositions. For example, German subdivides the case handled by the English "on" depending on whether the landmark supports the trajector (*auf*) or not (*an*). Mixtec, meanwhile, makes analogies with body parts, such as *sini* meaning "on the (animal) back of" to describe an object on a table.

If insisting on neurally inspired computation was a strength of Regier's system, it probably also limited the model to learning words in isolation; each scene was presented while activating a single "neuron" representing the target word. Parsing and compositionality would have been difficult to add to the system while retaining a connectionist implementation, but were tangential to Regier's primary interests.

2.5 Bailey's Verb Learning

A related system that emerged from the same group as Regier was Bailey's x-schema system for learning verbs (Bailey, 1997). Bailey's system was also connectionist, with the underlying repre-

sentations called Petri nets (Murata, 1989) controlling the flow of actions. Action parameters were characterized by discrete values; for instance, objects being grasped were "small" or "large." Unlike Regier's system, Bailey's system did not receive input from sensors or simulations thereof; the system was given only sets of these discrete classifications, matched with verbal labels.

Bailey's system was primarily of interest in the way it built up verb definitions via Bayesian model merging. Each action given as an example of a verb began as a separate sense for the word, but models with many senses had small prior probabilities due to their complexity. The system then iteratively combined senses as long as this increased the posterior probability of the overall model. A single sense could contain a probability distribution over the attributes of the action – so that, for instance, "push" might refer to an action performed on a small or large object with equal probability ($p = 0.5$), but might strongly prefer an open palm ($p = 0.9$) to a fist ($p = 0.1$). In the end, different senses for the same word ideally only remained if the senses were quite different – for example, the "push" that moves an object across the floor versus the "push" that depresses a button.

In the end, the Petri net formalism did not seem to matter much to Bailey's system; its inclusion seems to have been largely cosmetic, to appease connectionists (or perhaps his advisor). The use of discrete classifications for all actions is somewhat problematic, however, since real sensors would not provide such convenient classifications with certainty. One can imagine a system that uses more sophisticated Bayesian mathematics to deal with continuous distributions instead – but there are so many details missing, such as what underlying sensory details to use, whether or how to classify them into discrete parameters, and how to deal with sparse data and noise, that Bailey can't really be said to have built a system that would work with actual sensors. Nevertheless, the use of Bayesian model merging for building senses is interesting, particularly as it is similar to the expectation maximization method de Marcken used to discover words (de Marcken, 1996).

2.6 Siskind's Simulations and Logical Inference

Siskind's model of word learning in the presence of noise (Siskind, 1994) was performed entirely in simulation, and is a good example of the way simulations can be highly successful on their own terms by abstracting away some of the problems facing a real-world word learner.

Siskind's model of word learning treated each utterance as an unordered set of words $w_1, \ldots, w_n$ that had already been matched to a set of possible utterance meanings M. These possible meanings were themselves s-expression-like representations; Siskind gives "CAUSE(mother, GO(ball,UP))" as an example, though a more complete description of the space of possible meanings was not given. This did not matter much, however, because the system immediately decomposed these expressions into sets of the symbols that compose them (e.g., {CAUSE, GO, UP, ball, mother}). The system's task was to provide a set of senses for each word, such that these possible meanings were consistent with the input.

The basic algorithm consisted of adding to the set of "possible" meanings for each word and reducing the set of "necessary" meanings until the two were equal. An extension addressed the problems of noise and homonymy (multiple word meanings) by adding meanings to words if no one-to-one mapping was logically possible, then deleting extra meanings that were encountered infrequently.

A little thought about Siskind's algorithm reveals that it would probably not function very well in a real environment, because some meanings are logically *possible* for every single utterance – for example, "I exist," or "Pay attention," or "What I say is true." Without any notion of word or property frequency, or informativeness, only very artificial environments will have a uniquely consistent word-to-meaning mapping. In fact, Siskind only tested this algorithm under highly artificial circumstances: 80% of the time, utterances were presented with their correct meanings and nine randomly generated meanings, while the other 20% of the time all 10 possible meanings were randomly generated. With 1,680 word senses alone, the number of possible random utterance meanings was so high that a collision between possible meanings across utterances was highly unlikely.

It is also difficult to see how Siskind's algorithm would function with real data, instead of symbols

alone. Using numbers requires that one specify how exactly a pattern of numbers should generalize. There are literally an infinite number of ways of doing this in a matter consistent with data, making logical consistency alone an insufficient criterion for learning meaning.

Siskind's later work has dealt with ways of modeling verbs in a form appropriate to learning their definitions – for instance, extracting support, contact, and attachment relations from line drawings (Siskind, 1995) and later from video (Fern et al., 2002). The former did not actually involve learning or language at all, though it is often cited in this area, perhaps because of its overly promising title ("Grounding Language in Perception"). The latter did involve learning positive examples of actions from video, but no words were actually involved. Perhaps these are milestones on the way to eventually circling back to the problem of word learning itself, but the difficult problem of changing the visual scene into a logical formalism appears to have sidetracked the linguistic aspects of Siskind's work.

2.7 de Marcken: Learning as Compression

Carl de Marcken's Ph.D. thesis (de Marcken, 1996) is a good representative example of a system built to segment text or phonetic transcriptions into words and phrases by finding repeated patterns. Several other systems have used similar lexicon-building methods (e.g., Brent and Cartwright, 1996), but de Marcken's example also attempts to handle the case of "extralinguistic channels," or in other words, learning meanings for words at the same time as their segmentations.

de Marcken's strategy rested on the "Minimum Description Length" (MDL) principle, or the idea that learning is a kind of compression of data (Grünwald, 2005; Rissanen, 1972). Essentially, if one can represent the input concisely, then one has probably learned something about its structure. Seeking concise representations of the input encourages the use of concise hypotheses that explain the data; if a good explanation of the data can be found, it should result in an overall savings in the number of bits necessary to represent the input. The desirability of short hypotheses is a

principle most famously advocated by the medieval theologian William of Occam, and has long been used as a heuristic for judging scientific theories; the heuristic "is often referred to as *Occam's Razor* to indicate that overly complex scientific theories should be subjected to a simplifying knife" (Kearns and Vazirani, 1994). More recently, it has been proven that "Occam learning," or learning that guarantees a hypothesis that can be represented by a small number of bits, results in "probably approximately correct learning," or learning that has an arbitrarily high probability 1 - δ of arbitrarily low error ϵ, given a sufficient (polynomial in $1/\delta$ and $1/\epsilon$) number of examples (Blumer et al., 1989).

Given an input corpus of raw text (or, equivalently, a phonetic transcription), de Marcken's algorithm made many passes over it, replacing repeated patterns with shorthand symbols which were explained by a small lexicon. This transformation was only performed if the combined size of new, compressed text and lexicon was smaller than the original size. A basic version of the algorithm would only search for the lexicon entry that provided the greatest savings, but de Marcken included methods for adding several lexicon entries at a time, so long as they did not conflict. In addition, the lexicon could itself be compressed by referring to other symbols in the lexicon; in this way, the lexicon contained not only words, but larger phrase structures and smaller morphemes. When the text and lexicon could no longer be compressed, the final lexicon contained a fairly good representation of the kinds of words, phrases, and morphemes that could carry individual meaning.

Obviously, some words do not simply concatenate with their morphemes to create new words; sometimes consonants must be doubled when adding "ing," for instance. de Marcken allowed for this possibility by also including "perturbations" in the search. Thus, a new lexicon entry could contain not only the parts that composed it, but also rules that changed the parts' structure when they were combined.

This method of pertubations led to de Marcken's representation of meaning in the lexicon: a meaning was simply a perturbation that added a symbol for a "thing out there" to the lexicon entry. For example, if the text "walk" was encountered along with the "extralinguistic" symbol WALK, then the lexicon might add the entry "ω = w+a+l+k + $\{WALK\}$," associating the letters with the

meaning and a shorthand ω with which to represent both. de Marcken argued that this would allow entries such as "γ = c+r+a+n+ β + $\{RED\}$ and β = b+e+r+r+y + $\{BERRY\}$ for "cranberry" and "berry," respectively, which would allow "cranberry" to make use of the existing meaning for "berry" while tacking on the fact that it was red.

Though de Marcken's algorithm was remarkably successful on noisy phonetic transcriptions of audio when it came to finding meaning, his semantics was highly suspect: the meanings provided as "extralinguistic" information were actually the spoken sentences themselves! These sentences were treated as sets of symbols, with one symbol for each word; thus, a successful "meaning" paired the correct string of phonemes from the auditory channel with the appropriate word-symbol. Though this method was certainly easier to implement and evaluate for accuracy than a more realistic method, it made for a strange model of the world. Though lexicon entries were allowed to be compositional, the facts about the world were not; concepts such as WALK and ON floated out in space, unbound to any particular entity.

de Marcken's method of perturbation operators for meanings in the lexicon may yet prove to be useful, and his algorithm was highly successful at word discovery. Nevertheless, his representation of the world as a "bag of symbols" could have been helped tremendously by the introduction of something like Montague semantics (see Chapter 3). One also suspects that there must be a better way to compress the data than de Marcken's expectation maximization approach, which is not well-suited to online learning and seems a little too much like a brute force search.

2.8 Other Related Work

The projects mentioned above are the most similar to the current project in their aims and scope, but a few other projects similar to these deserve mention.

The earliest system that explicitly could learn new words was Terry Winograd's SHRDLU system (Winograd, 1971). The system could accept new definitions explicitly, such as "a 'steeple' is a stack

which contains two green cubes and a pyramid." Though SHRDLU predates the widespread use of Montague-style semantics in NLP, it performed a similar transformation of sentences into logical form. Quotes typed around a word indicated that it was new, and that a definition was being provided.

Tim Oates' PERUSE system (Oates, 2002) used "dynamic time-warping" to find repeated words in raw audio, without the use of a phoneme classifier. Dynamic time-warping is a dynamic programming approach to "stretching" segments of audio over time until they resemble each other; the amount of stretch necessary determines the goodness of the match. PERUSE was primarily a method for word discovery, and did not particularly otherwise interact with the environment (visually or otherwise).

Tim Oates also developed a method for inferring the rules of a context-free grammar from examples labeled with semantic types (Oates et al., 2004). Grammatical inference is a subfield unto itself, only tangential to the current work, but this is one of the few papers that combines learning with Montague semantics, which we shall discuss later as a very useful framework for word learning (see Chapter 3). Oates, in turn, cites Tellier (1998) for having used Montague semantics in her grammatical inference work; oddly, though, nobody appears to have used Montague's framework for learning the meanings of words.

Brent proposed an algorithm similar to de Marcken's for word discovery and segmentation, called INCDROP (Brent, 1999). It had the advantage of being an online algorithm, but it did not handle the problem of association with meaning, concentrating instead on word segmentation and discovery.

Latent Semantic Analysis (LSA) is a method that supposedly learns the meanings of words by processing large amounts of text and associating nearby words with each other (Landauer and Dumais, 1997). While the method does provide a metric of similarity between words (it was tested by asking it to choose synonyms), to describe what LSA produces as "meanings" might be a bit strong, given that the system cannot judge the truth of statements composed of the words it supposedly understands. "Associations" might be a better term for what LSA produces, since it can represent

what things are related to each other, but not the ways in which they are related, nor any factual information about the concepts in themselves.

The ADIOS system was another system that attempted to learn language through statistical means (Solan et al., 2005). Its strategy was to find common phrase structures, such as "X wants a Y," and then cluster words based on which other words that appeared to be interchangeable with them in the context of these phrases. ADIOS was more interested in grammar and segmentation than semantics, but it did cluster similar words while also discovering what phrases they could be used in. In that way, ADIOS did what LSA did, only better. Yu has recently shown (in simulation) that ADIOS's word clustering could be used to help match words to percepts (Yu, 2006).

2.9 Summary of Previous Work in Machine Word Learning

The greatest success of previous word learning systems was the ability to find new words in audio without any prior knowledge of language beyond phoneme recognition. Roy, Yu, de Marcken, Brent, Oates, and the ADIOS team all included interesting systems for picking out words from raw audio or phonetic transcriptions. In fact, these systems handled that aspect of the problem so well that I will hardly cover it in this thesis; I will for the most part assume that words are already segmented when they are to be learned.[1]

Of all the previous word-learning systems, only Yu's had a good solution for finding the referent of a new word, by tracking gaze. The other systems generally ignored the problem of how the referent of a new word is picked out of the environment. Roy, Bailey, and Regier worked around the problem by making their referents the only objects in sight. de Marcken treated the possible meanings as atomic

[1]There are three other justifications for this move, besides the fact that word segmentation has been well-studied. First, there is evidence that children can perform word segmentation long before they begin to speak and without any particular semantics for the words (Saffran et al., 1996), so segmentation *then* semantics seems to be the order that children follow. Second, it seems more effective to concentrate on the "inductive step" of using language to learn more language than the "base case" of not knowing anything; the latter might be harder, but the former seems more useful when some definitions can be preprogrammed. Third, very good speech recognition software already exists, at least when it is trained to hear a single speaker, so it makes sense to leverage this technology and concentrate on the semantics instead.

symbols floating out in space, as if the words could be associated with anything in the environment. Siskind avoided the problem by assuming some kind of black box gives the possible meanings for an utterance. Only Yu had the ingenious solution of using an eyetracking device to find the referent of a new word, which allowed him to use real footage instead of oversimplified environments (Yu and Ballard, 2004).

Almost all of the systems restricted their learning to a particular part of speech (Bailey, 1997; Regier, 1996; Roy and Pentland, 2002). Those that did not impose such a restriction usually treated meanings as sets of atomic symbols (de Marcken, 1996; Siskind, 1994), and thereby avoided the complexities of dealing with the scalars and vectors that are typical of real data. Again, Yu's word learning system (Yu and Ballard, 2004) was a notable exception, handling both verbs and nouns – although it is difficult to see how words of a particularly compositional nature, such as prepositions, could have been learned by the system.

Most previous systems could not capture the compositional nature of language, either; once the words were learned, these systems had no way of combining them into sentences. Again, the primary exceptions to this rule were the systems that were not actually connected to a real world at all: de Marcken and Siskind's systems. de Marcken's system was interesting in that it could learn meanings for whole phrases before breaking them into compositional parts, but the overly simplistic meanings were themselves not compositional. Siskind suggested solving for how the words composed after their individual meanings were learned by reconstructing the training sentences, but without an experiment with a real robot (or even a realistic simulation) it isn't clear how well this would actually work.

Finally, the curious commonality between these systems that led to my current research is that they had no way of learning the words "I" and "you." It seems strange that such simple words should so confound existing approaches to word learning, but not one of them could handle these words. Roy's system would have incorrectly memorized a shape to go with each word, or at best would give no definition. Yu's system likewise would have assumed a meaning based on shape, color,

and texture, if it could find the referent for "I" at all. Regier and Bailey's systems were not designed to handle nouns at all. deMarcken's system would have remained flummoxed at the paradox that there is always a SPEAKER in the environment for every sentence, giving it no reason to associate SPEAKER with "I" in particular. Siskind's system might have done it under the right assumptions for its "black box" of meanings, but hoping that a black box solves the problem is hardly a solution.

We shall see in later chapters how, in the process of tackling the problem of "I" and "you," I eventually came to address several of the other issues, such as how to find a referent and learning truly compositional language. In the next chapter, I will cover some of the background material in psychology and linguistics that motivated my particular approach to the problem of word learning.

Chapter 3

Important Principles in Word Learning

We ended the last chapter with a puzzle: how could a machine learn the meanings of "I" and "you" from examples? This chapter will present some general principles of word learning and semantics that had heretofore been overlooked by the machine word learning community, including intensions and extensions, Montague semantics, and the Principle of Contrast, which all proved to have more applications than just the "I" and "you" problem.

3.1 Passive Word Learning in Children: The Case of "I" and "You"

Suppose a child could only ever learn words through one-on-one teaching sessions with other people. Thus, whenever the child heard the teacher say "you," the word would refer to the child. It would therefore be logically consistent for a child to believe that "you" refers exclusively to herself, as a kind of proper name (Dale and Crain-Thoreson, 1993). If the child believed this hypothesis, then during production, the child would use "you" in the place of "I" – a behavior called *pronoun reversal* (Dale and Crain-Thoreson, 1993; Lord and Paul, 1997). Pronoun reversal is quite common among children using pronouns before the age of two (Dale and Crain-Thoreson, 1993), suggesting that

young children do entertain this mistaken notion about the meanings of "I" and "you" at first.

The fact that children eventually do learn the correct meanings for "I" and "you" suggests that they are not just learning from speech directed toward them, but from overheard speech as well. To understand that "you" generally means whoever is being addressed, the child must hear "you" refer to other people besides herself.

Psychologist Oshima-Takane has done the most to argue that the observation of speech not directed at the child must necessarily figure into the learning of "I" and "you," presenting three different strands of evidence. First, there is some experimental evidence to suggest that children whose parents actively demonstrated the use of the word "you" for a few weeks learned its correct use earlier than those that received no such instruction (Oshima-Takane, 1988). Second, a neural network simulation produced the correct hypotheses about the meanings of "I" and "you" only when exposed to multiple (virtual) speakers, and its performance improved with more speakers (Oshima-Takane et al., 1999). Third, it appears that secondborn children tend to use these earlier than firstborn children, suggesting that they are learning from speech directed toward their siblings (Oshima-Takane et al., 1996).

Taken individually, each strand of Oshima-Takane's evidence is somewhat weak. The difference achieved by instruction was not significant due to small sample size. The neural networks were somewhat unnecessary given that the result they demonstrated more or less logically followed from the assumptions of their construction. The fact that secondborn children learned the deictic pronouns faster could have been explained through another mechanism, such as the availability of more input overall. Yet, taken together, they constitute the bulk of what is known about how typically developing children learn the words "I" and "you."

Interestingly, there are two groups of children for whom pronoun reversal is more common than is typical: blind children (Andersen et al., 1984) and autistic children (Lord and Paul, 1997). In both cases, it has been suggested that the children may be reversing pronouns more often than typical children because of a failure to shift perspective (Andersen et al., 1984; Loveland and Landry, 1986).

However, in the blind children's case there is not much reason to believe in an impaired "theory of mind" – in fact, researchers sometimes appear to infer this disability from fact of the pronoun reversal itself (Brown et al., 1997; Fraiberg and Adelson, 1977), despite no corroborating evidence.

A more reasonable theory of pronoun reversal might suggest that these two groups are more likely to reverse pronouns because they are not receiving the same amount of "overheard" input as other children. In the blind children's case, the blind children cannot see who the speaker is looking at when saying "you," making the word's reference more difficult to infer. Autistic individuals are thought to be at a disadvantage in learning to communicate with others partly because their initial disadvantages are compounded by a reduced overall social interaction (Lord et al., 1983); if they interact less with others, this will reduce the overall amount of linguistic input they receive. The theory that pronoun reversal occurs due to lack of data is also consistent with the case of precocious talkers; children at twenty months often either reverse pronouns or try to avoid them altogether (Dale and Crain-Thoreson, 1993). It has therefore been suggested that pronoun reversal is a stage that most children go through, but some simply progress past this stage before venturing into pronoun production (Clark, 1978). Thus, what is known about pronoun reversal supports Oshima-Takane's theory: in general, observing others' conversations seems to be necessary to get pronoun learning right.

I have heretofore been focusing on the case of "I" and "you" because it is such a clear example of the logical necessity of observing the conversations of others during language learning. But this line of argument is not limited to these deictic words; psychologists tend to argue that, in general, children can learn quite a bit from conversations that are not directed toward them, and that explicit instruction in a first language may not even be necessary. One famous study of a working-class African-American neighborhood in the 1970s found that adults there hardly spoke to infants at all; nevertheless, the children apparently did listen to conversations between adults, because the children could be overheard repeating sentences from these conversations that were not directed toward them (Heath, 1983). Others have argued that even in cultures where parental correction is the norm, it

often does not appear to do much of anything, as in this oft-cited example:

> Child: Nobody don't like me.
>
> Mother: No, say "Nobody likes me."
>
> Child: Nobody don't like me.
>
> *(this exchange is repeated eight times)*
>
> Mother: No, now listen carefully: say "Nobody likes me."
>
> Child: Oh! Nobody don't likes me.(McNeill, 1966)

This line of argument, and the sometimes anecdotal research that supports it, is more common in the study of children's grammar learning than word learning. One three-year study of three preschool children's verbal interactions with parents found that the adults were almost as likely to not express comprehension after their chilren's grammatical sentences (42%) as ungrammatical sentences (47%), while indicating comprehension was exactly as likely under either circumstance (45%; Brown and Hanlon, 1970). Marcus (1993) has used this finding to argue that adults do not give clear enough feedback to guide a supervised learning process for grammatical acquisition, though Marcus's argument is somewhat flawed in that he assumes a child must hear the exact same sentence often enough to be statistically certain of whether the sentence is correct or not before making use of it as a learning example (pointed out by Dana Angluin, personal communication). Interestingly, Brown and Hanlon (1970) suggested that parental disapproval does tend to occur in response to *semantic* errors. For example, "Mama isn't a boy, he a girl" received "That's right" in response, but "There's the animal farmhouse" received the reply "No, that's a lighthouse" (Brown and Hanlon, 1970, p. 49). Thus, one must be careful in extending the "no supervised learning" argument to semantics. In fact, there is some evidence that word learning rate improves with adult interaction (Akhtar et al., 1991; Harris et al., 1983; Tomasello and Todd, 1983). Nevertheless, it is often argued that even if adult instruction helps, there are cases in which such instruction is absent, and the children learn to speak anyway (Bloom, 2000; Heath, 1983; Pinker, 1994).

Why should all of this matter to the roboticist? The case of children's learning "I" and "you" shows that it is not only *possible* to learn words without explicit supervised feedback, but that it may in fact be *necessary* for certain classes of words. Unfortunately, this kind of learning challenges many assumptions that roboticists would like to make, because conversations that are not directed toward a robot are unlikely to be tailored for its benefit. In conversations between adults, conversation may be about things the robot can't see, gaze direction may be harder to follow, pointing gestures may be absent, and speech recognition may be more noisy due to rapid speech and out-of-vocabulary utterances. On the bright side, if children themselves are proof that laborious teaching sessions are not ultimately necessary to learn language, this opens up the exciting possibility that robot vocabulary learning may not need to be supervised at all.

Of course, even if the ideal is a system that can learn from a totally unstructured environment, experiments must necessarily be planned so that learning can occur within the span of an hour instead of years. Nevertheless, the ideal of learning from passive observation can be distilled into two principles to guide the present work. First, the present work will use purely unsupervised learning, on the principle that even if children can make use of corrections, the unsupervised part of their learning constitutes the bulk of their input and so deserves the focus of any attempt to replicate child-like performance. Second, all information about the learning situation or examples must be gleaned from the robot's sensors alone; there can be no extrasensory channel for the labeling of examples, the identification of referents, or cues about what was meant to be learned from the exchange. With these two principles, the ways in which the environment had to be structured for the robot's benefit could be limited to necessary concessions to the robot's limited sensors and the limited timeframes of the experiments, without sacrificing the possibility of scaling the same methods up to observing conversations in less constrained environments.

3.2 On the Meanings of Words: Extensions, Intensions, and Ideas

When one speaks of learning the "meaning" of a word, one must be careful, because a word can have several kinds of meanings. On the one hand, a word in a sentence has a *reference*, or *extension*: something in the world which it is about. There is also a word's *sense*, or *intension*: its shared meaning for the various speakers of a language, which allows one to judge the truth of a sentence that includes the word. Finally, a word can be associated with an *idea* or *exemplar*: a subjective association with the word that is unique to each individual.

The distinction between "sense," "reference," and "idea" can be traced back to the philosopher Gottlob Frege. His essay "On Sense and Reference" (Frege, 1892/2003) argued that even though the claims "$a = b$" and $a = a$" had essentially the same factual content when $a = b$ was true, it seemed that the two sentences should have different meanings, because one held *a priori* while the other did not. To take a more concrete example, Frege argued that even though "The morning star is a body illuminated by the Sun" and "The evening star is a body illuminated by the Sun" both contain the same factual information, since the two expressions refer to the same heavenly body, in fact the two sentences have different meanings. One could logically believe one of these statements and not the other if one did not know that the evening star and the morning star were one and the same, and so it must be the case that these were two different propositions somehow (Frege, 1892/2003).

To resolve this distinction in meaning, Frege argued that while the *reference* of "the evening star" was the same as "the morning star" – namely, the object we call Venus – the *sense* of these two expressions differed, because the two were not logically equivalent. It could have been the case that the evening star and the morning star were different objects; this just turns out to not be factually true. Though Frege did not express his arguments in "On Sense and Reference" in the form of propositional calculus, one can capture Frege's distinction by treating a word's "reference" as a particular literal in a symbolic calculus representation of the world (e.g., Evening Star v), while

allowing the "sense" of an expression to correspond to a lambda calculus expression in predicate logic (e.g., $\lambda X.EveningStar(X)$" vs. $\lambda X.MorningStar(X)$). This was the approach of Church, who "remains one of the most (and perhaps one of the only) prominent proponents of *broadly* Fregean semantics" (Klement, 2002, p. 96).

Interestingly, Frege also introduced an oft-overlooked third category of meaning in this essay: the "idea" of a word. While a "sense" was a meaning shared by all or most users of a word, an "idea" of a thing referred to the idiosyncratic mental image one imagined when thinking about the word. "In the light of this, one need have no scruples in speaking simply of *the* sense, whereas in the case of an idea one must, strictly speaking, add to whom it belongs and at what time" (Frege, 1892/2003, p. 178). This third kind of meaning, idiosyncratic to the learner, was largely omitted from the work that built on Frege's, presumably because it did not sit well within the formal logical systems that Frege's work inspired.

The philosopher Carnap formalized Frege's first two kinds of meaning, while ignoring the third (Carnap, 1947). In place of Frege's "sense" and "reference," Carnap proposed a system of "intension" and "extension." Carnap defined his extensions and intensions differently depending on whether he was speaking about words, sentences, or properties, but his definitions all had in common the distinction between logical equivalence (or "L-equivalence") and mere equivalence, perhaps best called "factual equivalence." If two expressions happened to denote the same set of individuals, they shared the same extension. In the case of a property, this extension would be the set of individuals that possessed the property; in the case of an expression that uniquely identified an individual, the extension was that individual. Two expressions only shared the same intension if the were logically equivalent; that is, if they shared the same truth-conditions. Thus, while "unicorn" and "dragon" might both share a null extension in a world that possesses neither beast, the two words have different intensions, because they are not logically equivalent. Though Carnap apparently did not invent the terms "intension" and "extension," as he refers to these terms as having "various customary uses" (Carnap, 1947, p. 18), he was the first to formalize Frege's distinction between sense and reference

(Dowty et al., 1981, p. 145).

In a modern approach, the *extension* of a linguistic fragment corresponds to a set of individuals, in the case of a predicate, or a truth value in the case of a sentence. An *intension* of a sentence is a function (usually logical in form) that, given a state of the world, returns whether the sentence is true, and the intension of a word is, correspondingly, a function that requires the rest of the sentence as well as the state of the world to return a truth value. Given the intension of a sentence and a world (or "model") to compare it with, one can generate the extension of the sentence and determine whether the sentence is true (Dowty et al., 1981). Frege's third kind of meaning, the "idea" generated by a word that is unique to each individual, can be compared to the idea of indexing intensions by speaker to generate extensions for indexicals and such (Dowty et al., 1981), but it is more similar to the idea of an "prototype" in psychology: a representation that is a typical example of a particular class (Rosch, 1973).

In making these distinctions in meaning, we can avoid several common mistakes that result from confusing these levels. Several previous systems for learning word meanings have confused extension with intension when learning word meanings. For example, in learning the meaning of "Mommy raised the ball," Siskind's system used representations such as GRASP(**mother**, **ball**) as a possible meaning for the utterance, then mapped the word "Mommy" to the symbol **mother** and "ball" to the symbol **ball** (Siskind, 1994). But this is a mistake, at least in the case of the word "ball"; if **ball** is a literal, we do not wish to map the word "ball" to *just that ball*, but to any object that shares the right properties to qualify as a ball. Siskind's system has thus failed to generalize the word "ball" at all, but it isn't obvious until one notes that the literal **ball** is in this case the extension, and not the intension, of the word. The same mistake continues to be made in modern word-learning work (e.g., Kate and Mooney, 2007).

Another mistake would be to confuse the intension of the word – its logical contribution to the overall proposition of a sentence – with its "idea,", consisting of its prototype or subjective associations. Latent Semantic Analysis, for example, attempts to get at word meaning by finding

which words appear near each other in text (Landauer and Dumais, 1997). While the resulting similarity matrix bears some resemblance to a net of subjective associations between words, the result is more or less useless for judging the truth value of a proposition. An LSA system may know that the word "sun" is highly associated with the words "light," "sky," and "clouds," but this still would give no basis for judging the truth of the statement, "The clouds have blocked out the sun" when presented with the appropriate sensory evidence. Associations are not sufficient for using words to communicate.

A system that attempts to get at word meaning by creating a sensory prototype for each word would likewise have difficulty in judging the truth of propositions in cases where some sensory information is irrelevant. For example, a person's imagined prototype for the word "human" may be of a particular gender or skin color, but the same person may know that these attributes do not matter for judging whether someone is human or not. This is important when evaluating techniques which define words as balls of radius r within a sensory space (e.g., Roy and Pentland, 2002); such techniques assume that every dimension is equally important when evaluating distance from an exemplar, which does not actually make sense in general. Painting a bachelor purple does not make him any less a bachelor; only marrying him does.

The TWIG system eventually settled on learning word intensions as the best way of capturing the meaning of a word. Not only does a formal semantics better equip the robot for natural language processing, but as Frege wrote in defending his notion of "sense," "one can hardly deny that mankind has a common store of thoughts which is transmitted from one generation to another" (Frege, 1892/2003, p. 177). In attempting to learn meanings common to many speakers of a language, rather than learning private and idiosyncratic meanings based on prototypes and associations, the learner is better equipped to both understand the truth or falsity of utterances and also to profit from the distinctions between words that previous generations have found useful.

3.3 The Principles of Contrast and Mutual Exclusivity

Once I had implemented a simple system for "I" and "you," a natural yet at first puzzling question was how one would learn the meaning of "he" from unsupervised observation.[1] The interesting thing about "he" is that it implies that the speaker is not referring to himself, and also not referring to the addressee – yet, with only positive examples to learn from, the learner has no way of knowing that using "he" in those cases is unacceptable. How could these qualifications on "he" be learned?

The answer was that these qualifications were implicit in contrast to "I" and "you." Because "I" and "you" are always used in their respective situations of the speaker referring to himself or the addressee, it is implicit that "he" is *not* to be used in these situations. Even though it is logically consistent with the evidence that "he" might validly refer to the speaker or the person being addressed, this mapping is not consistent with the assumption that different words refer to different things.

Clark states this "Principle of Contrast" (Clark, 1987) as follows: "Speakers assume that any difference in form signals a difference in meaning" (Clark, 2003, p. 144). If a different word is used for an object than the best-known word, the listener can assume that the new word refers to a different aspect of the word. If a speaker points to a rabbit and says, "That's my *pet*," a child who knows the word "rabbit" will assume that "pet" refers to something else besides its physical form. Similarly, if a speaker points to a kangaroo jumping and says "kangaroo," a child who knows the word "jump" will assume that the new word refers to the animal, and not to the act of jumping (Clark, 2003, p. 145).

Clark suggests that the Principle of Contrast is actually a corollary to a "Principle of Conventionality": "For certain meanings, speakers assume that there is a conventional form that should be used in the language community" (Clark, 2003, p. 143). Indeed, the two statements are more or less logically equivalent; the Principle of Contrast is "if word W has meaning M, then not-W implies not-M," while the Principle of Conventionality is "if word W has meaning M, then M implies W." We

[1] I apologize for the use of "he" as the sole third person pronoun throughout.

shall usually find it more useful to speak of the Principle of Contrast, since it speaks more directly to establishing distinctions in meaning between different words.

A stronger version of the Principle of Conventionality is the Principle of Mutual Exclusivity, which is that for any object, only one label is appropriate for the object itself (Markman and Wachtel, 1988). The Principle of Conventionality allows for different words to refer to the same object, as long as they have different general meanings, but the Principle of Mutual Exclusivity forbids different words for the same thing altogether. For example, when asked whether a doll is a "toy," a two-year-old may say "no"; when asked to give an example of a "toy," the child will then present a whole group of toys, assuming that "toy" cannot refer to the doll alone (Markman and Wachtel, 1988).

Markman and Wachtel have argued that mutual exclusivity is necessary to explain why children cease to use "dog" to mean "four-legged animal" when they learn the word "cat": the two terms cannot refer to the same object, and so the concept of "dog" is narrowed to not include cats. The Principle of Contrast, Markman and Wachtel argued, is insufficient to explain this phenomenon, because "four-legged animal" and "feline" already mean different things (Markman and Wachtel, 1988). On the other hand, Clark has reported children as young as 1;7 using "food" and "cereal" to refer to the same item, and a 2;1 child named Damon exclaimed "I 'Damon,' I 'cookie,' I 'sweetheart'!" (Clark, 2003, p. 149). If there is a constraint that forbids the use of multiple words to refer to the same object, it either disappears quickly or is specific to only certain kinds of words; perhaps both. Clark does concur with the report that children will reject superordinate category labels if they have more specific ones, e.g., "It's not an animal, it's a dog" (Clark, 1987). But, oddly enough, Clark also reports a child naming animals one by one as they are put back in their container – lion, tiger, zebra – only to conclude with the comment, "Animal back" (Clark, 2003).

This puzzle is not entirely worked out yet. The examples that Markman and Wachtel (1988) and Clark (2003) provide suggest that category words such as "animal" or "toy" are first learned as names for collections of things, and only later are used to refer to individuals within those collections. If so, then Markman and Wachtel's best example of mutual exclusivity has more to do with the

internal representations of category words than with a general word learning principle; Clark seems to have many examples of exclusivity being violated for other kinds of words. On the other hand, Markman and Wachtel raise a compelling point about the problem with the Principle of Contrast during word learning, namely that this principle alone cannot explain children's scaling back of their overextensions when new words are learned. It is for this reason, the usefulness of exclusivity as an assumption during learning for influencing word definitions, that TWIG assumes mutual exclusivity. (Problems with the assumption of mutual exclusivity will be revisited in section 8.4.1.)

3.4 Using Syntax to Aid Word Learning

Children also use syntax to infer aspects of word meaning. In one study, children between the ages of 3 and 5 were shown a picture of a pair of hands kneading confetti-like material in a low, round container. The experimenter would then use a new word, "sib," "niss," or "latt," either as a verb ("Do you know what it means to sib? In this picture you can see sibbing"), a count noun ("Have you ever seen a sib? In this picture, you can see a sib"), or a mass noun ("Have you ever seen any sib? In this picture, you can see sib"). When asked later to point to "sibbing," 10 of 16 children chose a picture of a similar movement over similar material or a similar container; 11 of 16 chose a similar container when asked to point to "a sib," and 12 of 16 pointed to similar confetti-like material when asked to point to "sib" (Brown, 1957). Though Bloom has pointed out that the children might have just been responsive to the word categories in the questions themselves, without learning them (Bloom, 2000), Brown reports that children sometimes immediately commented on the action in the appropriate form, e.g., "The latt is spilling" (Brown, 1957).

Other researchers have found syntax effects at earlier ages. When shown a pile of a novel substance, two-year-olds interpret a mass noun to refer to the substance, while a count noun is interpreted to refer to the accumulation of stuff (as with "puddle" or "pile") (Soja, 1992). Two- and three-year-olds told that an object is either "a zav" (noun) or "a zav one" (adjective) tend to assume that

"zav" refers to the object's form or class in the noun case, and its other properties such as color and texture in the adjective case (Taylor and Gelman, 1988).

It is possible to read too much into the importance of syntax, as children can sometimes make inferences about words without syntax cues. As Bloom points out, "If I pointed to a strange object and said 'gloppel,' you would take the word as a name for that kind of object; if I pointed to a strange substance and said 'gloppel,' you would take it as a name for that kind of substance; and if I pointed to a person and said 'gloppel,' you would take it as a name for that particular person" (Bloom, 2000, p. 197). Nevertheless, in many cases where syntax can be used to resolve ambiguity, young children appear to use it.

3.5 Other Proposed Heuristics in Children's Word Learning, and the Maxim of Quantity

A variety of other heuristics have been proposed to describe children's assumptions about word meaning. The "whole object assumption" is that a new word presented with a new object refers to the entire object, and not a particular feature, property, or aspect of the object (Markman, 1989, p. 26). Related to this idea is the idea of "shape bias," that new words tend to be extended to objects of a similar shape instead of size, color, or texture (Clark, 1973; Landau et al., 1998). Children have also been said to have a "basic level assumption," that new words are assumed to refer to "basic" categories such as *dog* instead of subcategories such as *dachshund* or superordinate categories such as *animal*, and a "type assumption" that new words refer to classes of things instead of individual instances of them (Clark, 1973).

One thing we might note about all of these biases is that they are regularly violated by even children's language; we certainly have words for parts ("nose"), for non-shape categories ("water"), for non-basic categories ("animal"), and for particular people or things ("mommy"). These effects tend to only be observed in ambiguous situations, where the adult labels a completely novel object.

Note, then, that if an adult were to present a novel object and violate one of these principles, the adult would also be violating Grice's Maxim of Quantity: "Be exactly as informative as is required" (Grice, 1975; Jurafsky and Martin, 2000). The most relevant thing to say in the presence of a novel object is indeed the type of object it is. Naming only a part, referring only to the object's texture or color, referring to an inappropriate level of abstraction, or giving the object its very own name would fail to provide complete and useful information about the object, while an object's basic level category provides clues to all of this information (except the object's name, but this usually does not exist). Once the most useful information about an object has been conveyed, only then does it make sense to speak of the object's other properties, parts, or categories – and this is in fact what adults do when speaking to children (Masur, 1997). It is also unsurprising that children extend words based on shape; the act of pointing to an object and naming it implies that the object is visually recognizable, and shape typically affects an object's functionality in a way that color and texture do not. Thus, even shape bias can be construed as an assumption that an adult has said something useful rather than useless.

We shall return later to this idea that assumed *informativeness* is of central importance in word learning. It is a powerful idea, and one that may explain many smaller findings.

Chapter 4

Materials: The Robot Nico

This chapter will describe the sensors and algorithms used on the robot Nico (Figure 4.1) in the experiments to be described in later chapters. Nico is an upper-torso humanoid robot that has been used for experiments in several different domains, including learning to point (Sun and Scassellati, 2004), modeling infant looking-time experiments (Lovett and Scassellati, 2004), testing algorithms for intention recognition (Crick et al., 2007), drumming to a conductor's beat (Crick et al., 2006), and self-recognition (Gold and Scassellati, 2007b). It was not constructed specifically for the word learning experiments of this thesis. The word learning algorithms I shall discuss in chapters 6 and 7 could have been implemented on a robot with different sensory capabilities, but it is useful to keep the underlying robotic architecture in mind when dealing with such abstractions.

4.1 Vision

4.1.1 From camera to image stream

Nico possesses four 1/4 inch color CCD (charge-coupled device) cameras: a short focal length (f/2.8 aperture, f=2.2mm, 80^o horizontal viewing angle) and a long focal length (f/3.5 aperture, f=15mm, 13.5^o horizontal viewing angle) camera for each compound "eye" on the left and right side of its

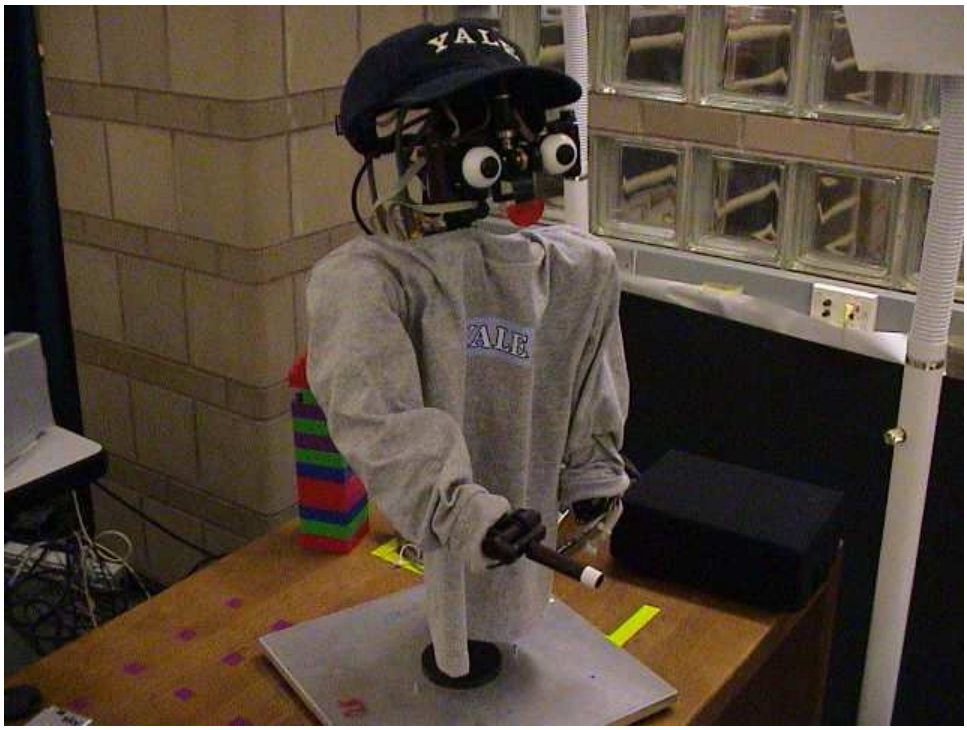

Figure 4.1: Nico, the robotic platform for the implementations and experiments to be described.

face.[1] These cameras correspond to low resolution, wide angle peripheral vision and high resolution, narrow angle foveal vision, respectively. Unless otherwise indicated, the experiments to be described here used only Nico's right wide-angle camera, since the extra cameras would introduce only needless complexity in the absence of depth perception and fine-detail image processing. The cameras produced images with 470 lines of resolution at a fixed rate of 30 frames per second.

The cameras were connected to digital signal processing control units[2] that could automatically adjust gain on the camera signal, but this process tended to be noisy and was not used. Imagenation PXC200a frame grabbers captured the frames from these control units; these frame grabbers could process up to 640×480 pixels and could match the 30 fps rate of the cameras.

In software, frames could be provided at a resolution of either 320×240 or 640×480; unless otherwise indicated, 320×240 resolution was used for faster image processing. Lens distortion could be removed by the grabber software, but this feature was usually not necessary, and was not used

[1]The lenses are Elmo product code 9821 and code 9831, respectively. The cameras were Elmo product QN42H.
[2]Elmo product CC421E.

unless otherwise indicated. The grabber software was run on its own 3.2 GHz Intel Pentium 4 running QNX[3]. At this resolution, it provided frames at a full 30 frames per second.

4.1.2 Faces and their facing

Faces were detected in each frame using the implementation of the Viola and Jones object-finding algorithm (Viola and Jones, 2004) that comes with the Intel OpenCV vision library (Bradski et al., 2005).

The Viola and Jones method for face detection relies on a series of weak classifiers (decision tree stumps) that detect adjacent rectangles of light and dark regions. These classifiers' values can be computed quickly using an "integral image" of the original image, an image that stores at each pixel the sum of the pixel values up to that point. The overall value of the classifier can then be computed by accessing only the rectangle corners within this integral image, for a constant time evaluation of the difference in brightness between the light and dark regions of the classifier. The orientations and placements of the rectangles are analagous to the center-surround, edge-detecting, and line-detecting cells of the human visual system (Figure 4.2). Each weak classifier outputs a simple "accept" or "reject" based on whether the brightness difference falls within a specific range. These weak classifiers are combined using a boosting algorithm (Freund and Shapire, 1996), trained on examples of the object to be found that have been reduced and scaled to 24×24 pixels in order to reduce the dimensionality of the feature space. In the final classifier, multiple boosted classifiers are used in serial, with the more quickly evaluated classifiers rejecting obviously bad examples before they ever reach the more complicated and time-consuming classifiers. The classifiers are easily scaled in size by changing the rectangle endpoints, and so the search can take place at multiple scales using essentially the same classifiers (Bradski et al., 2005; Viola and Jones, 2004).

The OpenCV implementation of Viola and Jones includes two trained classifiers: one for detecting

[3]QNX never officially released drivers for the Imagenation PXC200a compatible with QNX 6; the drivers on Nico were the result of combining untested, unreleased source code from QNX for the related PXC200 frame grabbers with code from the PXC200a DOS drivers.(Herberg, 2002)

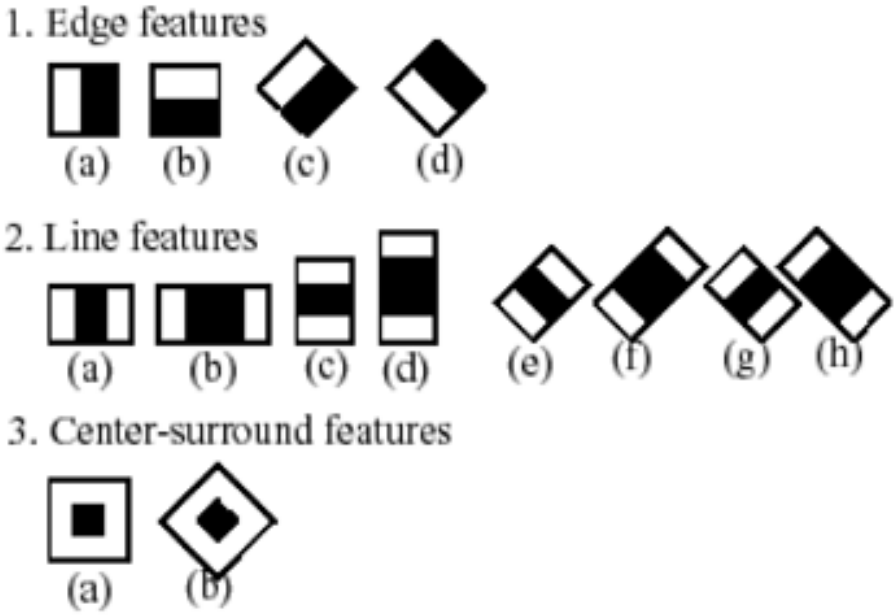

Figure 4.2: The features which the OpenCV implmentation of the Viola and Jones face-finding algorithm (Viola and Jones, 2004) uses to find faces. From (Bradski et al., 2005).

faces observed head-on, and another that detects faces viewed in profile. Each detector was run on a separate 3.2 GHz Pentium 4, processing the 320×240 images from the cameras at roughly 10 fps.

Then, to combine the information from the two detectors into a single classification of "no face," "looking at nico," or "looking across," I used the forward component of the forward-backward algorithm (Rabiner, 1989), sometimes simply called the forward algorithm (Russell and Norvig, 2003). The output from the two face detectors can be seen as two noisy evidence nodes that both stem from the same underlying state, which falls into one of the three categories just mentioned. This underlying state can change over time, either when a person looks a different direction or moves out of the camera's field of view, but is expected to generally remain the same over time. Using the forward algorithm, I can integrate all of the evidence observed so far from both detectors into the estimation of the face/no-face classification and the head pose. (The forward-backward algorithm is unnecessary here because the algorithms to come only use the classification of the most recent frame, which is unaffected by the backward propagation of probabilities.)

P(forward hit | looking at Nico)	0.972
P(forward hit | looking across)	0.0226
P(forward hit | no face)	0.0535
P(profile hit | looking at Nico)	0.0700
P(profile hit | looking across)	0.855
P(profile hit | no face)	0.0099

Table 4.1: Conditional probabilities for the probabilistic model of facing direction.

To estimate the conditional probabilities of this model, I recorded 3 minutes (1800 frames) of output from these detectors as I looked in one of the two directions (across Nico or at Nico) from various locations, and counted the number of hits from each detector for each ground truth state. This resulted in the conditional probabilities shown in Table 4.1.

I estimated the rate of change of facing direction to be roughly twice every four seconds, which at 10 fps results in a probability of changing facing direction of 0.05; a probability of transitioning from "no face" to a face (i.e., a person entering the field of view) to be roughly once every 5 minutes, or $1/3000 = 0.00033$; and the probability of leaving the field of view to be even less, 0.00001. Given all of these probabilities, the resulting Hidden Markov Model could integrate all of the output over time from both detectors and give a likelihood of each state that could be updated in constant time, using the following forward equation:

$$P(\Phi^i_{t+1}) = P(F_{t+1}|\Phi^i_{t+1})P(P_{t+1}|\Phi^i_{t+1})\sum_j P(\Phi^i_{t+1}|\Phi^j_t)P(\Phi^j_t) \tag{4.1}$$

where Φ^i_t is the event of hidden state i holding at time t, F_t is the output of the forward-looking face detector at time t, and P_t is the output of the profile face detector at time t. These calculations are carried out for any area of the image where a face is detected, where a detection in frame $t+1$ is assumed to refer to the same face as a detection in frame t if their regions overlap. If no face was detected recently in an area, the last detected region for an object is used for computing its overlap with a face in the most recent frame. The initial probabilities for each state were P(no face) = 0.99, P(looking across) = P(looking at Nico) = 0.005.

In addition to providing a means of deciding between classifications when both detectors returned "true," using the forward algorithm also presumably reduced the number of false positives from each detector, since a false positive would need to be consistent over time and across detectors for it to gain a non-negligible probability.

Occasionally, it was necessary to estimate the coordinates of a person in the room using the face detector output. This was done by assuming a constant depth of 60cm and solving for the other coordinates using the position of a detected face's centroid.

4.1.3 Color blob detection

Often, it is useful to have a detector for a particular uncommon color, so that objects of that color can be easily found in a complicated visual scene. In the experiments to be described, that color was the bright yellow used for Lego Duplo blocks. The filter code was adapted from that used in Sun and Scassellati (2004).

A filter discarded pixels with average RGB luminance less than a threshold (50/255) and marked pixels as salient if their luminance-normalized red and green values were both at least 1.5 times their luminance-normalized blue values. A second pass through the image applied labels to salient pixels in preparation for grouping them, applying the same label as any salient pixels found within 3 pixels above or 5 pixels to the right, or a new label otherwise. If different labels were found within this range, the regions were merged by applying one label to all the pixels in the area. A third pass created bounding boxes around each salient area by finding the minimum and maximum rows and columns for each label, as well as the centroid of each group.

4.2 Audio

4.2.1 Microphones and sound localization

For microphones, the system used a dual-channel microphone setup consisting of two microphone heads, connected via separate 6 ft. (1.83m) cables to a preamplifier powered by a 9V battery.

The two microphone heads were located roughly 30cm apart and 50cm in front of the robot. This location reduced the noise from the robot's motors and rack fans, and made speech detection and recognition slightly less noisy (see below).

The louder channel of the two microphones was found by counting the number of 10 ms segments in which one channel's output was at least 1.1 times as loud as the other and exceeded an adjustable volume threshold.

4.2.2 The Sphinx-4 speech recognition system

The robot used the Sphinx-4 speech recognition system (Walker et al., 2004) to segment audio into words. Since Sphinx-4 is highly configurable, I shall go into detail about some of the systems used in this particular implementation.

Sphinx-4 continuously monitored the average background volume and the average signal volume, marking audio as "speech" if the difference between the two exceeded a threshold and "non-speech" otherwise. An endpoint marking the beginning of speech was inserted where the first transition from non-speech to speech occurred, and an endpoint marking the end of speech was inserted at the beginning of the first 500 continuous milliseconds of non-speech.

Once a set of speech endpoints was found, the audio marked as speech was changed into a sequence of Mel-Frequency Cepstral Coefficients (MFCCs), sampled at a rate of 16000 Hz. Cepstral coefficients are a common means of representing the acoustic signal in speech recognition (Jurafsky and Martin, 2000), and the use of Mel frequency scaling to better model human psychoacoustics appears to improve recognition performance (Davis and Mermelstein, 1980).

Sphinx used the Viterbi algorithm (i.e., dynamic programming) combined with beam search pruning to attempt to find the maximum-likelihood path through a large chain of Hidden Markov Models (HMM) that would have produced the observed MFCC sequence. The structure of the search graph was determined by several nested components. The possible sequences of words and were determined by a simple context-free grammar that varied from one experiment to the next. For each word, the CMU phonetic dictionary [4] determined the possible phoneme sequences for each word. Each word's phoneme sequence was represented by a chain of HMMs that had been trained using readings of the Wall Street Journal as a corpus. Because phonemes can sound different depending on the surrounding phonemes and their position in a word, a separate HMM was used for each possible phonetic context (preceding and following phonemes) and word position context (beginning, end, middle, or whole word) for each phoneme. The search parameters were kept at their defaults.[5] The search was kept small with a small context-free grammar (CFG) containing the grammatical forms and words that were used in each experiment, which limited the recognition possibilities.

This setup was not optimal either for experimental success or maximum flexibility in learning. Had the HMMs been trained in the lab instead of using the Wall Street Journal acoustic model, recognition performance would have undoubtedly been better; as it was, the ambient noise from computer fans and motors typically resulted in quite poor recognition performance. However, collecting an acoustic corpus with a reasonable amount of data to cover all possible phoneme contexts is no small feat, and creating a smaller, specific acoustic corpus for each experiment would have resulted in much less flexibility to try new designs. Likewise, it would have been nice to perform experiments in the absence of a phonetic dictionary and context-free grammar, since relying on this existing language model reduces the system's flexibility in recognizing and learning new words. However, removing the language model so that the acoustic model alone was used to produce phoneme sequences (using whole-word contexts for each phoneme) reduced the recognition performance from very noisy to totally unworkable, with the system producing unrecognizable phoneme sequences as

[4] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[5] absoluteBeamWidth=-1, relativeBeamWidth=1E-80

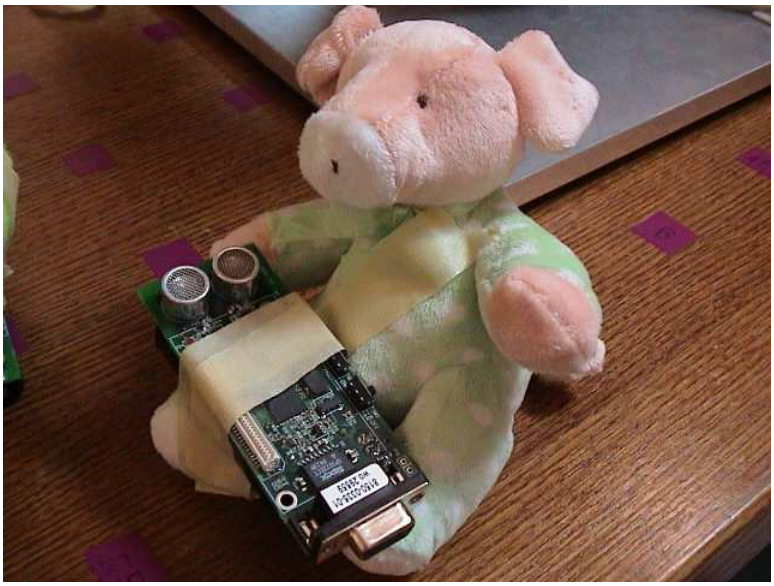

Figure 4.3: A Cricket sensor, attached to a stuffed pig used in the learning experiment of Gold et al. (2007).

output. Thus, the phonetic dictionary and CFG-based language model were used for better speech recognition.

4.3 The Cricket Indoor Location System

In later incarnations of TWIG, the Cricket Indoor Location System (Priyantha, 2005) was used to find the locations of objects in the room.[6] Each battery-powered Cricket node (Figure 4.3) sends out a simultaneous radio broadcast and ultrasound "chirp"; the difference in arrival times at a Cricket sensor can be used to calculate the distance between sender and receiver. The sensed distances between a mobile node and each stationary sensor can be combined by using a least-squares calculation, thus giving a best approximation as to the x, y, and z coordinates of the Cricket node.

Eight Cricket sensors were attached to the ceiling in a square, each roughly 1.5m from its neighbors. A ninth cricket hung from an overhead light in the center, 95cm from the ceiling, and a tenth was attached to the wall to Nico's right, 128 cm from the ceiling and 290cm from the overhead light.

[6]The Cricket system described here was implemented by Chris Crick, with some help from Marek Doniec in integrating it with the rest of the robot.

Typically two Cricket beacons were used for the objects of interest in a scene. Humans and Nico did not use crickets for localization, since there simply were not enough cricket beacons. Nico, being stationary, assumed a constant position at the origin of the robot's coordinate system, while the position of each human in the room was estimated from the output of the face detectors (see above).

4.4 Processing and Communication

Because any given video processing module tends to consume almost all of a CPU's cycles when running at a high frame rate, processing was split among several computers, which communicated via TCP/IP.

The frame grabber, the profile face detector, the forward face detector, the color module, and a module that integrated all of these inputs each ran on a separate 3.2 GHz Intel processor under QNX. The Cricket calculations were performed on a 2.26 GHz desktop running Windows XP, while the audio and language processing were performed on a 3.4 GHz laptop running Windows XP. The various modules communicated over TCP/IP, using a 1000 Mbps switch in the case of the QNX modules, a 100 Mbps wired ethernet connection in the case of the Windows machine performing Cricket calculations, and a 24.0 Mbps wireless connection to the laptop.

For most experiments, communication was handled via libtcpip a library of message-broadcasting functions developed by Marek Doniec.[7] This library allowed a module to write output to a single port that could be read by any process that connected to it. Each client process reading from a particular output buffer received its own thread that would send a copy of the buffer whenever it was requested. Reads on receiving modules could either block while waiting for new data from a buffer, or be left unblocking and have the chance of not retrieving any data if there was no new data since the last read on the port. Connections from the frame grabber to the sensory processing modules

[7]The experiments to be described in Chapter 5 were done using the "porter" system, a predecessor to libtcpip which used QNX-native message passing; but the porter system tended to be buggy, was difficult to port across platforms (or even from one version of QNX to the next), and resulted in network congestion. Its operation in practice was otherwise similar to the functionality described here.

were blocking, while all other connections were non-blocking.

Typically if a consumer process did not read fast enough to consume all frames or messages written to a buffer, the frames or messages in between reads were overwritten and lost. This allowed slow consumers such as the face detectors to remain in sync with the current image. However, in cases where all messages had to be kept in order to keep the world state consistent, all messages were read in order. This was done for messages from the Cricket system.

Because the libtcpip system was not ported to Prolog, the language inference system described in Chapter 6 communicated with the robot via a simple TCP/IP socket.

Chapter 5

Preludes to TWIG: Learning "I" and "You" with Chi-Square Methods

This chapter describes my research on learning "I" and "You" prior to the implementation of the TWIG system. These experiments did not use the full TWIG machinery described in chapters 6 and 7: the system in these experiments did not actually parse full sentences, and did not have the more sophisticated representations of meaning that TWIG used.

Nevertheless, this early system had the advantage of not needing utterances to remain in-grammar, since it did not use grammatical structure, and so it was the only incarnation of my word-learning system that was run on real transcripts of mother-child interactions. It also introduced the inference of reference from sentence context, one of the key ideas that would become central to TWIG. Finally, because the experiments run in this chapter were mostly performed in simulation, it allowed me to determine how robust my methods were to environment size and speaker localization error.

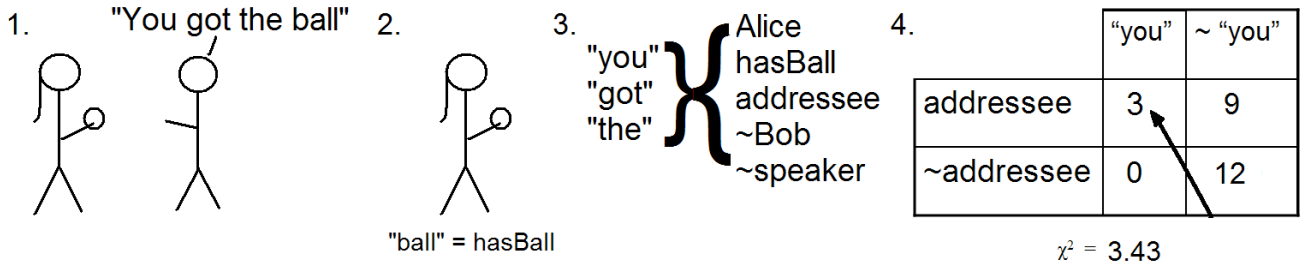

Figure 5.1: An example of how the system I described in (Gold and Scassellati, 2006e) associates words with properties. (1) Bob says to Alice, "You got the ball." The speech recognition software turns the speech into a string, while localization determines that Bob is the speaker and Alice is the addressee. (2) The system searches for words it already understands, and finds that "ball" corresponds to the hasBall property. The system designates Alice as the referent for the remaining words, because she has the ball. (3) Each word that was not understood is associated with Alice's properties, by increasing the words' collocation counts with those properties. (4) The updated collocation counts are placed in 2×2 chi-square tables to compute the significance of each word-property association.

5.1 The Chi-Square Method

Before TWIG, my primary methodology was to use chi-square tests to find words that were strongly associated with particular properties (Figure 5.1). The essential idea is that one can count the number of times a word appears in dialogue to estimate the probability $P(word)$ that the word appears in an arbitrary sentence, count the number of lines of dialogue in which the subject of the sentence satisfies a property att to estimate the probability $P(att)$ that a property is true of the subject of the sentence, and then calculate the number of times one would expect to see the word and property occur together if the two events were independent by multiplying these two probabilities. One can perform similar counts for the events $\neg word$ and $\neg att$, that the word does not appear in the sentence or the property is not true of the subject of the sentence, respectively. The likelihood that the word and property occur together only because of chance is then calculable by a chi-square test:

$$\chi^2 = \sum_{ij} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{5.1}$$

where O_{ij} and E_{ij} are the observed and expected values in the 2×2 table of events corresponding

to each of the possibilities for whether the word and/or property was observed.

The notions of "property" and "subject of the sentence" are here much simpler than they would become in the final system (Chapters 6 and 7). In the models described here, "properties" are all boolean values that can be determined to be true or false for each actor in the world, and a sentence refers to an actor if and only if it contains a word associated with one of the properties that is true of that actor. Thus, these systems contain no notion of parsing or grammar, and only a simple notion of reference and semantics.

For learning "I" and "you," the goal was to show that this simple chi-square mechanism could result in "I" being most strongly associated with a property of being SPEAKER, and "you" most strongly associated with the property ADDRESSEE.

5.2 Experiments in Simulation

5.2.1 Overview

Gold and Scassellati (2006d) introduced the two central ideas that would end up evolving into the two halves of the TWIG system. The first idea was that chi-square tests for significance could uncover links between words and properties, even in noisy environments and even if the words appear within full sentences. The second idea was that many of the conundrums surrounding "I" and "you" could be resolved if the learner could use the other words in the sentence to determine who was being talked about.

Chi-square tests are sometimes used in natural language processing to find words that appear together more often than one would expect due to chance (Manning and Schütze, 1999a); my initial idea was to perform chi-square tests to discover word-property collocations as well. A great advantage of the chi-square method is that it does not require parsing, which would be difficult with the ill-formed sentences that commonly appear in transcripts of real speech.

The idea of using sentence context to determine reference came out of the simple fact that there

is always a "speaker" and an "addressee" for each utterance, and so there was no way that these would become associated with particular words if they were always "true" for each utterance. Thus, the system had to focus on a particular person in the exchange in order to have these attributes be sometimes true, sometimes not.

My original experiments were performed on transcripts of child-directed speech. The CHILDES corpus (MacWhinney, 2000) contained transcriptions of child-caregiver interactions, and seemed to provide a natural data set to run simulations with. The corpus contained no stage directions, but luckily, one transcript (Bohannon, 1976) contained a fairly straightforward game of "catch" between a mother and her child, in which they exclaimed "I got the ball" at appropriate moments. It was straightforward to turn the transcript into a modest simulation of a learner perceiving this game of catch.

When this simulation failed to learn that "you" referred to the person being addressed, I noticed that it was difficult to find examples in the CHILDES database of statements involving the word "you." Much more often, "you" was used in questions about what the addressee wanted. A modified version of the system that assumed questions that included the word "want" referred to the addressee learned that "you" referred to the addressee as well.

This difference suggested an explanation for pronoun reversal, the phenomenon in which blind (Andersen et al., 1984; Fraiberg and Adelson, 1977), autistic (Lord and Paul, 1997), or particularly young (Dale and Crain-Thoreson, 1993) children confused "I" and "you": perhaps these learners were failing to correctly surmise to whom "want" questions were directed. Blind children might fail to see where the speaker was looking, while autistic and very young children might lack the ability to reason about other people's desires (Baron-Cohen, 1995), and thus fail to understand those critical "want" questions. A third experiment tested this idea by simulating a learner who always assumed questions including "want" referred to the learner himself. For a sufficiently descriptive model, this resulted in the system learning that "you" always referred to something about himself as well.

5.2.2 Simulation methods: "I"

The system was run in simulation on a transcript of a mother and her child playing catch (Bohannon, 1976; Bohannon and Marquis, 1977; Stine and Bohannon, 1983), taken from the CHILDES database (MacWhinney, 2000). Only the raw words were used from these transcripts, and not the CHILDES part-of-speech annotations. No "stop words" (filler words such as "the") were omitted from the analysis. The corpus was relatively small, consisting of 1707 words in 308 sentences. Of these, only 372 appeared in sentences with understood referents.

The simulation consisted simply of a list of boolean attributes for each participant in the scene. Actions that could be inferred from dialogue (e.g.: "You got it!", "Why are you blowing on it?", etc.) were added as stage directions to the transcript, and changed the state of the relevant simulated actor when they were read from the transcript. Annotating the text in this way produced six attributes that changed over the course of the text: throwing, catching, missing a catch, getting hit on the head, blowing on an object, and falling down.

The words that referred to these actions – "threw," "throw," "got," "catch," "caught," "dropped," "missed," "hit," "blowing," and "blew" were given to the system as referring to the corresponding stage directions. "Mommy" and "Bax" (the child's name) were given to the system as referring to the mother and child.

To more fully model a complicated environment, I added additional properties to the actors which did not correspond to anything in the script, but that changed with probability 1/2 from line to line. These represented other attributes of the attended objects that were changing, and could potentially be associated with the dialogue by accident. The number of these "dynamic variables," as I called them, varied according to experimental condition, as described below. In addition, six attributes with random values remained fixed for each actor over the course of the interaction, to simulate miscellaneous properties of the actors that were unchanging.

Finally, the two attributes SPEAKER and ADDRESSEE were set to true or false for each actor depending on who was speaking. Errors in the localization process could be artificially injected by

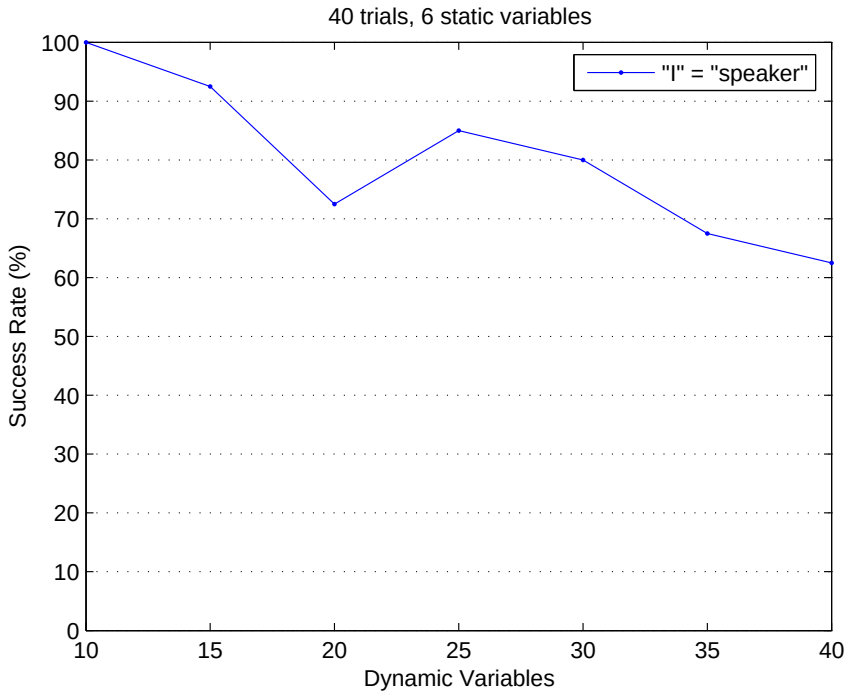

Figure 5.2: Success rates in simulation for learning "I" for varying numbers of speaker attributes that changed over the course of the simulation. No trial correctly surmised the meaning of "you" under these conditions. Reprinted from Gold and Scassellati (2006d).

swapping the attributes of the two actors with a fixed probability.

The simulation was run for varying numbers of dynamic variables (between 10 and 40) and for varying rates of speaker identification error (between 0% and 10%). Each condition was run 40 times.

5.2.3 Simulation results: "I"

A trial was considered a success if, by the end of the exchange, the most significant property-word association for the word "I" was associated with the attribute *speaking*. Figure 5.2 shows the success rates out of 40 trials for varying numbers of dynamic variables, which determined environmental complexity. These results suggested that a reasonable increase in the number of attributes sensed would not unduly cripple the learning.

Figure 5.3 shows the impact of speaker identification error on the success rate for learning "I." Even for small error rates, the chance of correctly learning the meaning of "I" falls off substantially,

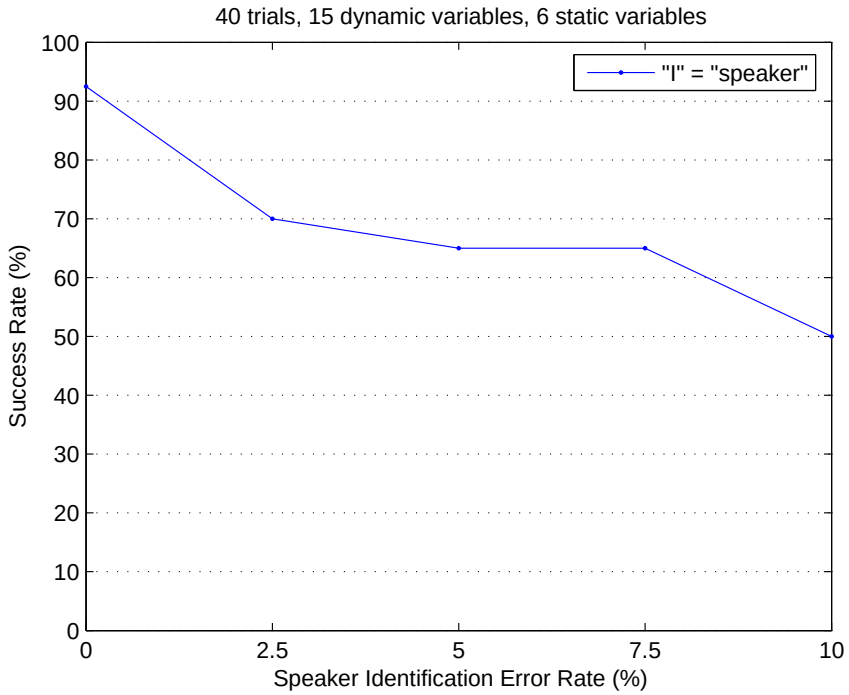

Figure 5.3: Success rates in simulation for learning "I" with varying rates of speaker identification errors. Reprinted from Gold and Scassellati (2006d).

suggesting that errors in localization offered more potential for disrupting learning than environmental complexity.

The trials produced very few erroneous associations. In the 6 dynamic variables case, only two words were given new associations: "I" was associated with speaking, and "it" was associated with catching (presumably because of the frequency of the phrase "I got it" in the script). Even in the 40 dynamic variable case, only 10 (25%) of the trials produced more associations than this, associating "you" or "the" with arbitrary dynamic variables.

However, in none of these trials was "you" correctly associated with the property of ADDRESSEE. The script contained very few utterances of "you got it," and I wondered whether this was simply a problem of insufficient evidence. Combing the rest of the CHILDES archive, I found few examples of statements involving "you"; most of the time "you" was used in a question, such as "You wanna sit in my lap?" (Bohannon, 1976) It stood to reason, I thought, that one generally doesn't tell another person about themselves; one asks. I also thought that perhaps the most common question

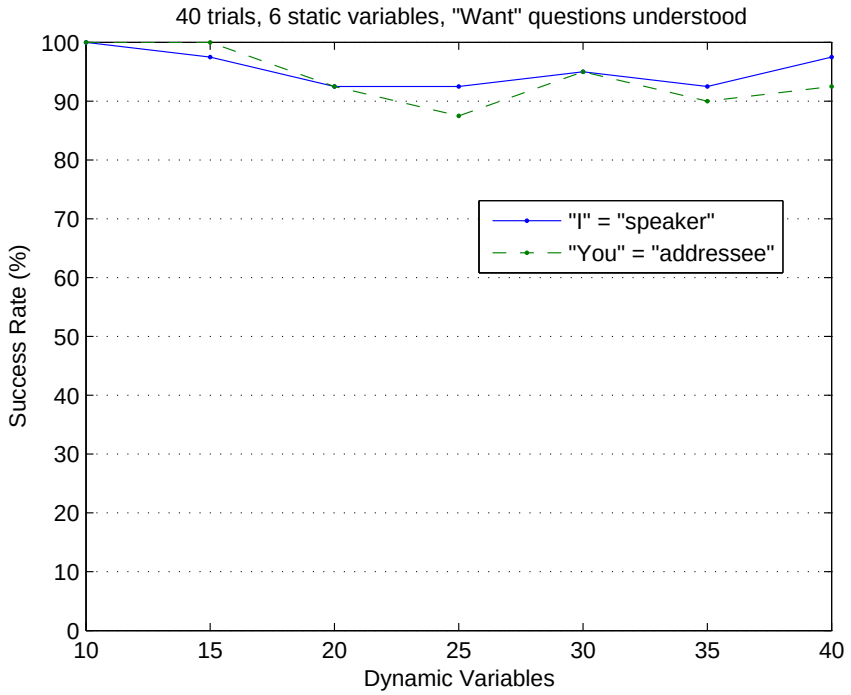

Figure 5.4: Success rates in simulation for learning "I" and "you" when questions about wants were assumed to refer to the addressee. From Gold and Scassellati (2006d).

to ask was about someone's mental state, which is not immediately observable; particularly, what the person in question wants. This led to the design of the second simulation experiment.

5.2.4 Simulation methods: "You"

The methods were the same as in the "I" learning simulation above, including using the same script, but with one small change. Questions that included variants on the word "want" were assumed to refer to the person with property ADDRESSEE, and all words in such sentences were assumed to refer to some property of that person.

5.2.5 Simulation results: "You"

The results for this experiment are shown in Figures 5.4 and 5.5. The correct binding for "you" was correctly learned even for large numbers of distractor variables, and its success rate was comparable

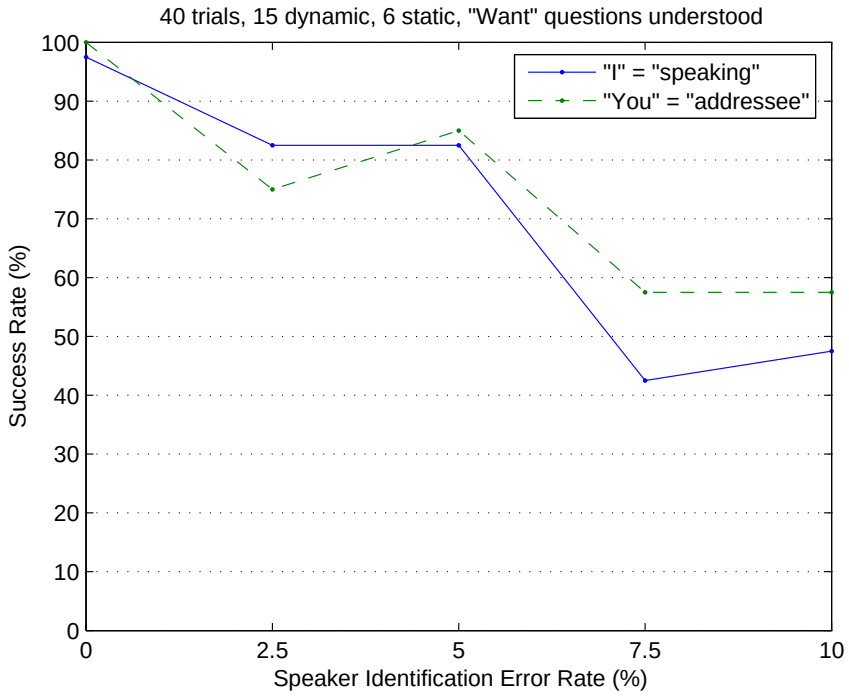

Figure 5.5: As figure 5.4, but with varying rates of speaker identification error. From Gold and Scassellati (2006d).

to that of "I" when speaker confusion occurred.

In addition, extraneous dynamic variables were not as detrimental to the learning of "I" in this trial, because the additional sentences that could now be comprehended provided additional statistical evidence for the "I" hypothesis, and evidence against association with other dynamic variables. (The number of words in sentences with at least one "known" word increased from 372 to 417.) This suggests that as vocabulary size increases, conversations can be used more efficiently to test the meanings of words.

Increasing the number of unchanging variables did not affect the performance of the system. This is unsurprising, as there were only four functionally distinct kinds of static variables in the simulation, depending on whether they were true of the mother and whether they were true of the child.

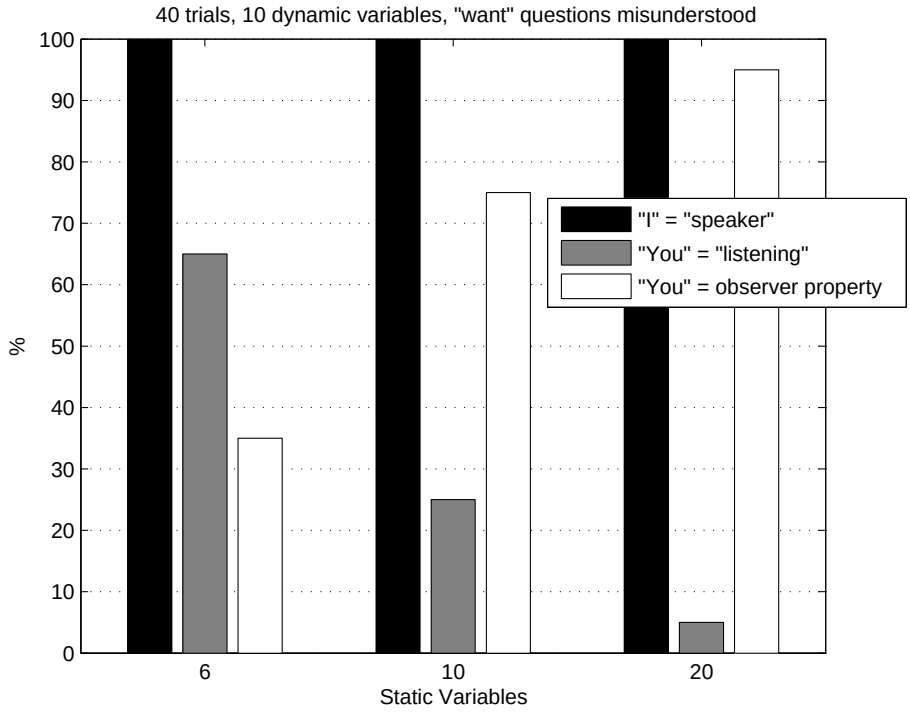

Figure 5.6: Response of the simulation when it assumed that questions about wants were always about itself. From Gold and Scassellati (2006d).

5.2.6 Pronoun reversal simulation: Methods

The simulation was run as before, but with three participants in the simulation: the mother, the child, and a learner, with random static and dynamic properties generated as for the other participants but no stage directions. Questions including variants on the word "want" were assumed by the system to refer to the learner, instead of the addressee. In addition, a new property of LISTENER was added to the system, distinct from ADDRESSEE, which was true of both non-speakers in the simulation.

The simulation was run for varying numbers of properties that remained unchanged (between 6 and 20), with 40 trials for each condition.

5.2.7 Pronoun reversal simulation: Results

As Figure 5.6 shows, when the number of random properties was increased sufficiently to ensure that at least one variable was uniquely true of the observer as well as the person being addressed, the

observer assumed that the word "you" referred to this property. In other words, when the system assumed that "want" questions were always directed at itself, it learned that "you" always referred to itself as well.

When there was no property that was uniquely true for the observer, the word "you" was typically associated with the property of LISTENER, which was always true for the observer.

Despite some evidence to the contrary in the script (e.g., "you got it," which would still associate you with only the catcher), none of the trials correctly associated "you" with the addressee alone when "want" was incorrectly associated with the learner. However, all trials still correctly identified "I" as referring to the property of SPEAKER.

5.2.8 Discussion

Gold and Scassellati (2006d) introduced quite a number of ideas that would come to play an important role in TWIG: the significance-testing approach to property learning, the idea of using other words in the sentence to establish reference, and a tentative introduction of a "theory of mind" concept to explain pronoun reversal.

One valid criticism of these simultions is that the experimental designs were often overly complex, obscuring the results. While the "dynamic variables" approach was a fine idea for making the simulation a bit more realistically complex, random "static variables" were a bit unnecessary; I could have simply introduced one variable per person that was uniquely true for that person, as I did later. This would have made the pronoun reversal experiment much easier to interpret.

The idea of using sentence context to narrow down reference was also obscured in this paper's original presentation; I didn't realize at the time how central it was. Some readers of the paper were confused as to why "I" in particular should be associated with SPEAKER, but not other words, even though somebody is always speaking; they had missed the idea that only the sentence referent's properties get associated with the words, and that SPEAKER could sometimes effectively be false.

Using the same transcript over and over for the various simulations was a bit scientifically suspect,

since the later experiments could be tailored to that specific transcript. It is therefore unclear how general the importance of the word "want" in learning pronouns really is; it could have been a peculiarity of this transcript. However, I could not find another transcript in the CHILDES database that included plenty of pronouns and also clearly indicated the participants' actions.

5.3 Robotic Implementation and Analysis

5.3.1 Overview

The experiments described in Gold and Scassellati (2006e) were streamlined, real-world versions of the experiments performed in simulation in Gold and Scassellati (2006d). While the fake variables introduced in Gold and Scassellati (2006d) had been an admirable attempt to make a simple simulation a bit more realistically complex, that paper had also shown that the environmental complexity did not particularly matter, and I had felt that they considerably muddied the presentation, particularly when explaining pronoun reversal. Fake noise seemed to have no place in a robotic implementation that seemed to have plenty of sources of real noise: in recognition, localization, and syncing audio to video.

I was also dissatisfied with the explanation of pronoun reversal given in Gold and Scassellati (2006d), because it explained congenitally blind children's reversal in the same way as autistic pronoun reversal. While some researchers have argued that blind children's pronoun reversal might stem from a lack of self-understanding (Fraiberg and Adelson, 1977), a deficiency in perspective-taking (Andersen et al., 1984), or other processing difficulties that resemble autism (Brown et al., 1997), it seemed that a simpler explanation might be that they often couldn't tell who was being addressed or what was being talked about, and so would learn pronouns much later. A robotic experiment in which the robot was blind and could only tell when it had the ball, confirmed that under such circumstances the learner would associate "you" only with himself.

Finally, the experiments in Gold and Scassellati (2006d) had given success rates for fixed numbers

of utterances, but no indication of how much evidence might be required for statistical significance, how fast learning was occurring, how susceptible the associations were to changing on new evidence, or what kinds of words might be harder to learn than others. These experiments focused on the development of the strength of association between word and property over time. More importantly, my analysis included several equations derived from the chi-square equations that shed light on some of these questions.

5.3.2 Development of associations over time: Methods

The first experiment consisted of the robot watching myself and another subject toss the ball back and forth, saying either "I got the ball," "You got the ball," "I got it," or "You got it" for 50 utterances. The robotic setup used the color blob detector to find the ball, and one face detector to find faces (see Chapter 4). The facing decision algorithm described in Chapter 4 was not yet implemented; speakers were assumed to be facing each other in this experiment. Microphones were 30 cm apart, and 40 cm from each speaker. Audio localization continued to be based on time-of-flight for the first sound louder than a threshold, rather than overall loudness over the whole utterance.

The word "got" was interpreted as a signal that all the other words in the sentence referred to the person who was closer to the ball, a property called HASBALL. The other properties were SPEAKER, ADDRESSEE, LPROP, and RPROP; these last two corresponded to "person on the left" and "person on the right," replacing the "static variables" of Gold and Scassellati (2006d) with one variable uniquely true for each participant. Being on the left or right side of the robot's visual field was used as a proxy for personal identification, since face recognition was beyond the scope of the experiment.

In the second experiment, the robot was unable to tell when anybody but itself possessed the ball, and did not sense the property ADDRESSEE. Though this experiment was meant to simulate blindness, the robot had no tactile sensors, so proximity had to be sensed by requiring the ball's size in the visual field to pass a large threshold. (Unlike the first experiment, the two people could

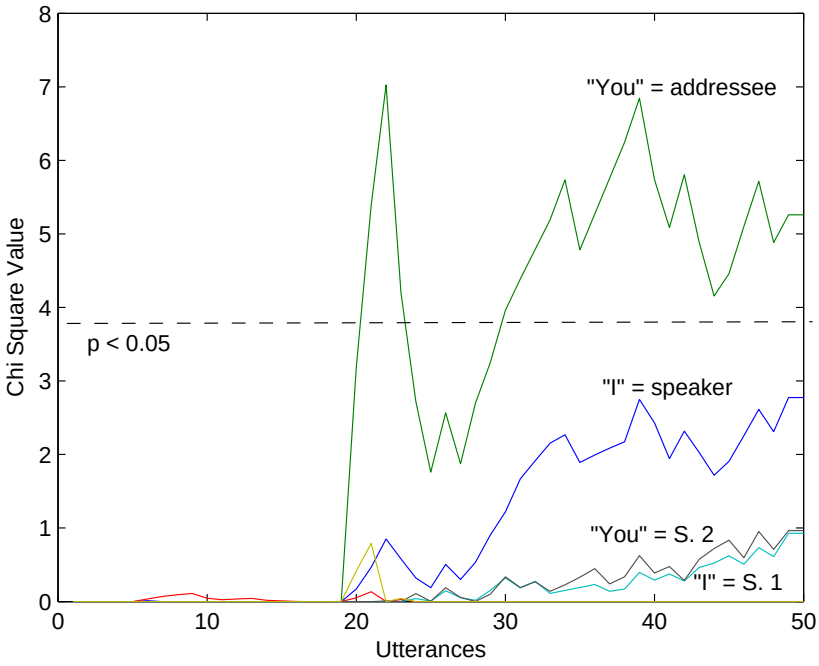

Figure 5.7: Chi-square values for the word-property associations in Experiment 1 of Gold and Scassellati (2006e). The large jump at utterance 20 marks the first time the system heard the word "you." The chi-square value for statistical significance (3.84) is given for reference, though meaning is attributed to a word based on its highest chi-square value. The second-best hypotheses, indicating that "I" and "you" are the names of the two subjects, are also shown.

now pass the ball to the robot.) The names of the participants in the experiment were added to the robot's vocabulary, as referring to LPROP, RPROP, or NPROP, a property uniquely true of the robot; this was to prevent "You got the ball" from being the only interpretable utterance.

5.3.3 Development of associations over time: Results

As Figure 5.7 shows, the unblinded system showed clear long-term trends toward learning that "I" refers to the speaker, and "you" to the person being addressed. These chi-square values increased steadily over time, while the competing hypotheses that they were names for the individuals rose at a much slower rate.

For the first 19 utterances, the speakers had only used the phrases "I got the ball" and "I got it." This produced zeros in the denominator for at least one chi-square term in all of the relevant

word-property associations. The system had no reason to assign any meaning to "I," because it was a part of every sentence, and no reason to assign any meaning to "you," because it was a part of none.

When "you got the ball" was finally spoken at utterance 20, the chi-square value for the association between "you" and the ADDRESSEE property spiked. This was because both "you" and reference to the addressee were rare events so far, making their coincidence highly significant. On the other hand, the usage of "I" and reference to the speaker were both still very common events, and so little could still be concluded from their common occurrence.

This points to a rather surprising fact about chi-square word-object associations: the more common properties generate *less* confidence. Because "you got the ball" remained less common than "I got the ball" (and reference to the addressee less common than reference to the speaker), the confidence in the I/speaker association remained lower than the confidence in the you/addressee association over the course of the experiment.

In the second experiment, association of "you" with the robot's identity reached significance in about twenty utterances when the robot was blinded (Fig. 5.8). This chi-square value would hold equally well for any property that was always true of the robot when "you" was spoken, but never detected about other agents. Thus, if the system had been able to tell that it was the addressee when it was being addressed, the chi-square value for associating "you" with *addressee* would have been the same as that for the association of "you" with the robot's identity. To distinguish between the two hypotheses, the robot would then have needed to employ some other criterion besides strength of association. "I" was not associated with any property in the blinded case because the system had no way of determining the referent of the sentence "I got the ball" when it did not possess the ball.

5.3.4 Analysis of the Behavior of the Chi-Square Method

The expected behavior of the system over time can be calculated through the following analysis, which first appeared in Gold and Scassellati (2006e).

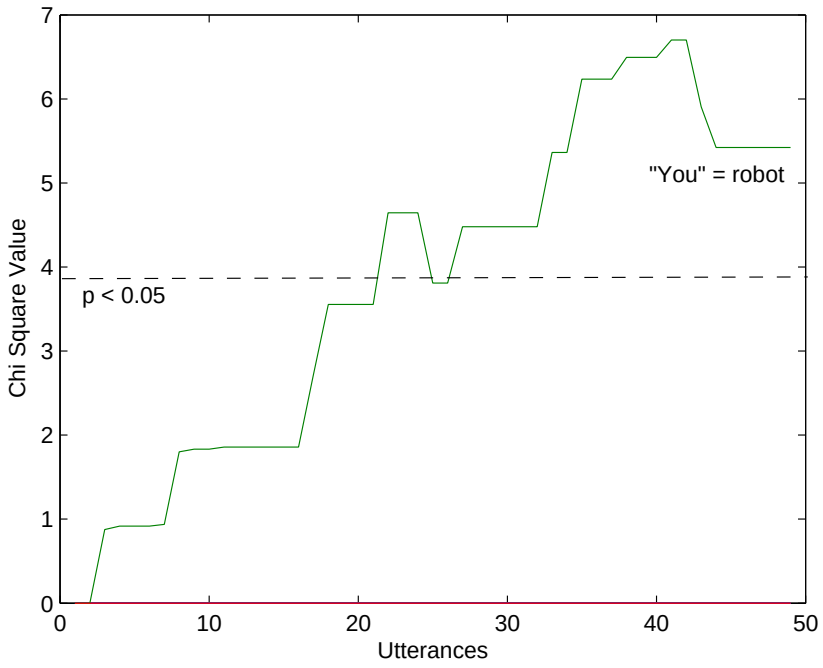

Figure 5.8: Chi-square values for the "you = robot" hypothesis when the system was blinded, from Gold and Scassellati (2006e). No other association achieved a valid chi-square value, though this value would hold equally for any other property that was uniquely and consistently true of the robot.

Suppose word W always refers to agents with property X, and for the moment assume that there is no error in hearing the word or perceiving the property. Let w be the number of times W has been heard to refer to an agent with property X, and let p be the observed frequency ($0 < p < 1$) with which property X is true of a referent regardless of what words are heard. Let C be the total number of words that the system hears, and assume $w \ll C$ so that the contribution from irrelevant words in the absence of the property is small. Then it can be shown that

$$\chi^2 \approx w(1/p - p) \tag{5.2}$$

The derivation requires the approximation $(Cp - w)^2/p(C - w) \approx Cp - w$ in the chi-square term corresponding to the case of ($\neg$word $\wedge$ property).

The chi-square value thus increases linearly with the number of times the target word is heard, but inversely with the frequency with which the property is observed to be true of a referent. This is more or less what one would hope to find: experience with a word increases certainty about its meaning, but very common properties are less interesting and disfavored as potential meanings.

It is also possible to derive the expected effects of sensory error on the chi-square values. If ϵ is the rate at which occurrences of the event (word $\wedge$ property) are mistakenly interpreted as the event (word $\wedge\neg$property), then the analogous assumptions to the errorless case result in the expression:

$$\chi^2 \approx w(\frac{\epsilon^2}{1-p} + \frac{(1-\epsilon)^2}{p} - p - \epsilon) \tag{5.3}$$

Here, w is the true count of the number of times the word was used in conjunction with an agent bearing the correct property; the agent's count is actually $(1-\epsilon)w$. The reader can verify that when $\epsilon = 0$, equation 5.3 reduces to the errorless case of equation 5.2.

The dominant effect of increasing the error rate ϵ is in the term $(1-\epsilon)^2/p$, where ϵ has an effect inversely proportional to p. This partly explains why the decline in chi-square is so great due to error during the first few utterances of "you"; not only is ϵ effectively greater because of the small

value of w, but the addressee property is uncommon, amplifying the effect of error. Asymptotically, however, the error merely changes the learning rate by a constant factor, leaving the rankings of word-property associations unchanged.

It has been suggested in the word-word collocation literature that chi-square results should be considered untrustworthy unless the expected values in each square of the chi-square table are at least 5 (Manning and Schütze, 1999b). In this case, this heuristic results in the rule that judgment should be withheld until the following condition occurs:

$$w > \max(5/p, 5/(1-p)) \tag{5.4}$$

In the first experiment of Gold and Scassellati (2006e), this would have resulted in the system withholding judgment on the word "you" until roughly utterance 32, thus avoiding the awkwardness of revoking and then reinstating confidence in the association with the addressee property. More balanced occurrence of the "I" and "you" cases, so that $p = 0.5$, would have resulted in confidence much earlier, around utterance 10.

Though these derivations were probably the most valuable contribution from Gold and Scassellati (2006e), they came about as I was attempting to explain the data from the first experiment, which made the experiment valuable despite the fact that it was only run once and did not result in significance for the definition of "you." (I believe I was still being hampered by bad sound localization at that point, and some issues in syncing the speech recognition with the visual data.)

The second experiment, by contrast, was not much of an experiment at all – there was really no way that the robot could have learned either correct definition, by the very setup, and so the outcome was obvious. It was more of a demonstration to prove the point that word meanings are difficult to learn without the relevant sensory input, and thus that theories explaining blind children's pronoun reversal as partial autism are spurious. It remains a valid point, even if the experiment was something of a one-sided oversimplification.

5.4 Integration with Self-Recognition

Though the experiments described above showed that the robot could learn the correct meanings of "I" and "you" in an abstract sense, it is useful to demonstrate that it could use these words appropriately, given a means of recognizing itself. In Gold and Scassellati (2006c), I integrated the chi-square based word learning system with a method for self-recognition devised by Michel et al. (2004), and also added the ability to respond appropriately to the command, "Say who got the ball."

The method for self-recognition was introduced in Michel et al. (2004) and developed somewhat in Gold and Scassellati (2006b): from its own random movements, the robot learned in what window of time to expect motion after it had sent a motor command. Any movement in the robot's visual field that started within this window, which typically began at about 500ms and ended around 1s, was labeled as "self." This allowed the robot to identify its mirror image as "self" as easily as it identified its physical, unreflected arm. The method's inability to integrate multiple pieces of evidence over time tended to result in quite a few false positives – the method I introduced in Gold and Scassellati (2007b) worked much better, using likelihood calculations on Markov models instead of time windows for motion-based self-recognition – but that fact does not matter much for the exposition here.

The natural language interface worked well enough in the simple setting: a command of "say who got the ball" would get the robot to produce a word that referred to a property of the person for whom *hasBall* was true, followed by "got the ball." To answer correctly, the robot had to equate its self-representation (i.e., list of Boolean attributes) with the boxes in the visual field labeled as "self" and set *Speaker* and *Addressee* appropriately for all parties before answering.

The robot used a face detector to find the experimenter, the color blob detector to find the yellow ball, and a module that found regions of motion in the visual field (Michel et al., 2004) to find motion that coincided with its motor movements. The motion module found pixels that differed in luminance from their values in the previous frame by more than 18/255 and clustered them together into bounding boxes using a region-growing technique identical to that used for the color module. These boxes were then tracked over time using a system designed by Andrew Lovett (Lovett and

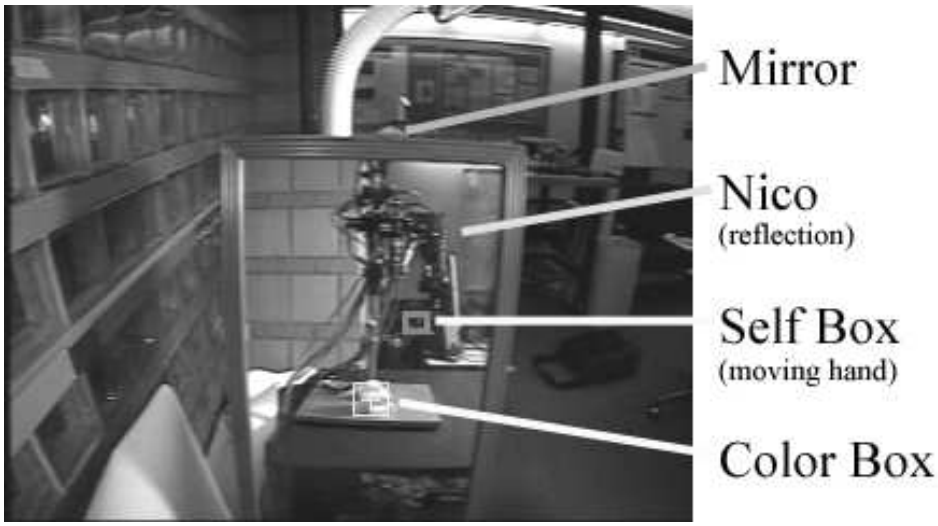

Figure 5.9: A view of the setup of Gold and Scassellati (2006c), taken from one of Nico's cameras. The robot can see its reflection in the mirror, center. Superimposed on the image is the bounding box produced by the "color" module, which has found the mirror reflection of the bright yellow ball, and the box produced by the "self-motion" module, which has found the reflected motion caused by Nico's arm movement.

Scassellati, 2004). If a motion box first appeared within roughly 300-800ms of a motor command to the arm – a time window learned in a previous self-recognition experiment (Gold and Scassellati, 2006b) – the box was labeled as "self." Sample output from the various systems is shown in Figure 5.9.

The following grammar was used for Sphinx speech recognition:

```
<utterance> = <subj> <verb> <obj>

<subj> = I | you | Alice | Bob | (say who)

<verb> = got | caught

<obj> = it | the ball
```

The system used the associations of "I" and "you" with *speaker* and *addressee* and "got" with *hasBall* generated in Gold and Scassellati (2006e). On hearing a sentence that began with "say who," the robot would set the *speaker* and *addressee* variables of that person and itself appropriately, then answer with a word for the entity for whom *hasBall* was true. (Distance in the two-dimensional

image was used to find which entity was closer to the ball, since depth perception had not been implemented.)

Two tests of the system were run. In one, the robot moved its arm every twenty seconds, while in the other, the robot moved only once to ascertain its position. Under both setups, the robot was told "Say who got the ball" forty times, twenty for each case of "I" or "you" being the correct answer. When the robot moved every 20 seconds, it correctly said who had the ball 27 out of 40 times: 16 out of 20 when the correct answer was "I," and 11 out of 20 when the correct answer was "you." When the robot moved only once at the beginning of the experiment, reducing its susceptibility to incorrectly labeling the experimenter as itself, it answered correctly 38 out of 40 times. Thus, the errors lay mostly with the false positives from the early self-recognition system, and not with its integration with the "I" and "you" learning per se.

Like the pronoun reversal "experiment" of Section 5.3, this was more of a demonstration than a true experiment; the only possibly surprising thing was how susceptible to false positives the self-recognition method was at the time. Nevertheless, this demonstration was a nice way of concretely showing that the system had learned something usable from both systems.

5.5 Summary

The chi-square based learning system represented a first pass at word learning that, though it was not as linguistically sophisticated as the system to follow, contained some of the core ideas that would later become the TWIG system. The idea of using sentence context to find the reference of unknown words was introduced in Gold and Scassellati (2006d). The idea of using chi square tests for significance to find word-property pairings was introduced in Gold and Scassellati (2006d) and its performance over time was analyzed in Gold and Scassellati (2006e). The applications of question-answering and relating the words to the robot's mirror image demonstrated that the robot's knowledge of the meanings of these words was usable, rather than being purely abstract (Gold and

Scassellati, 2006c). And finally, most of the sensory systems that would be used in the TWIG system were implemented in this robotic system.

These papers also suggested some of the ways in which the phenomenon of pronoun reversal, common to autistic, blind, and particularly precocious children, could occur. Gold and Scassellati (2006d) showed that difficulty in understanding the word "want" could delay the acquisition of the word "you" (Gold and Scassellati, 2006d). Gold and Scassellati (2006e) showed that simply being blind and unable to determine where people were looking could result in an overly restrictive definition of "you" and no definition at all for "I" (Gold and Scassellati, 2006e). Though these experiments were arguably oversimplifications of the environments of real children, they nevertheless were the first quantitative models of pronoun reversal in real environments. In the autistic case, misunderstanding "want" fits entirely with the "mindblindness" theory of autism (Baron-Cohen, 1995), while for the blind children, my experiment showed that blind childrens' pronoun reversal may be simply a direct effect of their blindness, rather than being caused by a deficiency in "theory of mind" or perspective-taking that is itself caused by blindness. This latter result is fairly important, as several researchers had proposed theories that possibly underestimated blind childrens' mental abilities, citing pronoun reversal as their evidence (Andersen et al., 1984; Brown et al., 1997; Fraiberg and Adelson, 1977).

Another important result was the simulation performed on a transcript from the CHILDES database, using real dialogue and a simulated complicated environment (Gold and Scassellati, 2006d). This result showed that the system was feasible even for real, unstaged dialogue, and for large numbers of variables – a good preliminary step before implementing a large-scale, real-world system. This was to be my last experiment with data generated from real parent-child interactions, as my research became more about developing a working robotic system and less about modeling human children's learning.

Despite these contributions, the most important developments were still to come. In Chapter 6, I shall describe how the system was extended to use logical semantics and sentence parsing. Instead of assuming one "referent" for the entire sentence, the system would now be able to find

reference for each part of the sentence, and logically solve for the reference of new words. In Chapter 7, I will describe a refinement that allowed the system to learn definitions involving conjunction, disjunction, and thresholds on continuous values. This would allow the system to learn complex meanings assembled from various sensory inputs and values, rather than requiring definitions to be drawn from existing concepts. These important differences would result in a system that was much better equipped to learn a variety of words besides "I" and "you" and use them in well-formed sentences.

Chapter 6

TWIG Part I: Using Formal Semantics to Understand and Create Sentences

The previous chapter introduced a simple setup by which the robot could infer the meanings of "I" and "you," but its treatment of language was quite basic. A sentence was assumed to have a single referent, and words were assumed to refer to true properties of that referent. Such a treatment of language was not very general, as it could not learn words for relations between things; nor could it transform language into a representation that could be judged true or false. This chapter shall describe how I added parsing and formal semantics to the system, so that it could parse full sentences.[1]

I dubbed this new system TWIG, for "Transportable Word Intension Generator": "transportable" because it should ideally work with any robot that produces predicate-logic representations of its world, and "word intension generator" because I was now fully aware of the extension/intension distinction I was making in meaning, and I realized that the idea of an "intension" captured the real essence of what I wanted the system to learn (see Chapter 3). More than a simple association, an *intension* allows the system to judge the truth or falsity of a sentence that contains the word. Of course, in some ways the associations of "I" and "you" are at least as interesting as their intensions,

[1]This chapter borrows much of its material from Gold and Scassellati (2007a).

but the criteria for success are much more nebulous there. Formal intension-based meanings would allow the system to not only understand and produce full sentences, but would also allow the system to make inferences about reference in a much more principled way. For example, the system would actually understand that "got" did not mean `hasBall`, but that "got the ball" meant `got(X, b)` $\wedge$ `ball(b)` for whatever X was the subject of the sentence.

The mechanism by which intensions were learned, and the space of possible intensions, were not fully developed yet in the experiments described in this chapter. The space of possible meanings was limited to variants on relations and predicates already present in the system, and learning was still accomplished by finding the highest chi-square value for a word-property pairing (see Chapter 5). In the next chapter, I will describe a more sophisticated learning mechanism for the intensions that allows definitions to include conjunctions, negations, and thresholds on continuous values.

Nevertheless, this system was sufficient to learn "I" and "you" in a more principled manner than before, and would also achieve two things that the previous system had not: it would use its new words to learn still more words, and those new words would refer to a relation. More specifically, once the system had learned "I" and "you," it could use evidence such as "I am Kevin" and "You are Eli" to infer that "am" and "are" referred to the identity relation.

6.1 Parsing and Finding the Extension

TWIG uses Prolog to parse the sentence into logical form and infer the extensions of any new words. The TWIG system adapts the following definite-clause grammar from Pereira and Shieber (1987):

```
s(S, W) --> np(VP^S, W), vp(VP, W).
np((E^S)^S, W) --> pn(E, W).
np(NP, W) --> det(N2^NP, W), n(N1).
vp(X^S, W) --> tv(X^IV), np(IV^S, W).
vp(IV, _W) --> iv(IV).
```

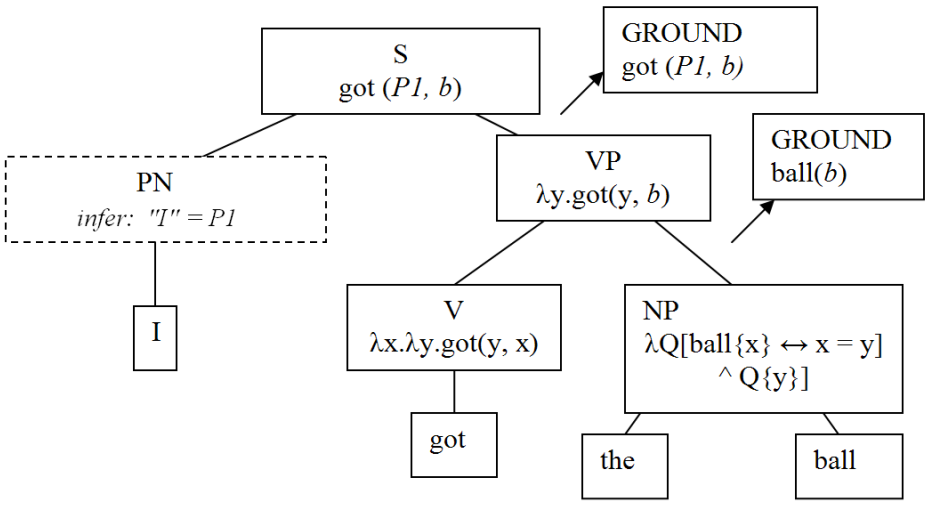

Figure 6.1: Parsing a sentence with an undefined word, "I." The parse partially succeeds on the right, and the system finds that the whole sentence can be grounded if "I" refers to person *P1*, who has the ball. The missing definition, that "I" refers to whoever is speaking, can only be learned over time.

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. Thus, the grammar allows parsing of simple sentences of the form *Subject Verb* or *Subject Verb Object*. W is a pointer to a list of predicates indicating the state of the world, which must be passed around 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. This notation is drawn from formal semantics (see Chapter 3), and represents the idea that a word corresponds to a fragment of a logical formula that can be combined with similar formulas to produce a statement that has a truth value. 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 definite-clause grammars, and the arguments of the functions are substituted as the parse provides them: see Figure 6.1.

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

```
tv(LF) --> [TV], {tv(TV, LF)}.
tv(Word, X^Y^pred([Word, X, Y])).
iv(LF) --> [IV], {iv(IV, LF)}.
iv(Word, X^pred([Word, X])).
n(LF) --> [N], {n(N, LF)}.
n(Word, X^pred([Word, X])).
```

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^S1)^(X^S2)^S2) :- contains(W, S1).
pn(E, W) --> [PN], {pn(PN, W, E)}.
pn(PN, W, X) :- contains(W, pred([PN, X])).
```

The predicate `contains(W, X)` is true if the world W includes the fact X. On parsing a proper noun or definite article, the `contains` clause effects a search for the extension, and the symbol for that extension 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. (In the case of multiple possible extensions, the system chooses one arbitrarily.)

If the robot knows enough to understand the sentence S, the end result when the robot hears a sentence is that it is transformed into either the form `pred[P, X]` or the form `pred[P, X, Y]`, where

X and Y are symbols that correspond directly to known objects and P is a sensory predicate. If the robot's world W contains this fact as well, nothing further happens. If W does not contain $P(X, Y)$, but there is some fact in W that contains P, the sentence is understood as new information, and is added to the knowledge base.

If the parse fails, 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 $pred(Word, Object)$, where $Word$ is the new word and $Object$ is the object to which it refers.

Figure 6.1 illustrates how this works. Suppose the robot hears the statement "I got the ball." It does not know who "I" 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 the word "I." It does, however, know that "got the ball" parses to $\lambda Y.has(Y, b)$. On retrying with the free variable, the robot finds that hypothesizing $I(a)$ allows it to match the sentence to $got(a, b)$, a fact it already knows. Thus, "I" is assumed to refer to a: the system has successfully inferred the extension.

6.2 Finding the Intension

This section describes how the system learned intensions for the experiments in this chapter.

On inferring an extension for a word, the system next formed hypotheses about the intension of the word. The system searched its knowledge about the world W for all facts about the extension. This included single-argument predicates as well as relations: for example, the facts retrieved if the ball b were the extension could include both $ball(b)$ and $got(a, b)$.

A new intension consisted of two parts: a predicate P and an argument number i. The `define` operator has the following semantics:

$$\text{define}(w, P, i) \iff P(\underbrace{\ldots}_{i-1}, o, \ldots) \models w(o) \tag{6.1}$$

Let the shorthand $[[w]] = P@i$ be equivalent to define(w, P, i). (The bracket notation is adapted from Dowty et al. (1981).) In the case of single-place predicates, this has the intuitive definition of words being defined by already existing single-place predicates – for example, [[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, it is possible to define [[I]] = $tells@1$ and [[you]] = $tells@2$, corresponding to the notion that "I" is the speaker and "you" is the person being addressed.

Once these possible intensions had been generated, the system used the chi-square comparison method described in Chapter 5 to decide which intension-word pairing was most significant (i.e., least likely to have occurred due to chance). For each possible definition Φ_{ip}, corresponding to predicate p and predicate argument i, the system counted the number of times ϕ_{ip} that any word's extension had fit the definition. For each word W_j, the system counted the number of times w_j the word had been used, and the number of times it had been used for each predicate-place pair, w_{ijp}. In addition, the system tracked 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.

As before, chi-square values could be high for word-definition pairs for which the word appeared *less* often than expected, which is not generally helpful for a word definition. Thus, the cases where $w_{ijp} < w_j\phi_{ip}/\sigma$ were excluded. Otherwise, the system estimated the best intension for a word to be the definition with the highest chi-square value of all that word's definition pairs.

If the program was halted, the word, property, and collocation counts were written to a file. This data was sufficient to resume learning where TWIG had left off. In addition, on restarting, TWIG asserted `define`(w, p, i) for any word-definition pair that was higher than all other chi-square values for the same word and exceeded a threshold of significance of $p < 0.05$ ($\chi^2 > 3.84$). This definition was then available for parsing sentences normally or making inferences about other words.

6.3 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 were learned in almost the same way. On encountering a new transitive verb v, the system's parse would fail the first time. On the second pass, the free variable A was appended to the world description W, and on reparsing it would bind with $pred([v, s, o)]$, where s and o were the extensions of the subject and object of the sentence. The hypothesized definition would then be of the form `define`(v, p, i, j), where i and j were both places of a predicate that relates s to o. For example, "addresses" might be defined with `define(addresses, tells, 1, 2)` to treat it as a synonym to "tells," while "listens" could be (loosely) defined as `define(listens, tells, 2, 1)`. Counts and chi-square tests proceeded as normal for each such definition found.

6.4 Robotic Implementation

The robot used the face detectors, color blob detector, sound localization, and speech recognition described in Chapter 4. The input from the robot's sensory systems was then 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, I 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 possessed a symbol n for itself. Each face and the ball received a predicate that uniquely identified it; I 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 $has(P, b)$ was true, where P was the symbol for that person.

On detecting speech, the audio system produced the predicate $tells(X, Y, Z)$, where X is the speaker, Y is the person being addressed, and Z is 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. (This decision of facing was made

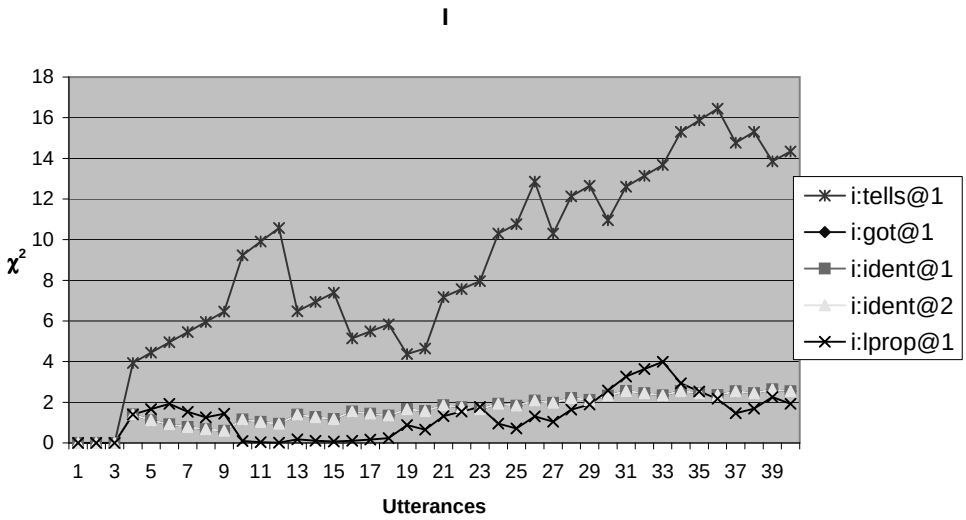

Figure 6.2: Sample chi-square values over time for one subject (M.D.) in experiment 1, for the word "I." The correct definition is tells@1, corresponding to the speaker.

using the forward-algorithm-based face decision process described in Chapter 4.)

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

6.5 Experiment 1: "I" and "You"

For our first experiment, I followed the experimental setup of Gold and Scassellati (2006e) (described in the previous chapter) using the new Prolog-based TWIG system to learn the words "I" and "you." Two people passed a bright yellow ball back and forth in front of the robot, using 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 to look at the robot when saying the other person's name. (I was always one of these speakers, as pairs of subjects left to their own devices tended to speak and act too quickly for the speech recognition to catch up.) All of the words were contained in a small context-free grammar 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.

Subject	A	B	C
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 6.1: 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.

6.5.1 Results

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 6.2 shows the progress of the definition of "I" for subject C, while Table 6.1 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. The high error rates also caused a high variance among the subjects in when the correct hypothesis was achieved.

6.6 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

Subject	A	B	C
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 6.2: Comparison of "am" and "are" learning with different sensory error rates for subject facing, sound localization, and speech recognition.

[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.

6.6.1 Results

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. Neither had a particularly high error rate, and the slow learning of "are" for subjects A and C occurred mostly because they often spoke this utterance immediately after "I am [name]," while the speech recognition software was still processing the first utterance, making "you are [name]" the more rare utterance in the data. Table 6.2 compares the results and error rates across subjects. (Facing errors were less common in Experiment 2 because the speakers consistently faced each other.)

6.7 Discussion

The ability to parse sentences into logical forms is a huge advance over the approaches described in the previous chapter. There, the results depended on some rather large assumptions built into the experimental setup, such as the fact that "got" was always used in the context of having the

ball, or the fact that each sentence only included one referent of real interest. Without grammar, the semantics of the previous chapter could only convey vague associations, instead of propositions about the world. The addition of formal semantics greatly increased the validity of the claim that the system was learning the meanings, or intensions, of words.

The power of this formal semantics was showcased in the second experiment, in which the system learned that "am" and "are" refer to the identity relation. The previous system could not have learned these meanings, because no reasonable representation could include the identity relation as a single boolean property of an object. The ability to learn the meaning of a relation was new to TWIG.

The ability to use the newly learned word meanings to learn more words could have been implemented in the previous, simpler semantics – but it was not, and the demonstration of this ability in TWIG was important for conveying the idea that this system could bootstrap its way into larger and larger vocabularies. I have argued elsewhere (Gold and Scassellati, 2007a) that it is better to focus in a word-learning system on the "inductive step" of using language to learn more language, rather than the "base case" of learning first words. A starting vocabulary is easily programmed into the system; it is the process of going from a size n to size $n + 1$ vocabulary that deserves the focus of research. As I demonstrated in these experiments, it only takes a few definitions to begin to grow the system's vocabulary.

The intension representations and learning mechanism used here were not too bad, and it is worth pointing out some of the strengths and weaknesses of this approach before moving on to the "definition trees" of the next chapter. Though the choice of predicates was limited to the existing predicates of the system, the ability to treat any slot of an n-ary predicate as a potential definition has more versatility than there might appear at first glance. Consider a predicate representation of a car that includes as an argument where the car is parked; hypothetically, this kind of system could learn that anything occupying this slot is a "garage." More generally, if predicates represent frames (Minsky, 1974) for common actions or scripts, this kind of representation would enable the system

to learn names for each slot of the predicate, or for relations between the various slots.

This representation and learning process for intensions also does not require that words be mutually exclusive in their definitions, unlike the definition trees of the next chapter; this will be discussed in more detail in the next chapter. Overall, however, the inability to learn conjunctions or predicates involving numerical thresholds severely limits the intensional representation of the present chapter.

One weakness of the present chapter's experiments is that there were not very many possible predicates available for definitions. In the second experiment, the only predicates with more than one place were `got\2`, `identity\2`, and `tells\3`, which did not leave very many possible interpretations for "am" and "are" (though variants on `tells` were possible when speech localization failed). Even though the system could use the new words it had learned as predicates, its inability to form conjunctions and disjunctions meant that these new predicates would always be highly constrained by the predicates that the system started with. The experiments in the next chapter significantly improve this situation.

Another weakness of these experiments is that they did not demonstrate the robot's ability to generate sentences about its environment, though this would have been a natural extension of the new formal semantics. This was implemented with the definition-tree based intensions of the next chapter, but it could have been implemented with these simpler intensions as well.

The addition of formal semantics to unsupervised learning put TWIG in an interesting position, as previously robotic word-learning systems had avoided full-sentence parsing and semantics, while word-learning systems that did use such semantics were not well-suited to implementation on real robots (see Chapter 2). The fact that TWIG combined these approaches made it promising not only as a system for learning deictic pronouns, but for learning words in general. Nevertheless, the case of pronouns would play a large role in motivating the innovation to be described in the next chapter: that of using "definition trees" for intensions.

Chapter 7

TWIG Part II: Finding the Intension with Definition Trees

"So, how would you learn the meaning of 'he'?" This question, which Drew McDermott had asked after a talk about some of my earlier results, motivated the approach to intension that I will present in the present chapter. My previous methods had required that definitions be constructed from true, single-term predicates. It seemed somewhat perverse to construct a single predicate that meant "male, but not the speaker, and not the person being addressed." Clearly, any reasonable system of intension learning should be able to construct definitions that include conjunction and negation. But how could I do this in a manner that would plausibly scale? Testing an exponential number of conjunctions as if they were single predicates did not seem reasonable.

A solution came to me when I realized that the meanings of the pronouns I was interested in could be arranged into a tree structure, in which the interior nodes were predicates that could be either true or false (Figure 7.1). What if such trees could be constructed using a standard decision tree creation algorithm (Quinlan, 1986)? The definitions could then be read off the tree by following a path from a leaf containing a word back to the root. This solution was elegant enough that I was finally satisfied that I had created a word-learning mechanism that was not *particular* to the words

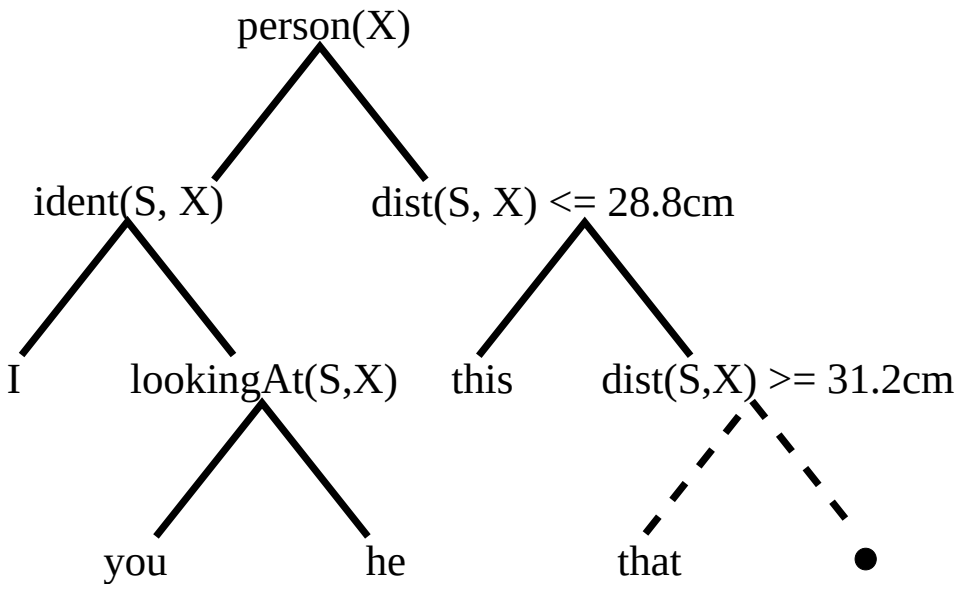

Figure 7.1: An example of a definition tree, taken from the experimental results to be described. Branches to the left indicate that a predicate is satisfied.

"I" and "you," but was powerful enough to *include* them.

As an additional advance over the previous methods, I introduced the ability to deal with numerical values, reducing the system's need for predicates tailored to the learning problem at hand. This allowed the system to create its own threshold on distance to define the difference between "this" and "that," rather than requiring a Boolean predicate `close(X,Y)` to exist prior to learning.

In this chapter, I will first describe the structure of definition trees in more detail. I will then describe the details of how they are learned, followed by a specific experiment in which meanings for the words "I," "you," "he" "this," "that," "above," "below," and "near" were learned. After that, I will present the culmination of this thesis: I show that the definitions learned using this method, combined with the extension-finding of the previous chapter, allowed the robot to form new sentences about its environment that correctly made use of the new words; and that TWIG's combination of extension-finding and definition trees resulted in more correct utterances and fewer incorrect utterances about various scenes than variants that lacked these features.

7.1 The Interpretation of Definition Trees

Definition trees[1] reconstruct the speaker's decision process in choosing a word for a referent or a relation. They are essentially decision trees, but they are built for use in both comprehension and production, rather than simply being used as classifiers. The words are stored at the leaves, and a word's definition is given by the path from the root to the word's leaf. The interior nodes can be decisions about the referent's properties itself, or relations to other objects or people, particularly the speaker. When TWIG learns the meanings of new words, it does so by constructing definition trees that include the new words. The structures of these trees implicitly define the words.

Figure 7.1 shows an example of a definition tree. A branch to the left after a decision indicates that the predicate is satisfied, while the right branch indicates that it is not. Each decision consists of attempting to satisfy a logical predicate with at most one threshold on a numerical argument to the predicate. We follow the additional convention that S always refers to the speaker, X always refers to a referent, Y is an optional second referent for the purpose of defining relations, and V is a numerical threshold. For example, one decision might be `dist(S,X,V)` & $V \leq 28.8$, indicating whether the distance between speaker S and referent X is less than 28.8 cm. (We will sometimes use the shorthand `dist(X,Y) <= V`, or omit mention of the threshold entirely if the attribute is boolean.) In choosing a word to describe X, the system would decide whether speaker S and X satisfy this predicate and threshold. If the predicate is satisfied, the path on the left is followed; if they do not, the path on the right. This process continues until a leaf is reached, at which point the most common word at the leaf would be chosen.

The structure of a definition tree implies the logical definitions of the words it contains. Each word is defined as the conjunction of the predicates that lie on the path from the root to the leaf, with a predicate negated if the "no" path is followed. For example, the tree in Figure 7.1 implies that the logical definition of "you" is

[1]This chapter covers very similar material to Gold et al. (2007), which called these structures "word trees." I have since decided that the term "definition trees" is a less vague word for these structures, since "word trees" could be confused with parse trees.

```
you(X) :- person(X) & \+ident(S,X) & lookingAt(S,X)
```

indicating that "you" is a person who is not the speaker (`\+` is the Prolog negation operator), but whom the speaker is looking at. (I shall continue to use `ident` to refer to the identity relation, which holds true for each object only with itself.)

If the rightmost path of the tree is followed to its end, the system realizes that it does not know enough to describe the item in question; this is indicated by a dot in Figure 7.1. The decision just before this final check is depicted in dotted lines to indicate that it is constructed in a slightly different manner from the other decisions. The system there is no longer contrasting words against other words, but choosing whether it knows enough about the referent to say anything at all.

The goal of the system is to construct such trees automatically from data, as the example in Figure 7.1 was. We turn now to the algorithm for constructing these trees.

7.2 Constructing Definition Trees From Data

Definition trees are constructed and updated using the output of the extension finding module. The mechanism is essentially a variant on Quinlan's ID3 algorithm (Quinlan, 1986): the available evidence is split into two groups based on the decision about the referent that is most informative to word choice, with one group satisfying the condition and the other not. This process then occurs recursively until no further decisions are statistically significant. The tree can be updated online, though the "batch" case shall be described first because it is simpler, and in some cases it must be called as a subroutine to the online version.

The output from the extension finder for a given utterance, and thus the input to the definition tree generator, is a 6-tuple $(W_i, T_i, X_i, Y_i, S_i, \Omega_i)$, where W_i is the new word, T_i is the inferred part of speech ("type"), X_i is the literal determined to be the referent, Y_i is an optional second referent (or null), S_i is the literal that was the speaker, and Ω_i is a list of predicates from the world that contain literals X_i, Y_i, or S_i as arguments. The second referent is non-null if the word was determined to

refer to a relationship between two referents, instead of to a particular referent.

TWIG generates a separate decision tree for each part of speech, where the part of speech is inferred from parsing. The use of separate trees is necessary because during language generation, the system should not attempt to use a noun as an intransitive verb, for instance, even though both have the same basic logical form and arguments. The rest of the exposition will assume we are dealing with the tree for a single part of speech.

Let t be an evidence 6-tuple; the following is the description of $D(t)$, the function that generates the set of possible decisions implied by the tuple. A *decision* here is a triple (P, V_0, σ), where P is a predicate with its arguments drawn exclusively from the set of variables $\{X, Y, S, V, \_\}$, V_0 is a threshold value, and σ is a sign indicating whether the direction of inequality for the threshold is $\geq$ or $\leq$. For each tuple, all instances of object X_i are replaced with the variable name X; all instances of object Y_i are replaced with variable name Y; instances of the speaker S_i are replaced with the variable S; and one numerical constant can be replaced with variable name V. All other arguments to the predicate are replaced with the anonymous variable "_", which can bind to anything and thus serves a similar role to existential quantification. The threshold V_0 becomes the constant value that was replaced with the variable V, or 1 if there was no constant value. Then, two decisions are added to the set: one for $\sigma =$ " $\leq$" and one for $\sigma =$ " $\geq$". If a single literal plays more than one role, by being both speaker and the referent, then all possible decisions that can be generated by these substitution rules are added to $D(t)$. For example, if Bob was both the speaker and the sole referent of the new word, the term `inchesTall(`*bob*`, 60)` would generate four decisions: two decisions for whether the speaker (S) was at least or at most 60 inches tall, and two decisions for whether the referent (X) was at least or at most 60 inches tall. $D(t)$ consists of the union of all such sets generated by each predicate in Ω_i.

The algorithm must decide which of these decisions is most informative as to word choice. For each decision, a $2 \times |W|$ table is maintained, where $|W|$ is the number of unique words. This table maintains for each word the number of times the decision was satisfied by the referents of the word,

and the number of times the decision was not satisfied. Note that this requires attempting to satisfy each decision with the variables X, Y, S and V bound to their new values for each tuple.

From this table, one can easily calculate the information gain from splitting on the decision. Let W be the set of all words w_i, and let $w_i \in W_d$ if it was used under circumstances when decision d was satisfied, and $w_i \in W_{\neg d}$ otherwise. Then

$$Gain(d) = H(W) - \frac{|W_d|}{|W|}H(W_d) - \frac{|W_{\neg d}|}{|W|}H(W_{\neg d}) \tag{7.1}$$

where

$$H(W) = \sum_{i:w_i \in W} -\frac{|w_i|}{|W|}\log\frac{|w_i|}{|W|} \tag{7.2}$$

Readers may recognize $H(W)$ as the *entropy* of W, characterizing the average amount of information in a single word. $Gain(d)$ is the expected reduction in entropy on learning the truth or falsity of decision d. The decision with the most information gain is thus the fact about the referent that maximally reduces the "surprise" about the choice of word (Hamming, 1986). This is the criterion used by Quinlan for the original ID3 decision tree algorithm (Quinlan, 1986). If two decisions are tied for informativeness, TWIG breaks the tie in favor of non-numerical predicates, thus penalizing the numerical predicates for complexity.

The maximally informative decision is added to the tree, so long as the decision and word choice are determined to be highly unlikely to be independent. A chi-square test can be computed using the same $2 \times |W|$ table to test for significance. TWIG uses Yates' continuity-corrected chi-square test (Yates, 1934):

$$\chi^2_{Yates} = \sum_{i=1}^{N} \frac{(|O_i - E_i| - .5)^2}{E_i} \tag{7.3}$$

where the sum is over the cells of the table, O_i is the observed value in that cell, and E_i is the expected value in that cell under the assumption of independence. (Yates' corrected chi is a compromise between the normal chi square test, which is misleading for small sample sizes, and Fisher's exact

test, which would take too long to compute for large numbers of decisions to be evaluated.)

Because the chi-square critical value depends on the number of degrees of freedom, which in turn depends on the number of unique words, a single critical value can't be used. Instead, the critical value τ is approximated using the following equations (Law and Kelton, 2000):

$$\tau = (1 - A + \sqrt{A}P^{-1}(1-\alpha))^3(|W| - 1) \tag{7.4}$$

$$A = 2/(9(|W| - 1)) \tag{7.5}$$

Here $P^{-1}(\phi)$ is the inverse normal distribution, $|W|$ is the number of distinct words in the table, and α is the desired significance level. In the experiments to be described, $p < 0.001$ was set to be the threshold for significance, so that even with hundreds of possible decisions, the chance of at least one of them reaching accidental significance remained less than $p < 0.05$.

With the decision added to the tree, the set of evidence tuples T is partitioned into two subsets – one subset consisting of the examples in which the decision was satisfied, and one in which the decision was not. These subsets are used at the left and right branches of the tree to generate subtrees, following the same procedure as outlined above. The process is then repeated recursively for each branch of the tree, until no more decisions can be added in which there is significant deviation in the data from chance (using the p ¡ 0.001 chi-square test). The most common word among all the tuples at a leaf is chosen as the "correct" word for that leaf. (Irrelevant words at a leaf are often the result of sensor noise or speech recognition error.)

Once the basic tree has been recursively built, one more operation needs to be performed, in order to make the rightmost branch meaningful. Unmodified, the rightmost path in a basic definition tree always corresponds to a failure to prove anything about a referent, since the right path must be taken when a value is unknown. Though there exist some words that describe items about which nothing is known (e.g., "it," "thing"), they do not exist for all parts of speech; for example, there is no "default" transitive verb that implies nothing about its referents. Thus, some meaningful words may

be assigned to this branch simply because there are no further distinctions to be made. Unmodified, this leaves the word mostly meaningless to the robot. Moreover, if the robot ever moved to a different environment or context, it would be constantly failing to prove anything, but then would mistakenly use its "default word" to describe every new situation.

For these reasons, the definition corresponding to the rightmost path is always augmented with the single non-negated decision that produces the highest information gain when contrasting the rightmost word with all other words in the tree. (This decision is represented in the decision tree diagrams with a pair of dotted lines.) The calculation uses all the evidence available at the root of the tree, but the $2 \times |W|$ table for each decision is collapsed into a 2×2 table, so that only the rightmost word vs. non-rightmost word decision is taken into account for informativeness. With this final check added, the system has a means of extracting slightly more information out of the evidence that falls into the rightmost classification, and it also has a way of determining whether it does not know enough to have a good word for a new situation.

7.3 Optimizations for Online Learning

The batch mode for tree generation is useful for reconstructing a tree from a datafile of evidence 6-tuples; most of the time, however, the tree is updated online in response to an utterance. If the $2 \times |W|$ tables for each decision (including decisions not chosen) are maintained at each node in the tree, a node update usually need only consist of updating these tables and added the few new decisions implied by the new tuple t. This update is performed first at the root, and then, if the root decision is unchanged, the new tuple is passed recursively down the tree to whichever branch it satisfies. The tree's structure only changes if, after updating the existing decision tables and adding the new decisions, the most informative decision at a node changes. In this case, the batch algorithm must be called for the entire subtree. While the worst-case running time for this online version is the same – theoretically, every update could change the most informative decision at the root, requiring

the whole tree to be rebuilt – in practice, most updates do not change any decision nodes, and the update is fast because it only updates tables down a single path to a leaf.

With these optimizations in place, a Java-based implementation running on a 3.4 GHz Pentium 4 could typically update a tree with 100-200 evidence tuples in 1-2 seconds, with a worst case of about eight seconds. The implementation used no optimizations besides those described here, and probably could have been further sped up with a better optimized proof mechanism for checking the truth or falsity of decisions. (See also "Complexity," below.)

Note that despite these optimizations for online performance, *the order in which evidence is received does not matter* in determining the final state of the definition tree. The inputs could be shuffled, and the final tree would remain the same, because the structure of tree $n-1$ has no effect on the structure on tree n; it merely affects the time necessary to compute the nth tree. The table of evidence stored at the root contains all the information necessary to rebuild the tree from scratch. If new data is received that makes different decision at the root more informative than the current root decision, the entire tree is rebuilt using this data; if a new decision is more informative at a different interior node, the subtree stemming from that node is rebuilt. The online optimizations are efficient under the assumption that these are rare occurences, but when the tree must be remade to better accomodate the data, it changes its structure to do so. TWIG makes use of its existing tree and data tables to avoid repeating calulations it has already made, but no decision is permanent. This also means that bad decisions introduced due to sensory error can be fixed or eliminated if the system is given more input.

7.4 Conversion to Prolog

After each definition tree update, the definition tree can be converted into Prolog, sent back to the extension generator, and asserted, so that the word meanings can be used for parsing immediately. For example, the meaning of "got" as implied by the definition tree in Figure 7.2b becomes the

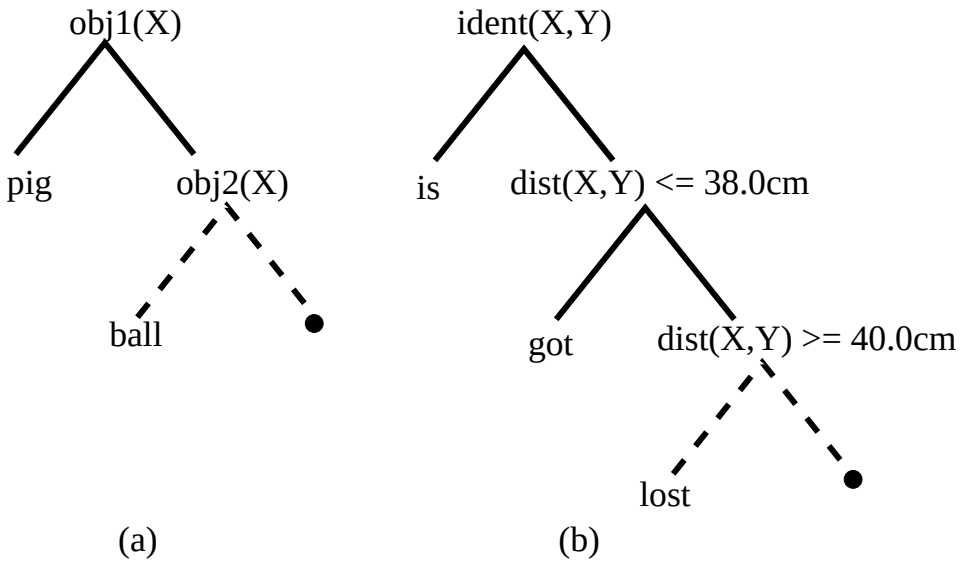

Figure 7.2: The definition trees available to the system at the start of the experiment of section 7.7 for (a) nouns and (b) transitive verbs.

following Prolog statement:

```
contains(W, pred([got, X, Y])) :-
member(W, pred([_P1, X])),
member(W, pred([_P2, Y])),
\+(contains(W, pred([ident, X, Y]))),
(contains(W, pred([dist, X, Y, V0])), V0 <= 38.0).
```

The "member" statements are necessary so that X and Y are instantiated before entering negated clauses. Prolog's negation-as-failure operator would otherwise dictate that `got(X,Y)` was false if `ident(X,Y)` held for *any* choice of X and Y in the environment – which is obviously not what is desired. If this definition had referred to the speaker, the clause `member(W, pred([says, S, _])` would perform the necessary binding.

A depth-first traversal of each definition tree can create an explicit list of valid words for each part of speech in the grammar. Since the language system treats internally defined predicates and words as nearly identical, explicit lists of defined words are necessary to prohibit the robot from using its internal sensory predicates as words.

7.5 Complexity

The time to build the tree is the time to evaluate a single decision set, multiplied by the number of nodes in the tree. This includes leaves, since decisions are evaluated at the leaves but ultimately discarded. The maximum number of leaves in the tree is $|T|$, the number of evidence tuples; the maximum number of total nodes in the tree is therefore still $O(|T|)$. The number of decisions to be evaluated at each node is at most $O(|P|V)$, where $|P|$ is the number of different predicates and V is the number of different values encountered for each predicate. (Despite the fact that there are several permutations of variables permissible for each predicate, that number is still bounded by a small constant.) Evaluating a decision requires in the worst case examining every fact in the environment, or $|E|$ steps. Thus, the complexity for the batch algorithm is $O(|T|^2V|P||E|)$. V typically will be linear in $|T||E|$, creating a running time of $O(|T|^3|P||E|^2)$; however, sampling a constant number of values for each predicate would bring the complexity down to $O(|T|^2|P||E|)$.

As mentioned previously, the worst case for the online version is a reorganization of the entire tree at each step, resulting in $O(|T|^2V|P||E|)$ steps for each new tuple. However, the more common case of an update that does not change the tree structure requires only $O(d|E|(|P|+|T|))$ operations to update, where d is the depth of the tree; the two added terms come from the time to update existing decisions at a node, $O(|E||P|)$, and the time to generate and evaluate new decisions from the new tuple, $O(|E||T|)$.

The linear dependence on the number of predicates and the size of the world means that the algorithm would scale well with the addition of a large factual database, containing information that the robot cannot directly perceive. However, if extended chains of inference are necessary to prove certain facts, the $|E|$ term would need to be replaced with the running time of whatever proof mechanism is used to evaluate the truth or falsity of decisions. Efforts at code optimization would be best directed at this proof procedure, since it is effectively the innermost loop of the tree update.

Practically speaking, most robots will probably have a small number of predicates, and the dominant terms will be the number of tuples observed so far and the size of the environment. In

the online case, the time to update is usually linear in both of these quantities. Furthermore, as the number of examples increases, the chance of a major reorganization of the tree decreases, so that in the limit the updates will almost always be $O(d|E|(|P| + |T|))$, assuming value sampling. Thus, the algorithm should scale well in the long term to more words and more examples, as long as the number of numerical values is kept in check through sampling.

The complexity analysis above assumes that the new words that are learned are not introduced into the system as possible predicates for decisions. Doing so would replace —P— in the above running times with $O(|N|)$ in the worst case. It is unclear whether this would be a helpful thing to do; on the one hand, it would increase the running time without actually expanding the hypothesis space (since the new predicates would necessarily be redundant), but on the other, it might allow some decisions to be represented more concisely and allow structure to be shared across parts of speech. In any case, the present experiments only used the system's basic predicates (those contained in the description of the environment) in generating decisions, and so did not suffer the running time penalty.

7.6 Starting Trees

Clearly, if the algorithm is to understand all but one of the words in the sentence, it needs some word definitions to begin with. This implies that the algorithm needs some definition trees from the outset in order to use its word extension finder. Unfortunately, merely specifying the structure of a starting tree is insufficient; the algorithm must have access to the data that generated a tree in order to update it.

For the initial vocabulary, the algorithm must therefore be initialized with some sensory data in which the words are directly paired with their referents, circumventing the word extension finder. The starting vocabulary can be fairly minimal; the implementation for the experiment to be described began with five words, "ball," "pig," "is," "got," and "lost." The definition trees generated for these

words (using 45 labeled examples) are shown in Figure 7.2. Importantly, the data used to generate these trees must be generated under fairly similar conditions to the real online environment, or any consistent differences will be seen as highly informative. Ideally, the robot would be able to use pointing gestures or gaze direction to find these extensions without the extension finder, as this would presumably be most similar to how children learn these first examples in the absence of grammar; Yu and Ballard's gaze tracking word learner provided an excellent system for doing this, albeit not in a predicate logic framework (Yu and Ballard, 2004). In practice, it is not too difficult to manually provide words and referents for 45 different environment descriptions pulled from the robot's sensors, and this is what was done; environment descriptions from previous learning experiments were manually inspected, and the starting words were given appropriate bindings to referents in the environment.

An alternative, which TWIG used previously (Gold et al., 2007), is to have the starting definitions exist outside the definition tree structure, rendering their definitions unalterable and moot for the purposes of definition tree construction. This approach is less elegant, however, as the known definitions should ideally always inform the definitions of new words, and vice versa.

7.7 Decision Tree Learning Experiment: I, You, He, This, That, Above, Below, and Near

In this experiment, the robot used the Cricket indoor location system (Priyantha, 2005, (see Chapter 4)) to sense the positions of two objects, a stuffed pig and a tennis ball. In addition to the pronouns "I" and "you," now familiar to the reader, this experiment was also designed so that the robot could learn the third-person "he," proximal and distal pronouns ("this" and "that"), and some simple prepositions ("above," "below," and "near").

7.7.1 Setup

Cricket beacons (Priyantha, 2005) were attached to the stuffed pig and tennis ball. The robot used its face/facing detection system to determine subject facing, but did not use the now redundant color blob detection system. Sound localization used overall loudness instead of time-of-flight, and a context-free grammar containing only the target phrases was used for speech recognition.

The grammar used for parsing in Prolog was as follows:

```
<s> = <np> <vp>;
<np> = (this | that| the ball | the pig | i | you | he);
<vp> = (is <np> | is <p> <np> | got <np>);
<p> = (near | above | below);
```

The robot's sensory information was changed into the following logical predicates. `person(X)` was true of the robot and all faces detected. The distance V between each pair of entities X and Y (where an entity is a ball, pig, person, or robot) was stored in the predicate `dist(X,Y,V)`. The predicate `relHeight(X,Y)` was also calculated as the difference between each pair of entities' Z-coordinates. `ident(X,X)` held for each object with itself. `lookingAt(X,Y)` was true if X was a person and Y was an entity in the halfspace 30 cm away from X that was normal to X's facing direction. Each entity also had a predicate unique to itself to identify it. The speaker was denoted by a special variable S, instead of a predicate.

7.7.2 Procedure

For 200 utterances, the experimenter and another subject moved the stuffed pig and ball to different locations in the room, and then spoke one of the following utterances ([noun] should be understood to be "ball" or "pig"):

This is a [noun]; *That is a [noun]*; *I got the [noun]*; *You got the [noun]*; *He got the [noun]*; *The [noun] is above the [noun]*; *The [noun] is below the [noun]*; *The [noun] is near the [noun].*

Sentence Type	Example	TWIG Reaction	Proportion
Understandable, accurate	I got the ball (true)	Accept	48%
Understandable, data mismatch	I got the ball (wrong speaker)	Accept	5%
Understandable, ambiguous reference	That is that	Accept	0.5%
More than one new word	He got that	Discard	28.5%
No extension produces valid fact	He got he	Discard	10.5%
Misheard as question	That is what	Discard	7.5%

Table 7.1: TWIG's reaction to various speech recognition outcomes encountered during the experiment.

The locations for the items included next to the robot, on the steps of a ladder, in the hands of one of the experimenters, on various tables situated about the room, and underneath those tables. The experimenters remained roughly 60 cm in front of the robot and 50–70 cm away from each other, and faced the appropriate individual (or robot) when saying "you" or "he." [2]

7.7.3 Results

Many utterances were incorrectly recognized by Sphinx: at least 46%, based on a review of the system's transcripts. But because these false recognitions typically either included too many unknown words (e.g., "*He* is *near* the pig") or resulted in utterances with no possible new extension (e.g., "He got he"), the system usually made no inferences from them. A recognition error only affected tree development when it resulted in a sentence that contained exactly one unknown word. Table 7.7.3 lists the frequencies with which various errors occurred, and TWIG's reactions to them.

Figure 7.3 shows the state of the pronoun tree at the 27th, 40th, and final updates to the tree. The `person(X)` distinction remained the most informative attribute throughout the experiment, as it served to classify the two pronoun types into two broad categories. The proximal/distal distinction of "this" versus "that" was the next to be discovered by the system. The difference between "I," "you," and "he" remained unclear to the system for much of the experiment, because they relied on two unreliable systems: the sound localization system and the facing classifier, which had exhibited

[2]This experiment was first reported in Gold et al. (2007), but the decision tree algorithm has changed, and the comparison under "Evaluation" is new to this thesis.

error rates of roughly 10% and 15%, respectively.

The final definitions learned by the tree can be rendered into English as follows: "I" is the person that is the speaker. "You" is a person whom the speaker is looking at. "He" is a person who is not the speaker, and whom the speaker is not looking at. "This" is a non-person closer than 30 cm, and "that" is anything else.

The words "above," "below," and "near" were stored in a separate tree, shown in Figure 7.4, because the system inferred that they were a different part of speech. Because there were no contrasting examples for "near," and in fact only four recognized utterances including the word, it did not receive its own leaf. However, the system did learn that "above" and "below" referred to the appropriate differences in relative height.

7.8 Evaluation

In addition to evaluating the TWIG system by the qualitative goodness of the definitions it produces, I examined the number and accuracy of the sentences it was able to produce about its environment, compared to similar systems.

7.8.1 Evaluation method

The data generated by our experiment (described above) was used to train four different word learning systems, each a modification of the core TWIG system. In one variant, TWIG's extension inference system was disabled, so that the words were not bound to any particular object or relation in the environment. The definition trees were created under the assumption that a decision was satisfied if *any* object in the environment satisfied the decision. For example, `dist(S,X) <= 30.0cm` was satisfied if the speaker was closer than 30cm to any object. I call this strategy of associating everything in the environment with each word "associationist," after Bloom (2000).

As another variant, extension inference was allowed, but the system did not use definition trees.

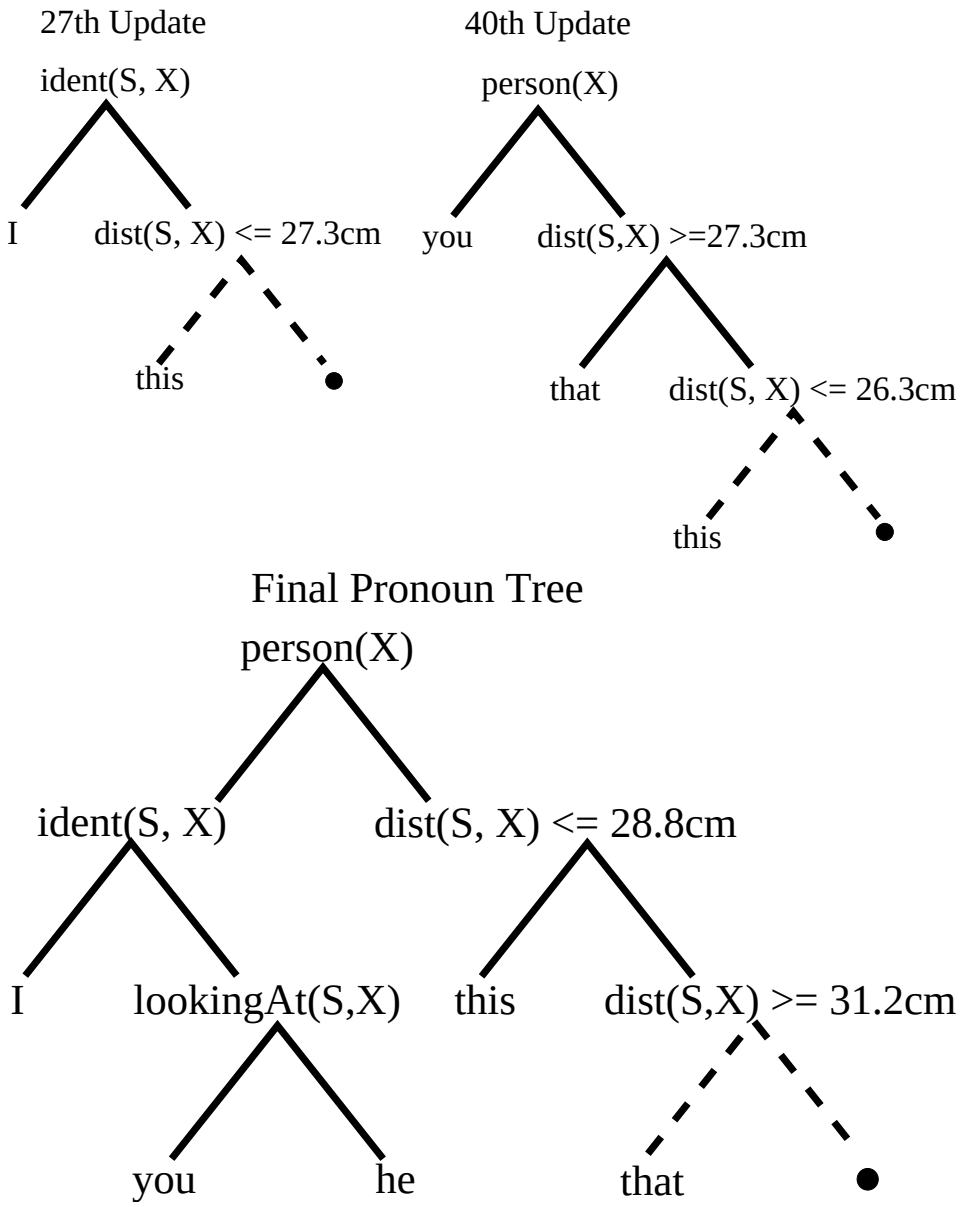

Figure 7.3: The state of the pronoun tree at the 27th, 40th, and final updates.

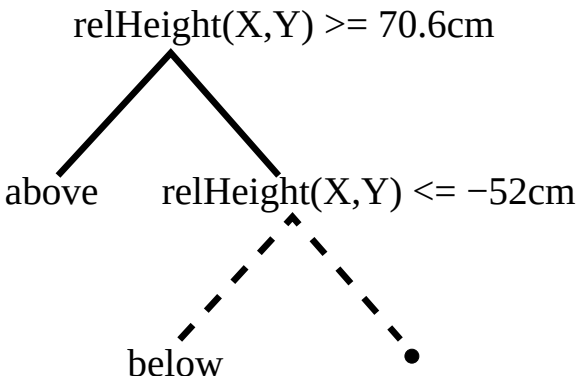

Figure 7.4: The definition tree created to define prepositions.

Instead, the system only used the single most statistically significant predicate to define each word, as measured by a chi-square test. This is the system that TWIG used prior to definition trees (Gold and Scassellati, 2007a), and demonstrates the behavior of systems that attempt to learn each word as a separate learning problem, rather than treating all of word learning as a single decision problem.

Toggling whether extension inference was allowed and whether full definition trees were created resulted in four system variants, including TWIG itself. Each system was presented with four test environments, taken from actual robot sensory data. For each environment, each system produced all of the sentences that were true of that environment, according to the semantics it had learned. For evaluation, the systems were each provided with valid Prolog definitions for the words "is," "got," "ball," "lost," and "pig."

7.8.2 Evaluation results

Figure 7.5 shows the results of the evaluation test. The associationist best-predicate system produced nonsensical definitions for every new word, but by chance managed to produce a fair number of correct sentences. The associationist tree-based system was more conservative and did not produce any more correct sentences than the tautologies implied by the definitions given to the system ("the pig is the pig"). The extension-finding best-predicate system was much closer to the performance of TWIG, as it produced all the correct sentences that applied to each scene that TWIG did. However, because

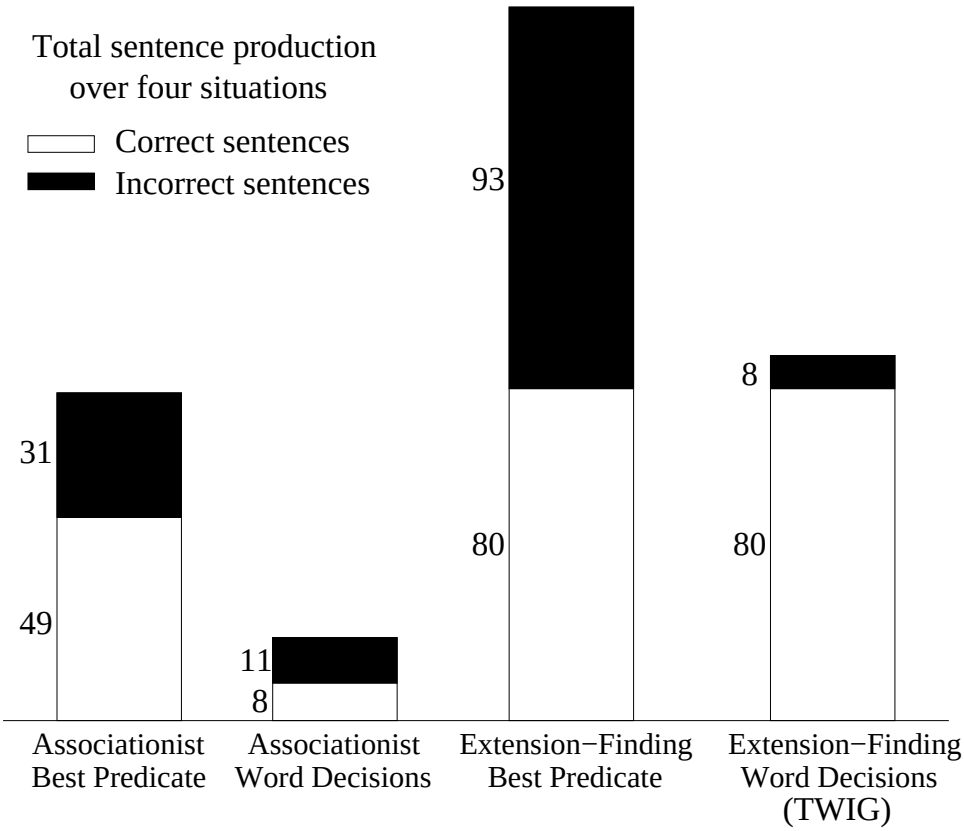

Figure 7.5: Sentence production across four variants of TWIG. Disabling TWIG's ability to find word extensions generally results in incorrect definitions, while using single predicates instead of definition trees tends to result in less nuanced definitions, and hence, overproduction.

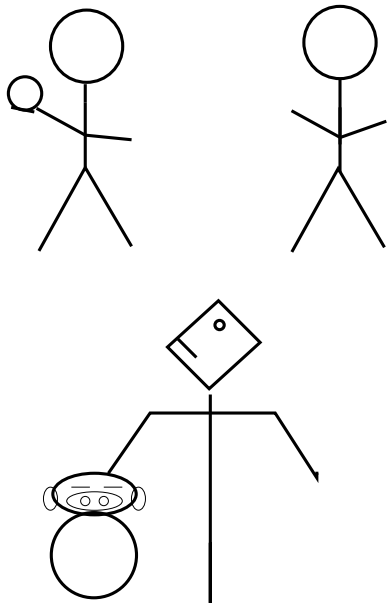

Figure 7.6: A diagram of the test environment that produced the utterances in Table 7.2. The robot has the pig toy and is addressing the person on the left, who has the ball.

Associationist Best Predicate	Associationist Word Decisions	Extension-Finding Best Predicate	TWIG (Extension Finding, Word Decisions)
∗I lost the pig.	∗I is above I.	I got the pig.	I got the pig.
∗That got the ball.	∗I is above the ball.	You got the ball.	You got the ball.
I lost the ball.	The ball is the ball.	∗That lost he.	He lost the ball.
That is that.	∗The pig is I.	∗That got the ball.	You got that.
∗That lost the pig.	∗The pig is above the pig.	This is the pig.	This is the pig.

Table 7.2: Sample sentences produced by each system to describe the situation in Figure 7.6. An asterisk indicates a statement counted as incorrect.

it could not represent conjunctions, its definitions of "this" and "that" omitted the requirement of being a person, while its definition of "he" was based purely on distance to the speaker. This resulted in sentences such as "that got the ball" (instead of "he got the ball") and "he is the ball."

TWIG's only errors in production were in the sentences "I is I" and "You is you" for each test environment, since it did not know the words "am" and "are." (Grammatically correct tautologies such as "he is he" were counted as valid across all four conditions.) Other utterances it produced that it had never heard before included "I got this" and "That is below the ball."

Table 7.2 shows some sample utterances that each system produced about the situation shown in Figure 7.6.

7.8.3 Discussion of evaluation results

First, it is interesting that TWIG can produce this kind of result at all, given that almost all previous learning systems had no means of producing language, but could only generate representations of their classifiers (Bailey, 1997; de Marcken, 1996; Regier, 1996; Roy and Pentland, 2002; Yu and Ballard, 2004). It is important to point out that this comparison is between reasonable variants of TWIG, but that in terms of production, TWIG and its variants are doing something that most other word learning systems cannot do at all.

Without extension inference, the system is unable to generate workable meanings for the words it sets out to learn, and its performance is more or less at chance levels. This supports the arguments of Bloom that "associationist" models of word learning, in which words are associated with everything that happens to be in the environment, are unlikely to be correct models of childrens' word learning (Bloom, 2000). It also places a heavier burden of proof on researchers who argue that children can do this kind of association, because usually their experiments have assumed that all words are concrete nouns, spoken in the presence of only a few possible referents, each of which shares no obvious properties with any others (Smith and Yu, 2006). This experiment suggests that in situations of any complexity, ambiguity in the extension begins to destroy any hope of learning the new words.

The definitions that the system produced under the "single best predicate" scenario were not as complex as the definitions produced by definition trees, which led to overproduction. For example, the definition of "that" could refer to distance or personhood, but not both, leading to a system that called people "that." This demonstrates that definition trees are a useful advance beyond the intensions described in the previous chapter and in Gold and Scassellati (2007a).

7.9 Discussion

Much of the discussion of the system as a whole will be reserved for the final chapter; here I will discuss some of the successes and failures of definition trees in particular.

Definition trees are a useful new approach to word learning because they reframe what was previously treated as a collection of individual concept learning problems into a single decision problem. As such, they accomplish much more than one might think possible with only "positive examples" and no negative feedback. They allow the learning of concepts that include conjunction (through consecutive positive branches), disjunction (through words appearing at different leaves), and negation (via the negative branches). The algorithm can also learn multiple meanings for the same word, because the same word can appear at different leaves with radically different meanings. The algorithm does not remain too conservative in its definitions from positive examples because the tree partitions the space of objects completely, and because items that exceed the threshold for a numerical property continue to satisfy the relevant property. Nico has never seen an object a mile away, but it would still know to call it "that," not "this."

The ability to learn logical conjunctions and thresholds is especially important, because it allows definitions that are more complex than the predicates which the system begins with. Learning new meanings would be a hopeless task if every word essentially already had to have a predicate built specifically for it; yet this is the assumption made in much word-learning work that uses logical semantics (Kate and Mooney, 2007; Siskind, 1994). Conjunctions and thresholds leave open the possibility that new concrete nouns or adjectives could be learned by chaining low-level sensory predicates.

Some of the definitions may seem too simple, but it is worthwhile to keep in mind that the system is built to learn *intensions*, or the facts that must be true of a referent for the word to pertain to it, and not *associations*, or what Frege called the "idea" of a word (see Chapter 3). "I" has more connotations to a human than it does to Nico, but Nico's definition is sufficient to interpret nearby speakers' sentences or produce its own. Still, the definitions of "this" and "that" are not as general as one might like, since they cannot capture reference to abstractions ("this idea") or take into account relativity of scale ("this great country").

The addition of the speaker variable S may have seemed an ad hoc solution to the problem of

deixis, but it should prove useful even for words that aren't deictic pronouns. For example, it can allow the learner to take a speaker's attitude toward a referent into account: one can imagine a system in which `likes(S,X)` distinguishes the words "good" and "bad." The case of interjections is also interesting, because they have no extension at all; yet every language has words that convey `angry(S)`.

The experiments here did not involve trees that could refer to variables that were neither an extension nor the speaker. Such a feature would allow the system to learn definitions such as "an object that is being looked at by A, for some person A." Early experiments did include this feature, but I eventually cut it. The system would often choose the splitter `dist(X,A) <= V` for some V and then never add any restrictions on A, because there was generally some distance for which this predicate split the examples best. I do not know whether this is an unavoidable problem with this kind of variable, or if it is fixable by introducing some kind of complexity penalty. I could not think of an application for this capability that was realizable with the sensors at hand, and so did not pursue the matter.

Another system which did not work, oddly enough, was using chi-square values instead of information gain as the splitting heuristic for the decision trees. Though some sources suggest that using chi-square values to decide when to split should be as good as information gain (Bremer, 1985), this was not at all true for these experiments. Chi-based trees tended to produce many unnecessary branches in the trees; even after pruning for high significance, the word "you" could be located at several spots in the tree, each corresponding to a different distance or some other irrelevant attribute. Statistical significance, it seems, can quite often signify a not very interesting or important distinction.

In Chapter 3, I discussed the Principle of Contrast, which states that young children tend to overextend words until they receive contrasting examples (Clark, 1987), and the Principle of Mutual Exclusivity, which states that children tend to accept only one word for a given object at first (Markman and Wachtel, 1988). I did not intentionally attempt to model these effects when designing

my definition trees; as I related at the beginning of this chapter, I came up with the idea of using definition trees primarily to solve the problem of how to learn the definition of "he" in a natural manner. So it is rather exciting that these effects naturally fall out of the definition tree approach. It suggests that one can gain insight into human development by attempting to solve the same problems as infants; and vice versa, that attempting to solve things in a human-like manner might generally be a good heuristic for design. Nevertheless, definition trees might implement these heuristics a bit too strongly, as children eventually do learn that multiple words can refer to the same object, but these definition trees currently do not.

It is tempting to give a developmental interpretation to the order in which the words were learned in the experiment, but this is probably fallacious. The order in which words are presented to the system matters a great deal to the order in which they are learned, and this was fairly arbitrary in the experiment presented here. Word tree development could be treated as a model of human word learning only if one took into account grammatical, conceptual, and perceptual development. To be used as a predictive model, word trees would need to be presented with realistic word and property frequencies, with certain properties becoming available to the system only at certain developmental milestones.

In the next chapter, I will summarize the contributions and shortcomings of the TWIG system.

Chapter 8

Conclusions

The TWIG system grew out of a simple puzzle: how could a robot learn the meanings of the words "I" and "you" from the evidence of its senses? Recall that the state of the art for robotic word learning was to associate words with fairly raw visual information (Roy and Pentland, 2002; Yu and Ballard, 2004), while the state of the art in simulation was to leave the production of possible meanings to a mysterious "black box"(de Marcken, 1996; Kate and Mooney, 2007; Siskind, 1994). Not only did TWIG answer the question of how a robot could learn the meanings of "I" and "you," but it could learn several other pronouns besides – "he," "this," and "that" – as well as, with a few simple modifications, some basic prepositions and even some transitive verbs. As I had hoped, answering the question of how to learn personal pronouns led to techniques that were useful across several kinds of words.

Below, I will outline some of the technical advances of the TWIG system, followed by the ways in which it informs our understanding of human development. I will then go on to point out some of the limits of the system and my findings, followed by a section on some of the ways in which the system might expand.

8.1 Technical Advances of the TWIG System

8.1.1 Full sentence production and comprehension

TWIG can create and understand full sentences composed of the words it knows. No other word learning system does this with words learned in an unsupervised manner. (SHRDLU allowed the user to give the definition of a new word explicitly (Winograd, 1971).)

This is not a trivial extension of the previous robotic work, but the result of a whole different attitude toward word meaning. The idea of evaluating the truth value of a whole sentence stems from much earlier work in language processing, which used formal semantics to change natural language into logical statements. Robotics has typically taken a different attitude about word meaning, treating it as the statistical association of sensory data and word (Roy and Pentland, 2002; Yu and Ballard, 2004). This is in fact the method of Chapter 5, and it makes some amount of sense for real, unconstrained speech, simply because vocabularies are so large and speech is rarely grammatical. (The CHILDES transcript of the game of catch (Bohannon, 1976) which was used in the simulations of Chapter 5 was no exception.) Even text processing has mostly moved toward statistical methods that look only for co-occurence of words instead of logical semantic structure (Manning and Schütze, 1999a).

But where is the system that can use these statistical associations to create a meaningful sentence? Though systems that rely on logical semantics to parse meaning may be more brittle than systems that do not particularly care about word order, in the end, grammar contributes to semantics. Systems that attempt to find the meaning of transitive verbs and prepositions will fail unless they can infer which two actors or objects are being related by the word; and the best way to do that is through grammatical inference. Thus, though TWIG currently would not work well with unconstrained speech, I believe its approach to grammar is a more solid foundation on which to build such a system.

8.1.2 Learning word meaning for multiple parts of speech

TWIG has inferred meaning for verbs, pronouns, and prepositions, and should be able to infer meaning for some kinds of nouns. Most systems equipped with sensors only attempt one part of speech (Bailey, 1997; Regier, 1996; Roy and Pentland, 2002). Again, this difference stems from a fundamental attitude difference when approaching word meaning.

When a system is build to only learn words within a particular part of speech, it can assume that the relevant features will remain roughly the same across all the words, and that all features will contribute in some degree to the meanings of all words. In Roy and Pentland (2002), for instance, a built-in assumption was that every word would have something to do with the shape in front of the camera. In Bailey (1997), the verb learner assumed that every definition would include some description of force. If Bailey's system had produced meanings for verbs that included Roy and Pentland's shape histograms, the result would have been a kind of type error, as shape cannot apply to an action.

It is exactly this dependence on all features that could not work for "I" and "you," for which most noun features would only be misleading. This is why TWIG creates definitions that only contain the smallest number of features necessary to differentiate words within the same part of speech. While it might have been possible to achieve the same effect with weights for all features, some of which tended toward zero, it is likely that some erroneous features would have remain in the definitions by virtue of the limited test environment, and that any sufficiently complex environment would produce some erroneous associations that might take thousands of trials before they were driven to zero. As it is, TWIG learns its meanings on a timescale that is much more acceptable in the realm of real-world online learning.

TWIG also uses grammatical inference to separate words that belong to different parts of speech, so that decisions do not refer to the wrong number of referents. This allows transitive verbs and prepositions to include decisions based on the relationship between their two referents, while keeping such decisions out of the pronoun decision tree, where such decisions would be nonsensical.

8.1.3 Complex meanings: negation, conjunction, numerical values, relations

TWIG can represent complex word meanings, including several boolean logical operators (not, and, or), thresholds on numerical values, and relations to other objects. In particular, including both numerical values and logical operators is somewhat rare in this line of research, as typically systems employ only one or the other (for example: Kate and Mooney, 2007; Roy and Pentland, 2002; Siskind, 1994; Yu and Ballard, 2004). The fact that meanings can be assembled from component predicates means that the system can learn new concepts even as it is learning the words for those concepts. This is an improvement over systems that assume each word corresponds to a single predicate that already exists in the robot's reasoning, which would limit the possible definitions to those explicitly anticipated by the programmer. TWIG is unique in that it learns numerical thresholds on sensor values, but can still produce and comprehend full sentences.

8.1.4 Passive learning without feedback

One of the reasons that children are such quick language learners is that they can learn without being explicitly taught (Bloom, 2000; Heath, 1983). As a result, children learn far more words than their parents would ever have time to explicitly teach them. It is an example which robot designers should heed. Any system that requires the user to explicitly train it will receive fewer training examples than one which can simply make use of the language it happens to observe.

Passive learning without feedback was critical for learning the meaning of "you," because at some point the robot had to observe someone speaking to somebody else. But speech that is not didactic is much more likely to be composed of complete sentences instead of isolated words, and is much less likely to be illustrated with helpful pointing gestures. The answer to these problems was to assume some limited vocabulary and grammar, and use sentence context narrow down which word was new and infer its extension. Though the experiments conducted here were obviously for the

robot's benefit, and the small speech recognition vocabulary limited what the participants could say, in theory passive learning should be much more useful than any method requiring explicit feedback.

Note as well that sentences which TWIG cannot comprehend do not adversely affect the learning; if a sentence contains more than one word that is not understood, it does not affect the definition trees, but is simply tossed out. Thus, the method is in theory usable as-is to learn from real speech, though in practice one could desire a better ability to make use of malformed sentences or more complex grammatical structures.

8.1.5 Learning in the presence of noise

Real systems must contend with faulty sensor readings and error-prone speech recognition, a fact that limits the utility of the cognitive modeling work that assumes noise-free input (e.g., Bailey, 1997; Regier, 1996). TWIG deals with noise by using information gain instead of strict logical inference as its means of determining which definition to use for a word. This approach is arguably cleaner than Siskind's simulated annealing-like process for dealing with noise (Siskind, 1994), since information gain is an exact calculation from the data, while Siskind's method includes more parameters that must be tuned.

TWIG is also robust to noise because recognition errors will typically result in sentences that do not make sense, and which therefore cannot be matched to anything in the robot's environment. Sentences that cannot be matched to the environment are discarded and do not affect learning. This is one of the major reasons which TWIG was able to function despite the very high error rate of the Sphinx speech recognition system. Though a single recognition error might result in the robot attempting to learn a new word definition for the erroneously recognized word, two errors, or an error and a new word, will typically result in the system discarding the sentence. One can think of this effect as being somewhat similar manner to that of error-correcting codes (Hamming, 1986).

8.1.6 Correct generalization of potentially infinite quantities

Many words refer to quantities that can be nearly infinite – for example, "far," "hot," "bright," and so on. Word learners will probably never experience these extremes themselves, and yet children correctly generalize these words to include extreme cases. How do they do that without any extreme examples?

Using the threshold generating method of TWIG produces the desired results. By giving each word the maximum coverage of the hypothesis space possible until it competes with another word, TWIG guarantees that some word will always apply to a situation, no matter how extreme, and that it will be the closest word learned so far. Using this method, Nico learned that "that" refers to anything farther than about 30 cm, whether it is a mile or a million miles away, despite the fact that it never observed examples that were much farther than a few meters.

By contrast, Roy and Pentland (2002) created audio-visual templates with strictly limited boundaries, Bailey (1997) avoided the problem by assuming that quantities only assume a small number of discrete values, Regier (1996) used a neural network to classify only objects within its finite field of vision, and Siskind (1994) does not handle scalar quantities at all.

8.1.7 Dealing with deixis and pronouns

This system is the first to learn meanings of words that involve deixis, or reference to the speaker's situation. As such, it can be seen as a novel contribution in the same way that Bailey's system was novel for learning verbs (Bailey, 1997) and Regier's was novel for learning prepositions (Regier, 1996). Unlike those systems, TWIG has advanced beyond its starting word category. In fact, the ability to handle this class of words is probably not nearly as important as the way in which solving the problem of learning deictic pronouns shaped the rest of the system to be more general.

Deixis is not limited to the deictic pronouns, since many words depend on the speaker's point of view for reference. As I point out in Gold et al. (2007), interjections are an excellent example. Rather than assume that swearing is mere verbal behavior without any semantics, it is probably

more useful to treat such words as having a speaker-centric, extensionless semantics, i.e., `angry(S)`. Words involving value judgments, such as "good," are another example. Being able to learn and use deictic definitions is an important step toward being able to reason more generally about the speaker's point of view.

8.2 Contributions to Other Disciplines

8.2.1 Pronoun reversal as lack of linguistic evidence

The phenomenon of pronoun reversal, in which children misuse "I" and "you," has previously been thought to be caused by social disorder, such as an inability to take another person's perspective (Andersen et al., 1984; Brown et al., 1997) or a lack of self-concept (Fraiberg and Adelson, 1977). My early research into "I" and "you" (see Chapter 5) suggests a different hypothesis: that this is fundamentally a linguistic rather than conceptual error, caused by a lack of linguistic evidence.

In some cases, such as autistic pronoun reversal, the linguistic error may indeed stem from a conceptual problem, though whether that conceptual problem is an inability to reason about other minds (Baron-Cohen, 1995) or a more general learning disability (Minshew and Goldstein, 1993) remains unclear. It is possible that autistic pronoun reversal may be the result of an inability to shift perspective and use speaker-relative words; certainly, removing the "S" symbol from TWIG would be one way to cripple it in learning "I" and "you." It is difficult to make any conclusions about normal development from autism, because the symptoms of autism are numerous and poorly understood (Bailey et al., 1996).

But in the case of blind children, I have shown that it is a mistake to think that some additional social deficit is necessary to explain pronoun reversal. The mere fact of blindness is sufficient to cause a lack of sensory evidence with which to learn the words "I" and "you," with no intervening causal explanation necessary. It is a mistake to assume that an error with words betrays an error in underlying ability to conceptualize. Occam's Razor dictates that in this case, that proposed social

disability should be discarded. Though it is conceivable that a lack of sensory evidence might lead to deficits in social cognition, the researchers that take pronoun reversal as evidence of such a deficit (e.g. Andersen et al., 1984; Brown et al., 1997) are mistaken to do so. The ability to infer reference is critical to learning "I" and "you," and without an ability to see pointing or the other objects mentioned in a sentence, it should come as no surprise that blind children learn these words later than others.

This explanation perfectly coincides with the finding that pronoun reversal tends to occur with children that are precocious language users (Dale and Crain-Thoreson, 1993). Why are otherwise advanced children making "I" and "you" mistakes? I posit that it is because they simply lack experience. Again, it is possible to hypothesize an intermediate cause, and think that these children lack perspective-taking ability because they are young, and so fail to learn deictic pronouns correctly – but the mere fact of being young and inexperienced with the use of these words seems itself a sufficient explanation.

8.2.2 Evidence for the importance of inferring reference

There is disagreement in the psychology community about whether children associate the words they hear with everything in their environment, or associate words more specifically based on the meaning of the sentence. Some researchers have shown statistical evidence that, given a series of collections of objects described by isolated words with no cues to reference, children will still look longer at the correct object when they hear a word during testing (Smith and Yu, 2006). On the other hand, the fact of this statistical evidence does not necessarily prove that this is the normal state of affairs, especially since a real environment typically contains many more than the four or so objects used in those studies. Young children often hear new words while not attending to the objects to which they refer (Harris et al., 1983), leading Bloom (2000) to argue that children must have some means of narrowing down the possible meanings of a word besides sheer statistical association.

The evaluation study presented in the previous chapter provides some proof that the latter camp

is correct: statistical association of every word with every possible referent is probably just not very effective. Neither implementation of the system that excluded extension inference produced correct definitions. On reflection, it should be obvious that it is impossible to learn the correct definitions of "I" and "you" without finding their extensions, since all words are used in the presence of a speaker and a person being spoken to, and the only difference in the case of these words is that they *refer* to the speaker or the person spoken to. But an actual implementation carries more weight than a logical argument.

8.2.3 A decision tree model of the Principle of Contrast

The Principle of Contrast has not previously been modeled as a decision tree.Yet the decision tree is a perfect data structure for this purpose: it builds models of increasing complexity for each concept, is resistant to noise, does not require constant iteration over "epochs" to reach its final state, has an easily specified mathematical relationship to the information available in its input, and is transparent to analysis of its structure. Though neural networks remain by far the most popular computational models among psychologists, I hope that the elegance of the decision tree model might encourage psychologists to consider higher-level representations of knowledge, leaving the details of neural implementation for later.

8.2.4 The semantics of deixis

Pronouns are sometimes called "indexicals" because, philosophically, they are sometimes thought to require an "index" into an array of possible worlds for their truth value to be ascertained (Dowty et al., 1981). My research shows a different way of thinking about these words: namely, they do not require dealing with a "possible worlds" framework at all, but only require checking facts about the speaker in the real world.

The "possible worlds" framework for semantics is problematic, because the idea that a speaker must cognitively deal with possible worlds when constructing a perfectly innocent sentence about

the state of *this* world makes little sense for such simple words as "I" and "you." While it is possible to argue that semantics need not be entirely represented in the brain – that the semantics of a sentence could be instead an abstraction describing a sentence which its speaker need not understand – it is probably more elegant to keep the philosophical semantics of a sentence and its speaker's internal representation of the sentence as closely matched as possible. Otherwise, we would require an additional explanation for how people manage to successfully create sentences with truthful semantics without actually being able to manipulate faithful representations of the semantics.

To delve completely into the philosophical literature about this issue would take this thesis somewhat far afield from its primary purpose and findings, and I have no doubt that my understanding of the philosophical treatment of possible worlds in semantics is somewhat superficial. Nevertheless, it seems odd to posit computationally intractable entities when dealing with the semantics of deictic pronouns, and I believe my method of dealing with indexicals here is one step toward a more sane deictic semantics.

8.3 Intentional Omissions

Having briefly gone over some of the ways in which this work is a novel contribution, I will now review some of the issues which this project did not address and explain why they were omitted.

8.3.1 Word discovery and the segmentation of language

Some of the most noteworthy robotic word-learning projects also solved the problem of word discovery and segmentation: finding word boundaries when the words were previously unknown (Roy and Pentland, 2002; Yu and Ballard, 2004). Unlike those projects, TWIG assumes access to speech recognition technology sufficient to noisily transcribe the utterance into words. In fact, the speech recognition module even included an explicit grammar that included the words that were "new"! This no doubt must seem like cheating to researchers most familiar with these other projects, but

there are good reasons to treat segmentation as a separate problem from semantics, particularly in TWIG.

First, there is evidence that infants can find word boundaries in a nonsense language that lacks any semantics (Saffran et al., 1996). This suggests that the processes of learning semantics and segmentation are separable in humans, and that segmentation may even precede semantics, since the infants in Saffran et al. (1996) were preverbal. This would make sense, since there are undoubtedly many situations in which infants do not have sensory access to the referents of speech, and they would be wasting information if they ignored the speech entirely for this reason. Thus, from a modeling point of view, leaving segmentation as a separate problem is entirely justifiable.

Second, speech recognition software already exists as a commercial technology. It would be somewhat quixotic to attempt to rebuild a speech recognition engine from scratch with the additional requirement of lacking a language model unless one specifically wanted to model human development. A module that can make use of this existing technology is presumably more useful to roboticists, as it allows them to choose their own speech recognition technology and upgrade it at will as new commercial software is released. Building a homegrown system that learns to segment requires more effort on the part of researchers attempting to replicate the work, and is more likely to vary from lab to lab.

Third, TWIG makes the assumption that all of the words but one are understood in a sentence, so in the cases where TWIG can succeed at all, segmentation should theoretically be straightforward. In theory, one should be able to modify Sphinx to allow for out-of-vocabulary utterances, and save a best-guess phonemic representation for the segment that is most likely to be out-of-vocabulary. In practice, I had to use a specific context-free grammar with no out-of-vocabulary utterances in order to get decent recognition performance in our lab environment, but this was probably partly because my "I" and "you" work required speaker independent speech recognition, which generally has worse recognition accuracy than speech recognition that can be trained to an individual speaker (Jurafsky and Martin, 2000).

For these reasons, TWIG treats segmentation and semantics as separate problems, and the former is outside the scope of this thesis. I have, however, done some work on segmentation without any language-specific knowledge; see Gold and Scassellati (2006a).

8.3.2 Learning concrete noun representations

In the end, I never found a representation of shape and appearance that I was entirely satisfied with, and certainly not one that was better than the existing work. It was unclear how even the existing work on concrete nouns could be integrated into TWIG; the decision tree algorithm currently only finds a threshold for a single scalar when creating a split, making the exact basis for the representation more important than it might be otherwise. One could add features for length, width, height, and various spatial moments; but then what? And how would visual segmentation work without a blue screen background (Roy and Pentland, 2002) or a head-mounted eyetracker to find the object (Yu and Ballard, 2004)?

TWIG's ability to create definitions based on conjunctions and thresholds on real values suggests that there is probably some way to integrate a complex representation of shape into its decision process. But ontology becomes all-important when building decision trees that hinge on single scalars. It is possible that the decision tree algorithm itself needs to be modified so that it can create its own multidimensional thresholds, rendering the choice of basis less important; but a computationally tractable method of finding the surface of maximal information gain is an area for future study.

Despite this, it may be incorrect to attempt to shoehorn shape into the current word learning system, when shape is usually only a *cue* to whether a word belongs to a particular category. For example, a black plastic box may be decorated to look more or less like a desktop computer, but it will not qualify as a computer unless it functions as one. It is possible that appearance is usually better thought of as belonging to the associations with a word (what Frege called its "idea"; Frege, 1892/2003) instead of its intension.

8.3.3 "She"

As an altogether different kind of conspicuous omission, technical hurdles prohibited me from including "she" in the pronoun learning, when TWIG did include "he." Though using average voice pitch as a feature on which to split in the decision tree might have produced this decision, in the end, implementing this feature would require a significant amount of time without actually addressing a new conceptual problem, since it is clear that TWIG could make this split given an appropriate feature.

8.3.4 Proper nouns and definition trees

It is very difficult to deal with proper nouns in the definition tree schema. Proper nouns are grammatically treated as if they were pronouns, but if they are forced to contrast with other pronouns in a definition tree, the pronouns and proper nouns begin to warp each other's definitions to each include the other's negation. It then becomes impossible to form the sentence "I am Kevin," because "Kevin" may include "not speaker" in its definition (which is normally correct). It is possible that proper nouns obey their own rules and do not use the Principle of Contrast at all, or perhaps they are best thought of as being purely extensional. On the other hand, perhaps the statement "I am Kevin" is an example of a word becoming acceptable once the better word has already been used ("Kevin am I" sounds strange).

Note that the problem is not with learning "am"; in unpublished data, I have found that definition trees can handily learn that "am" means `ident(X,Y) & ident(X, S)` – not only replicating the result of Chapter 6, but taking its first-person aspect into account as well.

8.3.5 Using real open vocabulary speech

The experiments to use completely open vocabulary speech that appear in this these were the initial experiment in simulation (Gold and Scassellati, 2006d) using the CHILDES corpus (Bohannon, 1976;

Bohannon and Marquis, 1977; MacWhinney, 2000). For implementation on Nico the robot, all of my experiments required the subjects to utter statements that belonged to a small context-free grammar, even though this sometimes seemed absurd ("I am Kevin, you are Justin!" "I am Kevin, you are Justin!") In fact, the difficulty of dealing with unrestricted language is one of the main reasons why grammatical parsing has given way to less grammatically based statistical methods (Manning and Schütze, 1999a). It therefore behooves me to explain why I believe this kind of grammar-based parsing will eventually be able to handle unrestricted speech, despite the fact that the present system can only handle a tiny fraction of grammatical utterances, let alone ungrammatical utterances.

Ultimately, grammar is a way of encoding relationships between words over time. To understand the semantics of a complete sentence, there must be some way to bind adjectives to their correct nouns, and verbs to their correct subjects and objects. Any model that does not capture the actual rules of composition of a language is probably not going to do well in the long term. The use of bigrams and trigrams to capture similar information strikes me as a temporary fix, a way of making current technology suffice, since it cannot capture some of the long-range dependencies that are actually observed in language. Bigram and trigram models will not result in human-level language understanding, any more than current chatterbot technology will result in an A.I. that can pass the Turing Test with full generality. Sometimes performance must be temporarily sacrificed to make real advancements.

To add the ability to deal with arbitrary speech in a grammar-based manner would be no small feat. Not only would the basic grammar need to be huge, but there would also need to be some kind of edit-distance-like procedure for determining what grammatical parse is closest to the speaker's undoubtedly ungrammatical utterance. Opening up the vocabulary would also require very good speech recognition performance, something I was unable to squeeze out of the speaker independent Sphinx 4 platform. Still, TWIG's limited grammar is ultimately a stronger foundation on which to build than the grammarless statistical methods of Chapter 5.

8.4 Criticisms

The previous section dealt with features that are not a part of TWIG; this one discusses valid criticisms of the system as implemented.

8.4.1 Problems with mutual exclusivity

The system currently assumes not only that every word must vary somewhat in meaning, but that each word implies the negation of the others. This is particularly problematic for prepositions; why should "above" imply not "near," as it must under the assumptions of definition trees?

This is an entirely valid criticism which may prove deadly to the whole idea of using definition trees. There are three possible solutions that I can see. One is that definition trees simply do not work for certain categories of speech. Some other representation might be better for adjectives and prepositions, which sem to usually lack the kind of complex structure that would necessitate the use of definition trees. "Above," "blue," "long," and so forth do not seem to require conjunction or negation, and it is possible that words that modify other words obey their own rules and involve much simpler definitions.

Another possibility is that definition trees determine what the *best* word is for a given referent or pair of referents, but that some rule ought to be invoked that modifies subsequent words' meanings so as not to negate any part of the original word. For example, an adjective definition tree that includes both "giant" and "red" might dictate that using "red" implies not "giant" (but not the reverse, if giant is higher in the tree). The proposed modification to TWIG would dictate that once "giant" has been spoken, then the speaker is free to use "red" without implying that the extension is not large. Thus, it is possible that definition trees would not completely exclude multiple words from referring to the same object, but would only encode a requirement for using particular words *first*. Such a system might also allow salience or novelty to alter the hierarchy of words in the tree.

8.4.2 Problems with hard thresholds

The present system uses step-function thresholds for its numerical values, creating precise and sudden shifts in which word to use. This is clearly incorrect in the long term. It makes little sense to define an exact distance cutoff for a word such as "near," and even less for a word such as "this." Ideally, meanings should allow for variation in the goodness of fit of a particular word, rather than introducing artificial discontinuities.

This is not a problem which most other word learning systems have been able to solve, either, as most introduce some kind of discontinuity in defining their words (Bailey, 1997; Roy and Pentland, 2002, e.g.,). It is particularly difficult to solve in the context of a system that produces logical form semantics, in which statements are either true or not. A system that could produce compositional meanings that also include "fuzzy" or probabilistic truth would be a significant advance, but it is worth keeping in mind that other systems do not use numerical values in producing compositional semantics at all. TWIG is clearly a first step here rather than the last word.

8.4.3 The importance of context

Many words have intensions that depend on context – for example, "hot" can refer to different temperatures depending on whether it refers to an apple pie or a summer day. TWIG could conceivably capture some of this information by including separate branches in its tree for each possible context. For instance, `food(X)` could determine whether "hot" should use a threshold appropriate to food or for weather. Still, it is not clear that this approach could work with "this" and "that," for which the context is the general scope of the conversation, a context not easily made discrete. Clearly, the absolute distance thresholds for these words generated in the experiments of Chapter 7 are not the last word on learning these words; those results were merely a demonstration that the system could generate its own numerical thresholds from sensory data.

8.4.4 Problems with learning from descriptive sentences alone

TWIG only learns from sentences that state facts, and in these experiments, it only used sentences that described the immediate environment. However, Bloom (2000) argues that this kind of learning is still falling into a trap of "associationism," since children can often learn the meanings of words from contexts besides being in the immediate presence of the referent. For example, if the speaker asks the learner, "Do you want a *macaroon*?" the learner should be able to say yes, receive the treat, and thereby learn what a macaroon is, even though the speaker never uttered a declarative sentence. A world in which speakers only tell each other things that are immediately obvious by looking around is admittedly not very realistic.

The experiments leading up to TWIG did incorporate a heuristic for learning words from questions about wants (see Chapter 5), but it was not very general or principled and was not incorporated into the full system. There is clearly room for improvement here in making use of utterances that are not statements of fact. The process of making inferences from questions and imperatives should be quite similar to the process for declaratives, besides needing some extra set of facts to match the utterance to (a model of speaker motivation?), though admittedly this is not proven here.

TWIG at least provides a foundation for a system that could use questions and imperatives as sentence context, and that this is good enough for the time being.

8.4.5 The generative lexicon

Placing every word type in its own tree means that the system does not possess a "generative lexicon," in which learners use words in novel ways that sometimes cross type boundaries (Pustejovsky, 1995). For example, one can speak of "LaTeXing" a file despite the fact that "LaTeX" is a noun.

Though some kinds of compositional grammar allow for a fairly fluid mingling of nouns and verbs – for instance, nouns and intransitive verbs could theoretically be treated as belonging to the same category, since they each only require one referent – it is not at all clear to me how to design a system that would include this kind of freedom while still producing grammatical utterances.

(TWIG originally only created two trees, one for two-argument words and one for one-argument words, but the resulting "verbing" of pronouns into intransitive verbs produced some decidedly strange sentences.) It also seems as though a system that knows a noun meaning for a word should have some advantage in learning a related verb meaning, and that is not something that TWIG can do. In general, a system that could creatively extend words across type boundaries seems so far removed from TWIG, I cannot even begin to describe how to get from here to there; but this is not a problem that earlier robotic word learning systems have addressed, either.

8.5 Extensions and Future Directions

This section will describe some of the ways in which TWIG can be extended, the outstanding research problems that would need to be solved to implement these extensions, and some sketches of possible solutions.

8.5.1 Learning the meanings of phrases and morphemes

Ideally, one would like TWIG to not only learn the meanings of words, but also learn the meanings of larger or smaller units, such as phrases ("kick the bucket") or suffixes ("-ed," "-s"). If TWIG could do this, then it could also abandon the one new word per sentence limit; parts of sentences could be learned holistically at first, and broken down into their constituent parts later.

To accomplish this, TWIG could abandon its classic parts-of-speech classifications in favor of a categorial grammar (Dowty et al., 1981; Tellier, 1998). In a a categorial grammar, words are not classified by meaning, but by the parts that are necessary to be added to make a complete sentence. For instance, an `S\NP` fragment is anything that requires exactly one noun phrase to make a complete sentence; this could be an intransitive verb, or a transitive verb with an object, or a verb modified by an adverb. In this way, words in different parts of speech may end up in the same tree, but all members of a tree would have the same number of referents and rules of combination with other

parts of language.

The ability to learn the meanings of phrases and morphemes would make the system more versatile in its productive abilities, and would allow it to infer the meanings of new words and phrases more often.

8.5.2 Relation of phrase learning to grammar learning

Pursuing the speculations of the previous section: Once a definition tree that can learn phrases contains a certain amount of repetition in its structure, this may be a cue that its phrases can be broken further into their constituent parts. For example, if several branches of the tree all contain meanings that have to do with talking, and the phrases at the leaves all begin with "talk," this could be a cue that the remaining parts of these phrases are actually smaller units – "-ed" and "-ing," for example. Trees might be checked periodically for such repetition, and reorganized if the resulting representation is more concise.

Such a strategy might allow the system to create new grammatical categories, though it is honestly unlikely that things would be this simple. One difficulty with this approach might be that the system would create too many categories. This appears to be a common problem in systems that rely on compression for learning (e.g. de Marcken, 1996; Solan et al., 2005): far more categories tend to be created than the familiar noun, verb, and so on. Such systems tend to create far more specific categories, such as "food," "animal," and "things that can be thrown," and then fail to realize that these things can generally be used in the same structures. Still, this is a speculation worth pursuing, especially since Montague-like semantics has previously been shown to be useful in learning grammatical categories (Oates et al., 2004; Tellier, 1998).

8.5.3 Greater flexibility in recognition and segmentation

The Sphinx 4 language model assumes that there is some fixed transition probability of using word B after word A, and uses this assumption to aid language recognition. Besides being somewhat

misleading even in the limit, since word choice has more to do with intended meaning than the previous word, it is probably not a good assumption that we possess reliable transition probabilities for a new or uncommon word. Certainly it is nonsensical to model children's language acquisition with a built-in language model. This makes it desirable to find some way of using the system without incorporating prior knowledge of words and their probabilities – perhaps working with a noisy phoneme stream instead, as done in other work (de Marcken, 1996; Roy and Pentland, 2002; Yu and Ballard, 2004). But this increases the likelihood that the words themselves will be noisy, and we would like similar words to be classified together in the decision tree. In other words, during learning, "this" should be treated as very similar to "thith" for the purpose of decision tree creation, instead of being treated as a completely different arbitrary symbols.

Though removing the language model can make a speech recognizer's performance drastically worse, the decision trees might be able to cope with this noise if the entropy they use is the entropy of all the *phonemes* at a node, rather than simply the words. The entropy calculation would calculate the entropy of the phoneme sequence rather than treating each word as an atomic symbol. This would clump words with similar phoneme sequences together, since this would produce a lower entropy than grouping words with very different phoneme sequences. Ideally, the computation would produce a smaller entropy for sequences of phonemes that share articulatory features.

Such a procedure might help clump together words that share prefixes, suffixes, or roots, increasing the effectiveness of any modifications designed to deal with phrases and find repeated structures (see above). This would also be a necessary step in moving toward a system that could perform word segmentation and semantics learning at the same time, if that is desirable.

8.5.4 Learning action verbs

Emily Bernier and Lance Cai, two undergraduates at Yale, have shown in unpublished work how TWIG might be used to learn words for verbs having to do with relative motion. In their experiments, two subjects equipped with Cricket beacons enacted one of four patterns, which they called

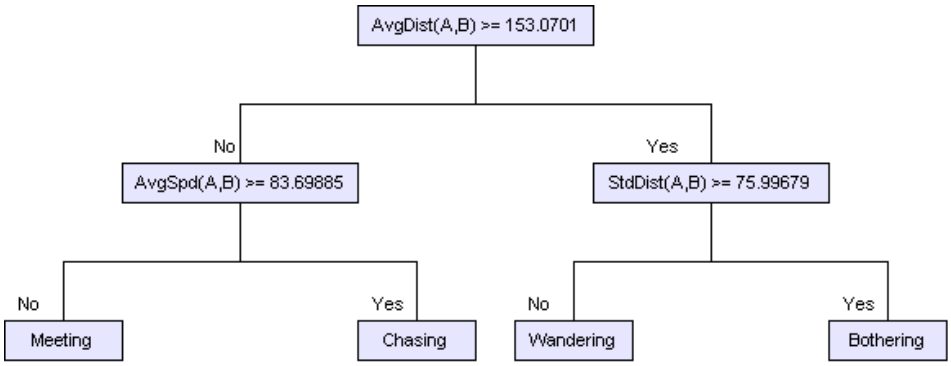

Figure 8.1: The tree generated in the experiments of Emily Bernier and Lance Cai, using the decision tree component of TWIG, from their unpublished project report.

"meeting," "chasing," "wandering," and "bothering." (They did not use the extension inference part of TWIG or speech recognition, but provided the word pairs directly to the definition tree generator.) As features, they included the relative speed, relative distance, and relative heading of the two referents, averaged over the whole period of observation, as well as the standard deviations of the speed and distance. The tree that was generated is shown in Figure 8.1.

Though this tree seemed fairly accurate in its classification performance, incorrectly labeling only two of their test trials drawn from the same four categories, there is probably a better way to characterize movement patterns, since these definitions do not actually capture the temporal structure of each action. "Meeting," for instance, consisted of the two actors converging and then proceeding to travel as a unit, while "bothering" consisted of the first agent repeatedly approaching the second, only to have the second agent move away. Ideally, one would like a movement representation that can capture these changes over time.

A related issue is that by bypassing the extension inference module, these students have also bypassed a key conundrum in learning verbs: how can the system determine over what timespan a verb is meant to refer? Though the extension inference system is capable of determining the agents to which these verbs refer, it is not currently capable of making inferences about temporal reference, and it is not clear how it would produce the action endpoints that the students assumed in their experiment. Thus, while Bernier and Cai have made a valuable first pass at this problem, there are

interesting outstanding problems here.

8.5.5 Learning plurals, nouns, and superordinate categories with sets

Plurals and the plural pronouns (e.g., "we") are interesting from a theoretical perspective because their referents are collections of people or objects instead of individual objects. It thus becomes necessary to rethink the assumption that the extension is a single entity, representable by a single symbol.

It seems to make the most sense to revise the language system to allow words to refer to sets, instead of individual symbols, and to include the number of elements in a set as a possible property to split on in the definition tree. But this raises some subtle issues in deciding the truth or falsity of other properties when they are applied to sets; for instance, when computing the distance `dist({kevin, eli}, ball))`, should the distance be the minimum, maximum, or average of `dist(kevin, ball)` and `dist(eli, ball)`? In general, it seems safest to explicitly specify how each property should apply to a set on a case-by-case basis, and allow properties to hold of the sets themselves rather than assume that they are true of any members of the set.

Adding sets as referents would also add a layer of referential ambiguity, since any time the speaker could be referring to an individual, the speaker may instead be referring to a member of the set that is the referent. For example, any time "I got the ball" was true in the experiments described previously, "We got the ball" would have also been a valid utterance. It therefore would be necessary to deal with referential ambiguity before tackling this problem; TWIG currently requires the extension to be deduced from sentence context, but this would be difficult when every utterance could refer to either an individual or a set.

If a set can be the extension of a word, it might also be possible to learn noun intensions in which the noun is treated as a heterogenous set, and the components of the intension refer to different components of the set. For example, the intension of "face" might include that it has two eyes, a nose, and a mouth; this might translate into a representation where "face" is assumed to refer to a

set, with the eyes and so forth as necessary members of the set.

Another interesting observation here is that young children tend to assume that words for superordinate categories, such as "toy," must refer to collections of objects instead of single objects (Markman and Wachtel, 1988). It is possible that all such words are learned by first learning the plural word and associating it with a heterogeneous set, and only later realizing that when the form of the word demands a singular referent, only one element of the set is intended.

Sets as extensions therefore have the potential to add quite a bit of power to this framework for learning meaning, and I intend to pursue this line of research further.

8.5.6 Visual information, shape bias, and affordances

Another unresolved question is how exactly shape, as opposed to functionality, contributes to a word's meaning. Though some psychologists have reported a bias in infants for using shape as a component of word meaning (Landau et al., 1988), others have argued that this effect is only observed when the affordances (Gibson, 1977) and function of an object are obscured, and that shape bias goes away if the shape can be explained by an object's function (Bloom et al., 1998, cited in Bloom, 2000).

If affordances are quantified in an appropriate way, such that predicates exist that apply to an object iff it possesses a particular affordance, then TWIG could naturally use these affordances as components of intension. Once affordances are accounted for, it is possible that the remaining components of shape that are meaningful will also be relatively simple (size, height to width ratio, roundness, etc.).

Another possibility is that shape is relatively important, but only insofar as it contains quite a bit of information (in the information theoretic sense). Studying how the complexity of an object's shape influences the priority of shape in the definition tree formation could provide a model for how shape influences object classification in infants.

8.5.7 Learning subjective words and interjections

Because TWIG automatically checks all relations to the speaker as possibilities for intensions, it could naturally include the speaker's attitude toward the referent in the meaning of a word, as long as the underlying representation of the world contained some way of representing such subjective relationships. For example, as long as the underlying world description contained the relation `likes(X,Y)`, then whether a word had positive or negative connotation could be added to its intension with the decision `likes(S,X)`. Such a decision would only be added, of course, if the system were exposed to a word which lacked the connotation, since no branch is added to the tree unless there is a contrast involved.

Interjections might also be thought of as referentless predicates on the speaker: "Damn!" conveys the intension `frustrated(S)`, "Ouch!" conveys `hurt(S)`, and so on. TWIG could do this in its current form, as long as interjections were added to the grammar and the robot were given some way to detect such states.

8.5.8 Learning "want" and "know"

Between the time of my early experiments described in Chapter 5 and my development of the logical extension inference and definition trees that would become TWIG, I thought that this thesis would be about learning the meanings of the words "want" and "know." These words are a bit more complicated than the subjective words discussed in Section 8.5.7, since they may require dealing with whole logical propositions as their content.

There may be some interesting research problems in adding these abilities, but I became reluctant to pursue this line of inquiry because all the scenarios I could think of for dealing with "mental facts" seemed overly simplistic. For instance, at one point, I implemented a simple rule in Prolog stating that if person A said something to person B, and B handed an object O over to A in response, then A must have been stating a desire for O. It all seemed a bit too artificial, and I decided I would prefer to wait until I had something genuinely interesting to say about learning "want," rather than

solving a toy problem.

With the addition of new channels for inferring the speaker's mental states – prosody, for instance, or facial expression, or more precise gaze direction – a wealth of interesting words might be learned. However, unless the representation of mental states is relatively rich, the associated word learning problem may not be very interesting, since the primary puzzle – namely, how words are learned for non-visible states – can be easily solved by TWIG as long as there is some module providing the facts pertaining to those states in predicate form.

8.5.9 Salience and other aids to reference

Introducing a larger environment or more referential ambiguity may necessitate adding some notion of salience into the system. Currently the system chooses an arbitrary referent when there is more than one possibility. When ambiguity is uncommon, the few times that reference is incorrectly inferred can be treated as noise.

It is unlikely that this approach would survive the introduction of the ability to refer to the past. When learning a new transitive verb, the possibilities would include every relation that ever held between the two referents; when learning a new noun, the possibilities would include everything that ever participated in the action described by the verb. The system currently implicitly assumes that everything in the immediate environment is equally salient, and everything not in the environment is not at all; obviously, some amount of fine-tuning could be done here.

If a notion of salience were added, care would need to be taken to ensure that evaluation of the salience function were done in such a way that it was computationally tractable. Placing it at the wrong level of the search for a valid parse might result in its being evaluated for many irrelevant cases. It may be interesting to investigate how the extra ambiguity introduced by new words influences these concerns, which exist for parsing even with full knowledge of the vocabulary.

Though this dissertation has emphasized the importance of sentence context for finding reference, the use of pointing gestures, gaze direction, dialog structure, internal motivation, and perceptual

salience could all be used to aid reference finding, and integrating these methods with the use of sentence context would be a worthwhile endeavor.

8.6 Final Thoughts

Though there are plenty of ways that TWIG could be extended and improved, TWIG is a significant advance in the way word learning is implemented on a robotic platform, and presents a more nuanced view of learning new words in general. It is the first system implemented on a robotic platform to generate a complete compositional semantics for each word that can include conjunction, negation, numerical thresholds, multiple meanings, and deixis, all without supervised feedback and in the presence of noise. Its contributions include a proper treatment of extension and intension in word learning, a reframing of the word learning problem that results in more appropriate definitions, and a demonstration of how grammar and logic can work in conjunction with machine learning methods to produce a more nuanced semantics than would be possible with either methodology alone.

Perhaps most importantly, TWIG points the way toward a kind of word learning system that can use its linguistic knowledge effectively to bootstrap an ever-improving knowledge of language. The case of learning first words is interesting from a cognitive science perspective, but it seems as if infants learn far more words once they already know a few words. The theorist in me suggests that the "inductive step" of how to go from a vocabulary of n words to $n+1$ words is far more important than the "base case" of first words, if the number of words the system will learn is to be unbounded.

When I set out on this research project, I had no particular plan to build a general word learning system, capable of learning prepositions and "this" and "that" and embodying a Principle of Contrast while using compositional grammar. TWIG is the result of two puzzles to which I genuinely did not know the answer when I began: how could a robot learn the meanings of the words "I" and "you," if it never received an example in which it was the speaker? And, how could the same system learn the meaning of "he," conjunctions and negation and all, with no negative feedback? I count myself

lucky to have stumbled upon puzzles that provided such a useful perspective on the word learning problem.

Bibliography

N. Akhtar, F. Dunham, and P. J. Dunham. Directive interactions and early vocabulary development: The role of joint attentional focus. *Journal of Child Language*, 18:41–49, 1991.

E. S. Andersen, A. Dunlea, and L. S. Kekelis. Blind children's language: resolving some differences. *Journal of Child Language*, 11:645–664, 1984.

A. Bailey, W. Phillips, and M. Rutter. Autism: towards an integration of clinical, genetic, neuropsychological, and neurobiological perspectives. *Journal of Child Psychology and Psychiatry*, 37: 89–126, 1996.

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

Simon Baron-Cohen. *Mindblindness*. MIT Press, Cambridge, MA, 1995.

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

Paul Bloom, L. Markson, and G. Diesendruck. Origins of the shape bias. Unpublished manuscript, 1998.

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36(4): 929–965, 1989.

J.N. Bohannon, III. Transcript /eng-usa/bohannon/bax/leah1.cha, 1976. Available through the CHILDES project: http://childes.psy.cmu.edu/.

J.N. Bohannon, III and A. L. Marquis. Children's control of adult speech. *Child Development*, 48: 1002–1008, 1977.

Gary Bradski, Adrian Kaehler, and Vadim Pisarevsky. Learning-based computer vision with Intel's open source computer vision library. *Intel Technology Journal*, 9(1), 2005. Available online.

Martin Braine. The acquisition of language in infant and child. In C. E. Reed, editor, *The learning of language*, pages 7–95. Appleton-Century-Crofts, New York, 1971.

M. A. Bremer. Experience in the use of an inductive system in knowledge engineering. In A. E. Hart, editor, *Research and development in expert systems*. Cambridge UP, Cambridge, 1985.

M. R. Brent and T. A. Cartwright. Distributional regularity and phonotactic constraints are useful for segmentation. *Cognition*, 61:93–125, 1996.

Michael R. Brent. An efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning Journal*, 34:71–106, 1999.

Rodney Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, 6:3–15, 1990.

R. Brown, R. P. Hobson, A. Lee, and J. Stevenson. Are there 'autistic-like' features in congenitally blind children? *Journal of Child Psychology and Psychiatry*, 38:693–703, 1997.

Roger Brown and Camille Hanlon. Derivational complexity and order of acquisition in child speech. In John R. Hayes, editor, *Cognition and the Development of Language*, pages 11–53. John Wiley and Sons, New York, 1970.

Roger W. Brown. Linguistic determinism and the part of speech. *Journal of Abnormal and Social Psychology*, 55(1):1 – 5, 1957.

Rudolf Carnap. *Meaning and Necessity*. University of Chicago Press, Chicago, 1947.

E. Clark. The principle of contrast: A constraint on language acquisition. In B. MacWhinney, editor, *Mechanisms of language acquisition*, pages 1–33. Lawrence Erlbaum Assoc., Hillsdale, NJ, 1987.

E. V. Clark. From gesture to word: on the natural history of deixis in language acquisition. In J. S. Bruner and A. Garton, editors, *Human growth and development: Wolfram College lectures 1976*. Oxford UP, Oxford, 1978.

Eve V. Clark. What's in a word? On the child's acquisition of semantics in his first language. In T. E. Moore, editor, *Cognitive development and the acquisition of language*, pages 65–110. Academic Press, New York, 1973.

Eve V. Clark. *First Language Acquisition*. Cambridge UP, New York, 2003.

Christopher Crick, Matthew Munz, and Brian Scassellati. Synchronization in social tasks: Robotic drumming. In *International Symposium on Robot and Human Interactive Communication*, Hereford, England, 2006.

Christopher Crick, Marek Doniec, and Brian Scassellati. Who is IT? Inferring role and intent from agent motion. In *Proceedings of the 6th International Conference on Development and Learning*, London, UK, 2007.

P. S. Dale and C. Crain-Thoreson. Pronoun reversals: who, when, and why. *Journal of Child Language*, 20:573–579, 1993.

Davis and Mermelstein. Comparison of parametric representations for monosyllable word recognition in continuously spoken sentences. *IEEE Transactions on Acoustic, Speech and Signal Processing*, 28(4), 1980.

Carl G. de Marcken. *Unsupervised language acquisition*. PhD thesis, MIT, 1996.

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

Alan Fern, Robert Givan, and Jeffrey Mark Siskind. Specific-to-general learning for temporal events with application to learning event definitions from video. *Journal of Artificial Intelligence Research*, 17:379–449, 2002.

Jerry Fodor. Searle on what only brains can do. *Behavioral and Brain Sciences*, 3:431–432, 1980.

S. Fraiberg and E. Adelson. Self-representation in language and play. In S. Fraiberg, editor, *Insights from the blind*. Basic Books, New York, 1977.

Gottlob Frege. On sense and reference. In Arthur Sullivan, editor, *Logicism and the Philosophy of Language: Selections from Frege and Russell*, pages 175–192. Broadview Press, Peterborough, Ontario, Canada, 1892/2003.

Y. Freund and R. E. Shapire. Experiments with a new boosting algorithm. In *Machine learning: Proceedings of the thirteenth International Conference*, pages 148–156, San Francisco, 1996. Morgan Kaufman.

J. Garofalo. Getting started with the DARPA TIMIT CD-ROM: An acoustic phonetic continuous speech database. Technical report, National Institute of Standards and Technology (NIST), Gaithersburg, MD, 1988.

J. J. Gibson. The theory of affordances. In Robert E. Shaw and John Bransford, editors, *Perceiving, Acting, and Knowing: Toward and ecological psychology*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1977.

Kevin Gold and Brian Scassellati. A robot that uses existing vocabulary to infer non-visual word meanings from observation. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI-07)*. AAAI Press, 2007a.

Kevin Gold and Brian Scassellati. Audio speech segmentation without language-specific knowledge. In *Proceedings of the 28th annual meeting of the Cognitive Science Society*, Vancouver, Canada, 2006a.

Kevin Gold and Brian Scassellati. A Bayesian robot that distinguishes "self" from "other". In *Proceedings of the 29th Annual Meeting of the Cognitive Science Society (CogSci2007)*, New York, NY, 2007b. Psychology Press.

Kevin Gold and Brian Scassellati. Learning acceptable windows of contingency. *Connection Science*, 18(2), 2006b.

Kevin Gold and Brian Scassellati. Deictic pronoun learning and mirror self-identification. In *Proceedings of the 6th International Conference on Epigenetic Robotics (EpiRob-06)*, pages 49–54, 2006c.

Kevin Gold and Brian Scassellati. Using context and sensory data to learn first and second person pronouns. In *Human-Robot Interaction 2006*, Salt Lake City, Utah, 2006d.

Kevin Gold and Brian Scassellati. Grounded pronoun learning and pronoun reversal. In *Proceedings of the 5th International Conference on Development and Learning*, Bloomington, IN, 2006e.

Kevin Gold, Marek Doniec, and Brian Scassellati. Learning grounded semantics with word trees: Prepositions and pronouns. In *Proceedings of the 6th International Conference on Development and Learning*, London, UK, 2007.

H. P. Grice. Logic and conversation. In P. Cole and J. Morgan, editors, *Syntax and Semantics Vol. 3: Speech Acts*, pages 41–58. Academic Press, New York, 1975.

P. Grünwald. A tutorial introduction to the Minimum Description Length principle. In P. Grünwald, I. J. Myung, and M. Pitt, editors, *Advances in minimal description length: Theory and applications*. MIT Press, 2005.

Richard W. Hamming. *Coding and Information Theory*. Prentice-Hall, Englewood Cliffs, NJ, second edition, 1986.

Stevan Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

M. Harris, D. Jones, and J. Grant. The nonverbal content of mothers' speech to infants. *First Language*, 4:21–31, 1983.

Shirley Brice Heath. *Ways With Words: Language, life, and work in communities and classrooms*. Cambridge UP, New York, 1983.

Matthew Herberg. Development of a vision system for a humanoid robot, 2002. unpublished senior project report; http://zoo.cs.yale.edu/classes/cs490/01-02b/herberg.matthew.meh34/.

Daniel Jurafsky and James H. Martin. *Speech and Natural Language Processing*. Prentice Hall, Upper Saddle River, NJ, 2000.

Rohit J. Kate and Raymond J. Mooney. Learning language semantics from ambiguous supervision. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI-07)*, Menlo Park, CA, 2007. AAAI Press.

Michael J. Kearns and Umesh Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA, 1994.

Kevin C. Klement. *Frege and the Logic of Sense and Reference*. Routledge, New York, 2002.

B. Landau, L. B. Smith, and S. S. Jones. The importance of shape in early lexical learning. *Cognitive development*, 3:299–321, 1988.

B. Landau, L. B. Smith, and S. S. Jones. Object shape, object function, and object name. *Cognitive Development*, 5:287–312, 1998.

T. Landauer and S. Dumais. A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

Averill M. Law and David Kelton. *Simulation Modeling and Analysis*. McGraw-Hill, New York, 3rd edition, 2000.

C. Lord, D. Merrin, L. Vest, and K. M. Kelly. Communication behavior of adults with an autistic 4-year-old boy and his nonhandicapped brother. *Journal of Autism and Developmental Disorders*, 13:1–17, 1983.

Catherine Lord and Rhea Paul. Language and communication in autism. In Donald J. Cohen and Fred R. Volkmar, editors, *Handbook of Autism and Pervasive Development Disorders*, pages 195–225. Wiley, New York, second edition, 1997.

K. Loveland and S. Landry. Joint attention and language in autism and developmental language delay. *Journal of Autism and Developmental Disorders*, 16:335–349, 1986.

Andrew Lovett and Brian Scassellati. Using a robot to reexamine looking time experiments. In *Proceedings of the 4th International Conference on Development and Learning*, San Diego, CA, 2004.

Brian MacWhinney. *The CHILDES project: Tools for analyzing talk*. Lawrence Erlbaum Associates, Mahwah, NJ, third edition, 2000.

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

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999b.

Gary F. Marcus. Negative evidence in language acquisition. *Cognition*, 46:53–85, 1993.

E. M. Markman. *Categorization and naming in children*. MIT Press, Cambridge, MA, 1989.

Ellen M. Markman and Gwyn F. Wachtel. Children's use of mutual exclusivity to constrain the meanings of words. *Cognitive Psychology*, 20:121–157, 1988.

E. F. Masur. Maternal labeling of novel and familiar objects: Implications for children's development of lexical constraints. *Journal of Child Language*, 24:427–439, 1997.

David McNeill. Developmental psycholinguistics. In F. Smith and G. Miller, editors, *The genesis of language: a psycholinguistic approach*, pages 15–84. MIT Press, Cambridge, MA, 1966.

Philip Michel, Kevin Gold, and Brian Scassellati. Motion-based self-recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004.

N. J. Minshew and G. Goldstein. Is autism an amnesic disorder? Evidence from the California Verbal Learning Test. *Neuropsychology*, 7:209–216, 1993.

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

James Morgan, Katherine Bonamo, and Lisa Travis. Negative evidence on negative evidence. *Developmental Psychology*, 31:180–197, 1995.

Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4): 541–580, April 1989.

K. Nelson. Structure and strategy in learning to talk. *Monographs of the Society for Research in Child Development*, 38:1–137, 1973.

Tim Oates. PERUSE: An unsupervised algorithm for finding recurring patterns in time series. In *Proceedings of the International Conference on Data Mining*, pages 330–337. IEEE, 2002.

Tim Oates, Tom Armstrong, Justin Harris, and Mark Nejman. On the relationship between lexical semantics and syntax for the inference of context-free grammars. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI-04)*, San Jose, CA, 2004. MIT Press.

Y. Oshima-Takane, E. Goodz, and J. L. Derevensky. Birth order effects on early language development: do secondborn children learn from overheard speech? *Child Development*, 67:621–634, 1996.

Y. Oshima-Takane, Y. Takane, and T. Shultz. The learning of first and second person pronouns in English: network models and analysis. *Journal of Child Language*, 26:545–575, 1999.

Yuriko Oshima-Takane. Children learn from speech not addressed to them: the case of personal pronouns. *Journal of Child Language*, 15:95–108, 1988.

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

Steven Pinker. *The Language Instinct*. HarperCollins, New York, 1994.

Nissanka Bodhi Priyantha. *The Cricket Indoor Location System*. PhD thesis, Massachusetts Institute of Technology, 2005.

James Pustejovsky. *The Generative lexicon*. MIT Press, Cambridge, MA, 1995.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–296, 1989.

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

J. Rissanen. Modeling by shortest data description. *Automatica*, 14:45–471, 1972.

Eleanor Rosch. On the internal structure of perceptual and semantic categories. In T. E. Moore, editor, *Cognitive development and the acquisition of knowledge*, pages 346–405. Academic Press, New York, 1973.

Deb Roy. Semiotic schemas: A framework for grounding language in action and perception. *Artificial Intelligence*, 167:170–205, 2005.

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

Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, New Jersey, 2nd edition, 2003.

J. R. Saffran, R. N. Aslin, and E. L. Newport. Statistical learning by 8-month-old infants. *Science*, 274:1926–1929, 1996.

John Searle. Minds, brains, and programs. *Behavioral and Brain Sciences*, 3:417–424, 1980a.

John Searle. Intrinsic intentionality. *Behavioral and Brain Sciences*, 3:450–456, 1980b.

Jeffrey Siskind. Grounding language in perception. *Artificial Intelligence Review*, 8:371–391, 1995.

Jeffrey Mark Siskind. Lexical acquisition in the presence of noise and homonymy. In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94)*, pages 760–766. MIT Press, 1994.

Linda Smith and Chen Yu. Infants rapidly learn word-referent mappings via cross-situational statistics. In *Proceedings of the 28th annual meeting of the Cognitive Science Society*, Vancouver, Canada, 2006.

Nancy N. Soja. Inferences about the meanings of nouns: The relationship between perception and syntax. *Cognitive Development*, 7:29–45, 1992.

Z. Solan, D. Horn, E. Ruppin, and S. Edelman. Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences*, 102(33):11629–11634, 2005.

E. L. Stine and J.N. Bohannon, III. Imitations, interactions, and language acquisition. *Journal of Child Language*, 10:589–603, 1983.

Ganghua Sun and Brian Scassellati. Reaching through learned forward model. In *Proceedings of the 2004 IEEE-RAS/RSJ International Conference on Humanoid Robots*, Santa Monica, CA, 2004.

M. Taylor and S. A. Gelman. Adjectives and nouns: Children's strategies for learning new words. *Child Development*, 59:411–419, 1988.

Isabelle Tellier. Meaning helps learning syntax. In *Grammatical Inference: 4th International Colloquium, ICGI-98*, Lecture Notes in Computer Science 1433. Springer, 1998.

M. Tomasello and J. Todd. Joint attention and lexical acquisition style. *First Language*, 4:197–212, 1983.

Alan Turing. Computing machinery and intelligence. *Mind*, 49:433–460, 1950.

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

Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. Sphinx-4: A flexible open source framework for speech recognition. Technical Report TR-2004-139, Sun Microsystems, November 2004.

S. Wang and J. Siskind. Image segmentation with ratio cut. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):675–690, 2003.

Terry Winograd. *Computer Program for Understanding Natural Language*. PhD thesis, MIT, 1971.

F. Yates. Contingency table involving small numbers and the chi square test. *Journal of the Royal Statistical Society (Supplement)*, 1:217–235, 1934.

Chen Yu. Learning syntax-semantics mappings to bootstrap word learning. In *Proceedings of the 28th annual meeting of the Cognitive Science Society*, Vancouver, Canada, 2006.

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