Learning Grounded Semantics With Word Trees: Prepositions and Pronouns

Kevin Gold
Department of Computer Science
Yale University
51 Prospect St.
New Haven, CT, USA

Marek Doniec
Department of Computer Science
Yale University
51 Prospect St.
New Haven, CT, USA

Brian Scassellati
Department of Computer Science
Yale University
51 Prospect St.
New Haven, CT, USA

***Abstract*— The authors present a method by which a robot can learn the meanings of words from unlabeled correct examples in context. The "word trees" method consists of reconstructing the speaker's decision process in choosing a word. The facts about an object and its relation to other objects that maximally reduce the uncertainty (entropy) of word choice become the decision nodes of this tree. The conjunction of the choices leading to a word becomes its logical definition. Definitions thereby become only as complex as is necessary to distinguish words in the vocabulary, making the method appear to follow a heuristic that developmental psychologists call the "Principle of Contrast." Combined with a method for inferring word type and reference, the method produces semantics complete enough to produce or understand full sentences. The method was implemented on a robot with visual, auditory, and positional sensors, and succeeded in learning the differences between "I," "you," "he," "this," "that," "above," "below," and "near."**

I. Introduction

How is it possible to learn the meanings of words without negative examples? Young children usually are not corrected when they misuse words, and when they are corrected, they appear to ignore the evidence [1]. Moreover, children can learn language almost entirely without explicit instruction [2], suggesting that they must learn primarily from correct examples of word use.

Learning from positive examples alone can be difficult because an algorithm that conservatively sticks to its examples, or even distributions about those examples, will never generalize in a human-like manner. For example, once something is "hot," there is no point beyond which increasing the temperature makes it cease to be called "hot"; a young learner does not need examples of infinitely hot items to learn this. If this seems intuitive, recall that many learning algorithms instead seek to impose normal distributions or finite boundaries on concepts.

There are more conditions that we might want for a general word learning system. It should be able to learn definitions involving conjunctions: "this" is both something that is close and is an object instead of a person. It should be able to learn terms of *deixis*, such as "here" or "I," which are defined in part by the situation of the speaker. It should handle *polysemy*, or multiple meanings for the same word. It should be usable both for comprehension and production of sentences, not just isolated words. If two words exist for a concept, it should choose the better one. It should make maximal use of its existing knowledge of grammar and vocabulary. It should handle misheard statements gracefully, since noise and error is a problem faced by infants and robots alike. And finally, it should be able to learn a wide variety of classes of words: prepositions, nouns, pronouns, verbs.

Below, we shall describe a system that addresses all of these issues. We propose that the learner's goal should be not to learn when words are "right" and "wrong," but to reconstruct the decisions that the speaker made in choosing a word for a particular object. When the time comes for the learner to choose a word for an object, the learner can follow the reconstructed decision process to come to the correct word. To understand a word, the learner can work backwards, positing the decision outcomes that led to that choice of word. For example, "he" might imply that the referent is not the speaker, or else the speaker would have said "I," and not the addressee, or the speaker would have chosen "you."

These principles can be combined with the principles of formal semantics [3] to create a flexible system for learning the meanings of words. Simply by observing the experimenters describe various situations using full sentences, the system learned logical definitions for the words "I," "you," "he," "this," "that," "above," and "below," in such a way that they could be used for either production or comprehension.

No previous word learning system has been able to handle all of the requirements mentioned above. Word learning systems that incorporated some kind of grounded semantics have typically restricted their inquiry to a particular part of speech [4]–[6], restricted concepts to finite regions of the concept space [4], [7], could not handle deixis or polysemy [4], [7], had no way to deal with numerical data [8], [9], or could not understand or produce complete sentences [4]–[9]. In exchange, these systems have often provided good algorithms for word segmentation [4], [7], [8] and/or visual classification

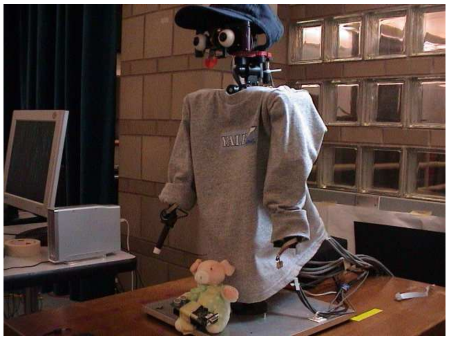

Fig. 1. The robot Nico, on which the word learning system was implemented.

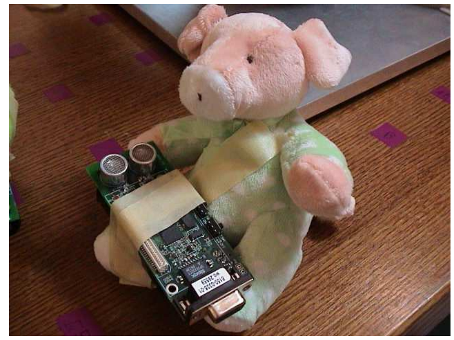

Fig. 2. The pig toy used in the experiment, attached to the Cricket sensor the robot used to find it.

[4], [6], [7], two issues that will not be addressed here.

II. Robotic Implementation

Our word learning system is built to be modular, and function with any robot that can provide its environment information in predicate form. Nevertheless, we will begin by presenting the details of the sensory systems of our robot, Nico (Figure 1). In doing so, we hope to make the abstract algorithms that follow a bit more concrete.

A. Vision: Faces and gaze direction

The robot's visual system was used for finding people and determining the directions they faced. A wide-angle CCD camera in Nico's right eye grabbed 320×240 images at 30 frames per second. These frames were passed to two separate face detectors using the Viola and Jones object finding algorithm [10]. One face detector was used to find profile faces, the other, faces looking directly at the robot. Output from these two face detectors was combined using the forward algorithm [11] to give an estimate of whether a person was more likely to be looking to the side or directly at the robot at a given time.

For each face detected in the visual scene, a new symbol `perL` or `perR` was added to the logical representation of the environment, corresponding to whether the person was detected on the left or ride side of the visual field. (We assumed there would be at most one person on either side of the robot to ease audio localization.) For each of these symbols, `person(X)` was added to the environment. In addition, `lookingAt(X,Y)` (short for "possibly looking at") was true for any pair X, Y such that Y was in the half-space 30 cm away from X, on the other side of the plane to which X's looking direction was perpendicular.

The robot also always had its own symbol, `nico`. The robot assumed `person(nico)` and `lookingAt(nico,Y)` for any Y in its visual field.

B. Sensor networks: Object localization and distance

To find objects and distances between them, we used the Cricket Indoor Location System [12]. Two objects, a tennis ball and a plush toy that looked like a pig were equipped with Cricket receivers (Figure 2), while the ceiling of our laboratory was equipped with 9 Cricket beacons arranged in a 3×3 grid, each roughly 1.5 m from its neighbors. The beacons and receivers communicated via radio signals to determine distances between each, and the robot performed a least-squares calculation to determine the location of each object in 3-dimensional space.

For each sensor-object, a symbol `obj1` or `obj2` was added to the environment. The distance in centimeters between each entity in the environment was then calculated and added to the environment with the `dist(X,Y,V)` predicate, e.g., `dist(perL, obj1, 30.5)`. For the purpose of this calculation, the faces described earlier were assumed to be a fixed distance of 60 cm from the robot, since the vision system did not have access to visual depth information.

In addition, the absolute height above the ground for each object could be computed from the Cricket sensors. The difference `height(X) - height(Y)` for each object pair (X, Y) was encoded as the predicate `relHeight(X,Y,V)`.

C. Audio: Speaker detection and speech segmentation

A dual-channel microphone was used to determine the speaker of each utterance. The two microphone heads were placed 30 cm apart and 50 cm in front of the robot. Within the Sphinx-4 speech recognition system, whether the speaker was to the left or right was determined by comparing the average volume of the two channels over time. The symbol for the corresponding person, `perL` or `perR`, was then bound to the variable S (for speaker).

For the purpose of speech recognition, we gave Sphinx a context-free grammar containing only the words in our experiment. The grammar allowed all pronouns and nouns to be interchangeable.

D. Other predicates and predefined words

To perform its inferences, the system needed some existing vocabulary so that it could understand the context of new words. Five definitions were given to the system. "Pig" and "ball" were defined as $\lambda X.obj1(X)$ and $\lambda X.obj2(X)$, and these properties were always true of `obj1` and `obj2`, respectively. For symmetry, each other symbol was also given a predicate of the same name that was true uniquely of itself. "Is" was defined as $\lambda X.ident(X,X)$, the identity property; every object held this relation with itself. The word "got" was defined as $\lambda X.\lambda Y.got(X,Y)$ where $got(X,Y)$ iff $dist(X,Y) < 30$ cm and $X \neq Y$. The article "the" was defined as a function that looked for an instance of a noun in the environment. Thus, at the beginning of the experiment, the system could understand such sentences as "The pig got the ball" as meaning `dist(obj1, obj2) < 30 cm` or "The pig is the pig" as `ident(obj1, obj1)`.

III. Word trees

A. Inferring type from sentence context

Once the robot has recognized an utterance and changed it into text, it attempts to parse it in Prolog using a simple discrete clause grammar, adapted from [13]. Our parser additionally requires that the system can match all the nouns and pronouns to objects in its environment, and that the verb is one it has encountered previously.

If this parse fails, the robot then proceeds to search for a mapping between a word and an object, or a word and a relation, that will make the sentence parse correctly. We refer to this process as finding the *extension* of the word [3]. Whether the system searches for an object or a relation depends on the grammatical role the new word appears to play in the sentence. For example, if the robot hears "The pig foo the ball," and it knows the words "pig" and "ball," then it can infer that the overall logical form of the sentence is `foo(X,Y)` with "X" bound to the pig in this case and "Y" bound to the ball. On the other hand, if it hears "The pig is foo," it assumes that the unknown word has the logical form $\lambda X.$ `foo(X)` with "X" bound to the pig.

The search continues until the reference provided creates a meaning that the robot can match with its sensory knowledge. For example, if the robot hears "Foo got the ball" and only one person has a ball, the extension of "foo" will be assumed to be the person who has the ball. In this way, even if the speaker is neither looking nor pointing at the referent, the robot can use sentence context to determine the reference of the unknown word.

For further implementation details, we refer the reader to our forthcoming AAAI paper on the TWIG system for inferring extension and type [14]. The important result for the current paper is that the word tree algorithm receives the new word, whether it is binary or unary, and the symbols for the particular objects to which the word refers.

B. Interpreting word trees

New to this paper is the concept of a *word tree* for determining meaning. Word trees can be interpreted as representing the decisions a speaker faces in choosing a word. Then, to understand a word, the system can trace a path from a word back to the root to generate its meaning. Word trees are a kind of decision tree [15], but to our knowledge, this is the first time decision trees have been built for use in the "reverse direction" to create logical formulas.

The reader may consult Figures 3 and 4 for examples of word trees. At the leaves of the tree are the words, while the interior nodes represent decisions about aspects of the referent. Each decision consists of attempting to satisfy a logical predicate with at most one real number, such as `dist(S,X, V)`, indicating the distance between objects S and X is V, and a threshold on the predicate's value, such as $V \leq 30$. (We will sometimes use the shorthand `dist(X,Y) <= V`, or omit mention of the threshold entirely if the attribute is boolean, assuming it to be ≥ 1.) In choosing a word to describe the relationship between objects S and X, the system would decide whether 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 variables in a word tree have a special meaning, because they relate back to the semantics of the word. In formal semantics, nouns and intransitive verbs are represented by lambda functions of the form $\lambda X.word(X)$, while verbs and prepositions tend to have the form $\lambda X.\lambda Y word(X,Y)$. These correspond to the notion that nouns are fundamentally "unary," referring to one thing, while transitive verbs and prepositions are "binary," referring to relations between things. The variables X and Y in the word tree correspond to these variables, which can be bound using partial parses of the sentence. Thus, if the system hears "The dog *foo* the cat," and only lacks a meaning for *foo*, it will make its decisions with X bound to its dog symbol and Y bound to its cat symbol. Trees for unary words and trees for binary words must be kept separate, lest the tree refer to Y when it is undefined in the unary case. In addition to binding X and possibly Y, the variable S is always bound to the speaker.

To look up the meaning of a word, the system finds a leaf that corresponds to the word, and then rebuilds the meaning of the word by following the path back to the root. The meaning of the word is the conjunction of the predicates encountered on the way back to the root, with the predicate negated if the branch used to reach the node is a right (unsatisfied) branch. For example, in the word tree shown in Figure 3c, the meaning of "this" is $\lambda X.\lambda S.\neg$`person(X) & dist(S,X) <= 28.8cm`, indicating that "this" is something that is not a person, but is no more than 28.8 cm away from the speaker.

It is thus straightforward to compile an arbitrary word tree into a Prolog file of logical word definitions, which can then be added to the robot's existing vocabulary for parsing, understanding, and producing utterances.

C. Learning word trees

Since we wish the robot to be able to learn by observing other conversations, the algorithm for learning word trees is unsupervised. The tree is built recursively by accumulating evidence from the world at each leaf, then "splitting" a node into an interior node and two leaves when there is sufficient statistical evidence to do so. The methods used here to decide when to split are from Quinlan's ID3 system [15].

The tree begins as a single leaf node with no associated meaning. On hearing a word in reference to an object, the word, object, speaker, and the state of the environment are stored at this leaf node together. The word is stored as text, the object and speaker are represented by their symbols in the predicate logic, and the state of the environment is represented as a list of predicates generated by the robot at the time of the utterance. We shall refer below to this collection of word, object, speaker, and environment as a single piece of "evidence."

On receiving a new piece of evidence, the node also generates "splitters" based on the evidence: a list of predicates and thresholds that could be used to split all the overall examples seen so far into two groups. These predicates can only refer to the variables X, S, V (for the floating point value), and possibly Y if the tree includes words with "binary" semantics. The thresholds are chosen to be the same values as are present in the evidence itself, with a splitter generated for both the "less than or equal to" and "greater than or equal to" case for each value seen in the evidence. For instance, with the bindings $X = i$ and $Y = j$, then `dist(i, j, 30.0)` would result in the splitters `dist(X, Y, V) & V >=30.0` and `dist(Y, X, V) & V <= 30.0`, among others. The splitters implied by the new evidence are added to an existing pool of splitters generated by previous pieces of evidence.

Next, the algorithm must decide which of these splitters (if any) it will use to split the evidence. This is done by finding the splitter which maximizes the *information gain* provided by the splitter. Let W be the set of all words w_i at the branch, and let $w_i \in W_S$ if it was used under circumstances when splitter S was satisfied, and $w_i \in W_{\neg S}$ otherwise. Then

$$Gain(S) = H(W) - \frac{|W_S|}{|W|}H(W_S) - \frac{|W_{\neg S}|}{|W|}H(W_{\neg S}) \quad (1)$$

where

$$H(W) = \sum_{i:w_i \in W} -\frac{|w_i|}{|W|}\log\frac{|w_i|}{|W|} \quad (2)$$

Readers may recognize $H(W)$ as the *entropy* of W, characterizing the average amount of information in a single word. $Gain(S)$ is the expected reduction in entropy on learning the truth or falsity of S. The splitter with the most information gain is thus the fact about the referent that maximally reduces the "surprise" about the choice of word.

The split is only made final if it is also significant ($p < 0.001$) by the standards of Yates' continuity-corrected chi-square test [16] performed on a $2 \times n$ table of splitter truth vs. word choice. If the split is both informative and significant, the evidence is then divided between the two new children of the node; otherwise, the current node becomes a leaf, with the word it represents determined by simple majority among the evidence stored there. (Using maximal statistical significance alone as a splitting criterion tends to produce trees with a plethora of almost-synonymous definitions for each word when the data set gets large. Distinctions can be highly significant without being terribly informative.)

The tree can be updated online without reconstructing the whole tree. When a new piece of evidence is added to an existing tree, it updates the tables containing the information gain and significance data for the existing splitters at the root, and adds its own splitters to the pool. If this leaves the best splitter unchanged, the root remains the same and the piece of evidence recursively updates the branch of the tree that it satisfies. Thus, the new evidence only needs to update the nodes that it satisfies. However, if the new evidence results in a new best splitter for an interior node, the whole subtree of that node must be remade. Even in this worst case, the running time remains polynomial: each update is linear in the number of pieces of evidence, the number of predicates, the size of the new vocabulary, and the number of thresholds on each predicate. (This last parameter is currently linear in the number of pieces of evidence, but thresholds could be sampled to bound it by a constant.)

If there is a tie for information gain between a boolean predicate and a rational-valued predicate, it is broken in favor of the boolean predicate, on the assumption that these tend to be more "informative" in a broader sense.

IV. EXPERIMENT: I, YOU, HE, THIS, THAT, ABOVE, BELOW, AND NEAR

A. Setup

For 200 utterances, the experimenters 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]*.

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 took care to face the appropriate individual (or robot) when saying "you" or "he."

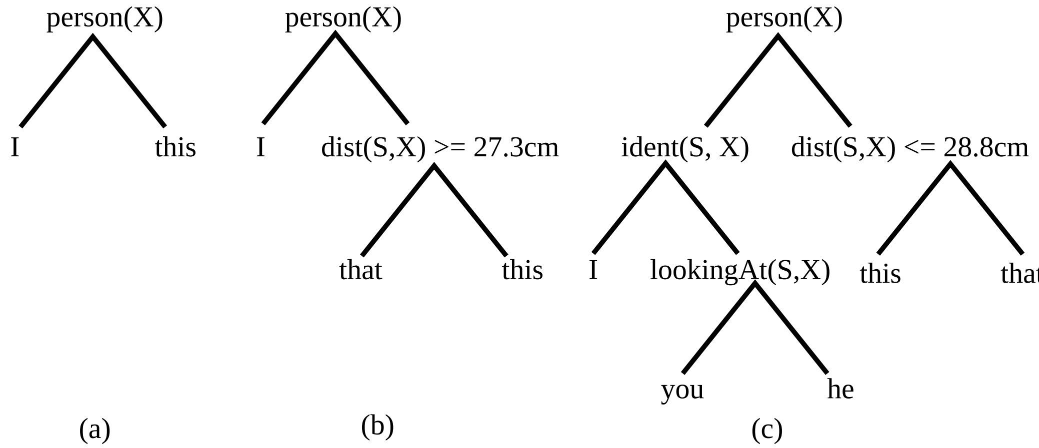

Fig. 3. The word tree that the robot created for pronouns at the (a) 27th, (b) 40th, and (c) final update to the tree. S refers to the speaker, and X to the word's referent; left branches indicate that the logical predicate is satisfied. "ident" indicates the two terms are equal.

B. Results

Many utterances were incorrectly recognized by Sphinx: at least 46%, based on our 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 tautologies (e.g., "That is that"), the system made no inferences from them, and so the errors usually did not affect tree development.

Figure 3 shows the state of the unary 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 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, because the grammar determined that they were binary relations instead of unary descriptions. Unfortunately, phrases including these words were longer than the others, and particularly vulnerable to speech recognition error; by the end of the 200 utterances, Sphinx had recognized only four sentences including "near," and the word tree had yet to split it from "below." However, providing the system with one more example of "near" produced the word tree shown in Figure 4. "Above" and "below" were defined in terms of relative height, while "near" described an object neither particularly higher nor lower than the other. Though "near" could not be defined specifically in the absence of "far," the system had at least learned when "above" or "below" would be more informative.

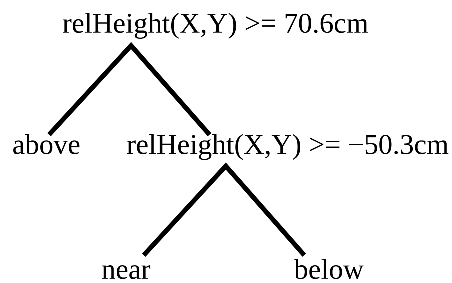

Fig. 4. The tree the system created to define prepositions.

V. Discussion

Word trees solve several conundrums in the automated learning of semantics. They avoid the problem of "negative evidence," because they make the target problem a matter of choice among alternatives, some of which are simply better than others. 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. Finally, 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. Our robot 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 is especially important, because it allows definitions that are more complex than the predicates which the system begins with. When the system is extended to learn concrete nouns, this capability should allow the system to chain simple, low-level visual properties into complex representations of shape.

Some of the definitions implied by Figure 3c may seem too simple, but they must be understood in the context of the robot's conceptual and sensory capacities. "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. 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"), but they do capture some interesting subtleties, such as the fact that "this" should not refer to the addressee even if he is close to the speaker.

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 information-theoretic decisions used in word trees may provide a parsimonious explanation for several different heuristics observed among children. For example, young word learners cease to overextend particular words when they learn more apt ones, a heuristic known as the Principle of Contrast [17]. Similarly, young children reject perfectly good descriptions of objects if there are more appropriate words, a heuristic known as the Principle of Mutual Exclusivity [18]. The act of reasoning backwards about why a particular word was used is sometimes categorized as "theory of mind" [19]. Word choice based on maximal informativeness obeys Grice's Maxim of Quantity [20].

Word tree development can be treated as a model of human word learning, but only if one takes 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. The order in which words, concepts, and situations are encountered plays a large role in determining order of acquisition.

There are still many questions that remain to be explored with this system. How would the system work with raw audio, instead of text from a speech recognizer? What challenges await in learning concrete nouns from visual data? And can the "one unknown word" restriction be removed, so that the system can learn phrases before understanding their parts? These are all exciting avenues for future work.

Acknowledgments

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

References

[1] W. O'Grady, *How Children Learn Language*. Cambridge, UK: Cambridge UP, 2005.

[2] S. B. Heath, *Ways With Words: Language, life, and work in communities and classrooms*. New York: Cambridge UP, 1983.

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

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

[5] D. R. Bailey, "When push comes to shove: A computational model of the role of motor control in the acquisition of action verbs," Ph.D. dissertation, Dept. of Computer Science, U.C. Berkeley, 1997.

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

[7] C. Yu and D. H. Ballard, "A multimodal learning interface for grounding spoken language in sensory perceptions," *ACM Transactions on Applied Perceptions*, vol. 1, no. 1, pp. 57–80, July 2004.

[8] C. G. de Marcken, "Unsupervised language acquisition," Ph.D. dissertation, MIT, 1996.

[9] J. M. Siskind, "Lexical acquisition in the presence of noise and homonymy," in *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94)*. MIT Press, 1994, pp. 760–766.

[10] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[11] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Upper Saddle River, New Jersey: Prentice Hall, 2003.

[12] N. B. Priyantha, "The Cricket Indoor Location System," Ph.D. dissertation, Massachusetts Institute of Technology, 2005.

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

[14] K. Gold and B. 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, 2007.

[15] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

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

[17] E. Clark, "The principle of contrast: A constraint on language acquisition," in *Mechanisms of language acquisition*, B. MacWhinney, Ed. Hillsdale, NJ: Lawrence Erlbaum Assoc., 1987, pp. 1–33.

[18] E. M. Markman and G. F. Wachtel, "Children's use of mutual exclusivity to constrain the meanings of words," *Cognitive Psychology*, vol. 20, pp. 121–157, 1988.

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

[20] H. P. Grice, "Logic and conversation," in *Syntax and Semantics Vol. 3: Speech Acts*, P. Cole and J. Morgan, Eds. New York: Academic Press, 1975, pp. 41–58.