Learning in Wubble World

Wesley Kerr, Shane Hoversten, Daniel Hewlett
Paul Cohen, Yu-Han Chang
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
{*wkerr,shane,dhewlett,cohen,ychang*}*@isi.edu*

***Abstract*— Why do children master language so quickly and thoroughly, whereas gigabytes of text and enormously sophisticated learning algorithms produce at best shallow semantics in machines? Because children have help from competent speakers who relate language to what's happening in the child's environment.**

To facilitate the task of machine word learning, we developed a simulated environment, called "Wubble World," and populated it with entities called *wubbles*. Children interact with the wubbles using natural language, and act as teachers when the wubble needs help. This paper presents our word learning algorithms and provides some empirical results.

***Index Terms*— language, development, online learning, semantics, virtual environment**

I. Learning language in wubble World

The purpose of this project is to have machines learn language in the same way as young children do. Machines learn language models by extracting statistics from gigabytes of text. Children learn language from competent, facilitative speakers who strive to associate their language with what's going on in the child's environment. When a child says, "More!" the parent says, "More milk?" and points to the empty milk glass.

Why do interactions like this produce such rapid mastery of language, whereas gigabytes of text and sophisticated learning algorithms produce at best shallow semantics? The reason is that, to a human language learner, the semantics of sentences are immediately accessible in scenes—in what is going on when the language is uttered. In contrast, a machine, given only text, with no access to scenes, can extract only poor, shallow semantics from distributional statistics, syntactic classes, and other structural features of the text.

To facilitate the task of word learning, we developed a simulated environment, called "Wubble World," and populated it with entities called *wubbles* [9]. Children interact with the wubbles using natural language, and are told to treat them as younger siblings who might need help in understanding what is said to them.

In Wubble World the child is given a task that her wubble must accomplish in order for the child to advance in the game. The wubble is told what to do in English; it can parse sentences, but it doesn't know what words mean. The wubble hears a word, and looks at the scene. If it's uncertain about the word's meaning it asks the teacher a question.

This protocol mimics the way children learn words: the language is grounded - there is a scene to which it refers; it is functional - understanding it lets the wubble interpret its environment; there is a competent speaker to whom the wubble can pose questions; and there is no negative feedback.

Although we describe the protocol in these terms, right now Wubble World is in beta release and not accessible by the general public. Since we are unable to present results acquired using children as teachers, we instead present results acquired using a simulated teacher. The rest of this paper will focus on how the wubble learns, the theory behind it, and our experimental results.

II. Previous Work

Related work begins with SHRDLU, a blocks world created by Terry Winograd [26]. This system generates and understands natural language situated in a simulated world. It could acquire new knowledge and learn about its environment, but it differs from Wubble World in that it is a purely symbolic system.

Gorniak and Roy [8] developed a system in which two subjects share a scene. The first subject selects an object from the scene and describes it to the second subject, who must then try to identify its referent. After collecting the language data generated by the subjects, the authors created a computer model to perform the same task. Finally, Gorniak and Roy created a computer model that uses the data collected in order to perform the same task as the human subjects. This approach of gathering data and refining semantics is similar in some ways to ours, although we use an online model of word learning that allows instruction from a teacher.

Roy [18] also developed a batch-learning system using human language data to generate natural language, using raw image data coupled with natural language descriptions.

There is a significant body of research on the symbol grounding problem in the connectionist network community [21], [14], [7], [17]. Most focus on learning words associated with images, and are trained and tested in batch processes. Sankar and Gorin [20] developed a system similar to a 2D Wubble World, which successfully learned 431 words

from over 1000 conversations. They used a simulator that displayed a set of objects in a scene, and an interface for interacting with an agent. The agent is represented as an eye, and can be directed at any of the objects in the scene using natural language.

Others approaches to subsets of the problem include [12], [11], [25], [15], [17], [24], [23], [19], [5], [4]. However, we know of no attempt to tackle the whole problem — scaffolding language acquisition by facilitative speakers, bootstrapping and gradual acquisition of word meanings, and the interaction of syntax and lexical acquisition — on a very large scale.

III. Learning concepts

Consider the following experiment, similar to that described by Carey and Bartlett [2]. A child and an experimenter are placed in a room containing a set of M unique objects. The experimenter says a word and instructs the child to identify the corresponding object from the set of M in the room, each of which can be described by a perceptual feature vector with 5 attributes: *(color, shape, size-x, size-y, size-z)*. The child wishes to maximize her "reward" by guessing correctly each time. What strategy could the child use to maximize her reward? She needs to both guess at the meanings of unfamiliar words and exploit her knowledge once she is sufficiently confident as to its accuracy.

This is the same problem that wubbles face in our learning environment. In the following section we describe how our system learns concepts described by nouns and adjectives. (In Section IV-B we describe how we learn prepositions.)

First, we describe the input that the wubble receives, and then we describe the wubble's internal representation and learning algorithm. There are two forms of input: the first is the sentence typed from the teacher, and the second is the perceptual input from the scene. The combination of the two different forms of input will guide the wubble in understanding natural language.

A. Input

Children communicate with their wubbles by "speaking" (typing) to them. When a child speaks to a wubble, the wubble "hears" this sentence as a parsed logical form (LF). An example is shown in Fig. 1, produced from the sentence "go to the cylinder."

```
((type imperative) (subj NIL)
 (verb-phrase (verb go) (obj (adj NIL) (noun NIL)))
 (prep-phrase (prep to) (obj (adj NIL) (noun cylinder))))
```

Fig. 1. Sample logical form (LF).

In effect, the wubble is endowed with an innate grammar that divides speech into linguistic classes. The classes we consider in our system are verbs, nouns, adjectives, and prepositions. Since we are able to extract the linguistic class for each word in the sentence, we can write specialized procedures that operate at the level of linguistic classes similar to [8], [26].

In the initial version of our system we assume that the wubble has acquired a basic level of motor proficiency, and a mapping of verb-words to the appropriate motor concepts. Extending the concept learning system for use with verbs is discussed in the conclusion.

The wubble's perceptual system is coarse; it knows its own position, and it can sense objects in the environment, their positions, and their properties. The features are abstract; for example, objects are described by the feature vector *(type, color, size-x, size-y, size-z)*.

B. Internal Representation

The wubble represents objects in its world by maintaining bundles of distributions over the possible values of each feature of an object. For each feature f, the wubble maintains a distribution in the form of a set of weights $W^f = \{w_v^f\}$, one for each possible value $v \in V$, where V is the set of possible values for the f. At any time t, we can derive a probability distribution from W^f:

$$p_v(t) = \frac{w_v^f}{\sum_{j=1}^{K^f} w_j^f} \quad ,$$

where K^f is the number of possible values of the feature f. Thus we will frequently refer loosely to W^f as a "distribution", and the wubble's internal representation of a concept as a bundle of distributions.

For each concept, the wubble instantiates a bundle of distributions that describes the feature values that the wubble believes are likely to be associated with that concept. For example, the concept "globe" would be described by a high probability (or weight) on `sphere` in the feature *type*, and uniform distributions over the possible values of all the other features.

C. Learning

The wubble learns incrementally, online, by updating the weights associated with each feature. At each time t, the wubble is given a new problem defined by a word and a set of possible objects, one of which is described by the word. The wubble wishes to maximize its overall "reward" by choosing the object that matches the given word for each problem.

If the wubble thinks it knows the answer, it picks an object. If it's too unsure to guess, it asks the teacher for help. Once the correct object is identified, the wubble updates its representation for the given word by increasing the weights on the appropriate feature values. Formally, assume the correct object is perceived as a feature vector $(v_1, v_2, \ldots, v_n)$ where v_j is a value of the feature f^j.

Then the wubble updates its bundles of weights according to the following equation:

$$w_i^j(t) \quad = \quad \begin{cases} e^{\gamma} w_i^j(t-1) & \text{if } v_j = i \\ w_i^j(t-1) & \text{otherwise} \end{cases} \qquad (1)$$

where γ is the reward the agent receives for choosing the correct object. This multiplicative updating is similar in flavor to the algorithm described by Freund and Schapire [6], who proposed it in the context of betting, where the gambler wishes to maximize his reward by listening to the advice of his fellow gamblers, or "experts." At each time period, the gambler increases the weight he assigns to a particular expert if the expert's advice leads to a winning bet. The authors show that their algorithm is able to bound the total "regret" that the gambler experiences to within $O(\sqrt{T \ln N})$, where T is the number of time periods played, and N is the number of experts. This means that the total reward received by following this algorithm is close to the reward the gambler could have received had he listened only to the best expert.

Regret minimization techniques are useful in online learning problems because they allow rapid convergence to the correct prediction while guaranteeing low regret. In our setting, we use the multiplicative updating rule for exactly this purpose: to allow the wubble to quickly acquire the correct representation for the given word.

IV. Word Learning

Given the learning framework described in the previous section, we now provide a detailed description of the wubble-teacher interaction and provide an example showing the wubble using this framework to learn. We also discuss concept resolution: so far, we have described how the wubble learns, but the wubble also needs a way to use its representation of a concept to identify the object in the scene that best fits that representation. Nouns and adjectives are presented first, followed by prepositions.

A. Nouns and Adjectives

The combination of nouns and adjectives in the logical form, described in Fig. 1, result in a set of referents in the scene that correspond to the meaning of the words. There are two separate parts to the system; the first is the initial training loop. Within the training loop the wubble is uncertain of the correct answer. The second part of our system works with concept resolution after initial training, i.e. how does the wubble select among the possible referents in a scene?

1) Learning a noun concept: Assume a newborn wubble is in a room as portrayed in Fig. 2. The child types "go to the cylinder." This sentence is parsed into the logical form shown in Fig. 1.

Each wubble has an innate understanding of a small set of verbs, so it understands "go," but it doesn't know what a "cylinder" is. (Assume for now that the wubble also understands "to." Prepositions will be discussed later.) More precisely, the word "cylinder" is shorthand for a cylinder concept, which starts out as a maximum-entropy distribution for each of the features in its sensory experience, as shown in Fig. 3: (entries for size-x and size-z are omitted for brevity).

Fig. 2. A wubble in a room with lots of cylinders.

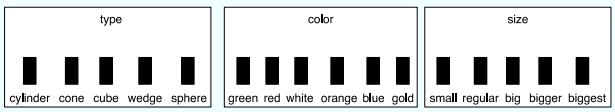


Fig. 3. All features have high entropy attributes.

The wubble will ask for clarification whenever its certainty about a concept is below some threshold, so it asks the teacher: "What is a cylinder?" The teacher responds with one of the cylinders. Note that there are four cylinders in this room; as a result, the wubble will learn a different lesson depending on which particular cylinder the teacher selects. Let's say the teacher chooses the largest cylinder. The wubble then updates its beliefs about the concept "cylinder" to reflect the properties of the cylinder selected by the teacher, using (1). Since the large cylinder has *size*: `bigger`, *type*: `cylinder`, and *color*: `green`, the weights associated with these feature values gets a slight bump, as shown in Fig. 4.

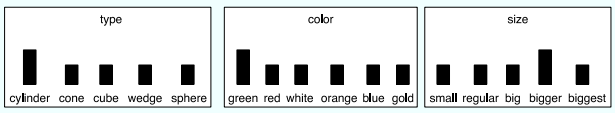


Fig. 4. Features after one training example.

Now imagine that the teacher repeats the request three more times. Since the wubble is not yet sure enough about what a cylinder is, after each request it will ask the teacher: "What is a cylinder?" The teacher will select one of the cylinders in the scene. Let's say the teacher responds to each query by clicking on a different cylinder; in this case, since the only feature value the four cylinders in the scene have in common is *type*: `cylinder`, after being instructed over four training samples the wubble will have a notion of cylinder that includes high-entropy beliefs about the *color* and *size*

features, but a low-entropy belief about the *type* feature, as shown in Fig. 5.

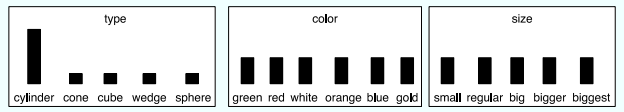


Fig. 5. Features after four examples; *type* feature has low entropy.

As a result of these interactions with the teacher, the wubble is now confident enough that it knows what a cylinder is (in this case, a high-enough probability that the object has *type*: `cylinder`) that subsequent references to "cylinder" can be resolved.

2) Concept Resolution: Once a wubble becomes confident in its knowledge, it begins to resolve words to objects in the environment without aid from its teacher. Imagine that the wubble from the previous example receives another request to go to the cylinder. It is confident enough in its knowledge to "imagine" what a cylinder is. By "imagining" we mean that the wubble creates a prototype of its notion of "cylinder" by sampling from the probability distributions corresponding to each feature associated with the concept.

Each feature of a concept is associated with a probability distribution, P, and the entropy of this discrete probability distribution is $H(P)$. Since this distribution is K probabilities that sum to one, we know $H(P) \leq \log_2 K$. With this in mind, we define the scaled entropy $H'(P)$ as:

$$H'(P) = \frac{-\sum_{p_i \in P} p_i \log_2(p_i)}{\log_2 K}$$

Let A be the prototypical object the wubble imagines, defined by the feature vector $(a_1, a_2, ..., a_n)$, and let B be any other object in the scene, defined by the feature vector $(b_1, b_2, ..., b_n)$. Also, let P_j be the probability distribution for feature j of the concept the wubble used to imagine its prototype. The distance from object A to object B, d_A^B is:

$$x_i = \begin{cases} 0 & \text{if } a_i = b_i \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

$$d_A^B = \sum_{i=1}^{n} x_i (1 - H'(P_i)) \quad (3)$$

This distance function provides an estimate of how similar two objects are. It states that for each feature $1 \ldots n$, the distance contributed by that feature is directly related to the entropy of the probability distribution: the higher the entropy, the less the wubble believes this feature contributes to the meaning of the word, and the less the distance matters.

Now we can combine all the components of the noun phrase and find the best referent. Let S be the set of objects in the scene, and O be the set of prototypical objects for the adjectives and nouns in the noun phrase. The referent of the noun phrase is the object that minimizes the sum of the distances between itself and the set of prototypical objects.

Let's pretend the wubble imagines a prototypical object: `(cylinder, green, small)`. The wubble computes the distance between this prototype and every object in the scene, looking for the best match. In the scene shown in Fig. 2, the two best candidates each share two features with the imagined cylinder; the wubble will pick randomly between the two.

The wubble can now fulfill the teacher's request. If the wubble goes to the correct object, then we reinforce it by updating the weights for each feature in the "cylinder" concept using the update rule defined in (1). If it goes to the wrong object, then instead of trying to sort out negative feedback, the wubble simply asks the teacher for the correct referent named in the utterance and applies positive feedback, as described in (1).

B. Prepositions

Computational models of prepositions [17], [16], [13] (and others) define prepositions as relations that hold between *landmarks* and *trajectors*. Typically the trajector is the object for which the relationship holds and the landmark acts as a grounding point for the relationship.

Our prepositions work much like our nouns and adjectives. When a wubble hears new preposition words, it creates new preposition concepts. We assume that each prepositional phrase has an accompanying noun phrase, which when resolved becomes the landmark of the preposition. Space is described relative to this landmark.

Like those for nouns and adjectives, concepts for prepositions are represented as bundles of features. The dimensions x, y, and z become the features of the preposition concept, each of which can take on one of the values `(negative-far, negative-near, zero, near, far)`. This set provides a coarse way to make continuous space discrete. The actual numerical values represented by the symbols are scaled in proportion to the size of the landmark.

In the same way that a wubble imagines a prototypical object, it imagines a prototypical region of space for the preposition. When the wubble is told to act on that preposition, it samples from the probability distribution of each feature. The three samples are then combined, yielding a region in space. For example, in Fig. 2, the blue cylinder is behind the green cone, occupying space that is described by the features: x = `zero`, y = `zero` and z = `negative-near`.

Using this approach, the wubble can learn many English prepositions, but cannot understand prepositions defined using multiple landmarks, such as "between," and those defined using action, such as "around" and "through" (as [17].) We can, however, understand prepositions that take multiple

words to describe in the English language; for example, a child could train her wubble to understand the prepositional phrase "in-front-of-and-to-the-right-of."

V. Experimental Protocol

To validate our word-learning algorithm, we performed experiments in a number of different environments. The first experiment measured how quickly and with what level of mastery a wubble can learn a vocabulary of nouns and adjectives. The second experiment measured the wubble's mastery of preposition words.

A. Adjective and Noun Vocabulary

We tested a wubble's mastery of adjectives and nouns in two separate types of environments: a *facilitative* environment containing relatively few objects, and a *challenging* environment, which contains 100 objects. It is our hypothesis that the facilitative environment fosters learning since it contains fewer referents for the wubble to reason about.

We generated a set of testing sentences that was held constant across all trials. Each sentence in the testing set is similar in form to the sentence: "choose a small red column." The sentences contain two adjectives, one describing the object's size and the other describing its color, and a noun describing its shape. This set of sentences contained 14 words that the wubble was required to learn.

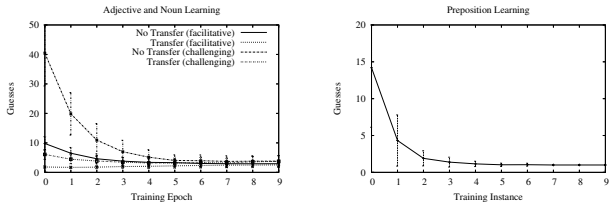
The next step was to populate the room with a set of objects. We ensured that at least one object for each sentence in the test set existed in the room. The rest of the objects in the room were randomly selected (with replacement) from the set of possible objects.

We generated a set of training sentences that provide guidance for a wubble to learn language. For each object in the environment, we generated the sentences that describe it, of the forms "choose a *type*," "choose a *color* object," "choose a *size-[x,y,z]* object," and all possible combinations. For instance, "choose a *color type*" combines the type sentence with the color sentence and results in a new set of referent objects. Each sentence generated by the process can refer to several objects within the scene.

We presented this collection of sentences as training sentences for the wubble, keeping the training and test sentences separate, so that while the wubble may have trained on the *individual* words, it has never experienced the specific combination of words in the testing sentence.

We defined one epoch as a sequence of 10 randomly selected training sentences, and after each epoch the wubble was presented the set of test sentences. Performance was measured by counting the number of guesses required to correctly identify the referent of the sentence. Perfect performance resulted in one guess. One trial is a collection of nine epochs with an initial testing phase prior to training.

The protocol explores our initial hypothesis that *facilitative* environments foster learning. We also ran several experiments



(a) *(b)*

Fig. 6. *(a)* Results for nouns and adjectives. *(b)* Results for prepositions.

to explore knowledge transfer for the wubble by augmenting the wubble with training experiences from a different environment. To simplify the setup of the transfer experiment each wubble started with the correct concept for each color word in the testing sentences. The task was to learn the rest of the vocabulary.

B. Preposition Vocab

Testing a wubble's mastery of prepositions was relatively straightforward. Since the wubble automatically determines the landmark of the preposition we were able to run experiments in a simple environment that contained only one object. This focused the evaluation of performance to the preposition word, rather than identifying the correct landmark. We presented the wubble with one sentence similar to: "choose a point behind the wedge." The challenge was to identify a region of space that is *behind* the wedge.

We cannot combine prepositions in the current system, such as "in front of and to the left of." Therefore, when testing a wubble on prepositions, we needed only one sentence. Each time the sentence was presented, it acted as both a training instance as well as a testing instance. The wubble's performance is the number of different locations tried before identifying the correct location.

VI. Results

The results for adjectives and nouns are displayed in Fig. 6*(a)*, and are an average of 100 trials. These results show that in *facilitative* environments the wubble can learn the meanings of words in fewer training epochs. When we introduced color concepts in the transfer case, the wubble has an automatic leg up because it is able to reduce the space of correct objects and quickly identify referents. By combining both the *facilitative* environment and the transfer case, the wubble is able to correctly identify the referent immediately.

Fig. 6*(b)* shows the results for learning a single preposition over 100 trials. After three sentences the wubble understands the meaning of the preposition.

We found several cases where our system is unable to learn the correct concept, all related to interactions between features. Since feature distributions are treated independently, words defined as a disjunction of values in multiple distributions cannot currently be learned. Examples include

"large" (at least `big` in any dimension) and "far away" (`negative-far` or `far` on any axis).

In spite of this, we *can* learn complex concepts. One of the words trained in the vocabulary above was the complex concept "column." Columns were either tall cylinders or tall rectangular prisms, but both were restricted to be taller than they were wide. Even though this concept provided a more intricate relationship between properties than other concepts such as "sphere," the wubble's performance was not affected.

VII. Conclusions

Based on initial results, we have a system capable of finding objects within scenes using natural language, and all after relatively few *positive* training examples. This is in agreement with the way that children learn language. One of the limitations of the system is our reliance on a statistical parser trained over a large training set [10]. Future work will explore how to remove this reliance and more closely integrate word learning with grammar learning.

Wubbles currently have primitive kinds of bootstrapping going on in their language learning: asking the user to point out an object, or asking whether it is in a prescribed spatial relationship with an object, and using the parse tree of a sentence to identify the possible syntactic classes of words. The challenge now is to implement the many other ways that children bootstrap lexical and syntax learning (e.g., [1], [2]). We have one particularly sneaky — and distinctly non-human — kind of bootstrapping in mind: wubbles will transfer knowledge of language between them, unbeknownst to the children, so they will learn language more quickly than any one child could teach it.

We currently have a spatial model for prepositions; an object model for relating nouns and adjectives to geometry, size and color; essentially no model for interpreting verbs; and no model of intentional states, such as focus of attention, goals, beliefs and plans. We need to add to the models we have and provide those we lack. For verbs, we already have some candidate models [3], [4] and so do others (e.g., [11], [22]). All are based on the *dynamics* of observed actions. We need to incorporate these models into wubbles and adapt regret minimization to learn verb meanings. As to models for intentional states, we think it is a significant research project to provide wubbles with models of their own focus of attention, goals and plans, and even an elementary theory of mind.

References

[1] P. Bloom. *How Children Learn the Meanings of Words*. MIT Press, 2001.

[2] S. Carey and E. Bartlett. Acquiring a single new word. *Papers and Reports on Child Language Development*, 15:17–29, 1978.

[3] P. R. Cohen, C. T. Morrison, and E. Cannon. Maps for verbs: The relation between interaction dynamics and verb use. In *Proceedings of the 19th International Conference on Artificial Intelligence*, 2005.

[4] Paul R. Cohen and Tim Oates. A dynamical basis for the semantic content of verbs. In *Working Notes of the AAAI-98 Workshop on The Grounding of Word Meaning: Data and Models*, pages 5–8, 1998.

[5] E. D. DeJong. The development of a lexicon based on behavior. In *Proceedings of the Tenth Dutch Conference on Artificial Intelligence*, 1998.

[6] Y. Freund and R. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999.

[7] M. Gasser and L. B. Smith. Learning nouns and adjectives: A connectionist account. *Language and Cognitive Processes*, 13(2):269–306, 1998.

[8] P. Gorniak and D. Roy. A visually grounded natural language interface for reference to spatial scenes. *Proceedings of the 5th international conference on Multimodal interfaces*, pages 219–226, 2003.

[9] D. Hewlett, S. Hoversten, W. Kerr, P. Cohen, and Y. Chang. Wubble world. In *Proceedings of the 3rd Conference on Artificial Intelligence and Interactive Entertainment*, page in press, 2007.

[10] D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, 2003.

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

[12] Tim Oates. Grounding word meanings in sensor data: Dealing with referential uncertainty. In *Proceedings of the HLT-NAACL-2003 workshop on Learning Word Meaning From Non-Linguistic Data*, 2003.

[13] P. Olivier and J. Tsujii. A computational view of the cognitive semantics of spatial prepositions. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 303–309, 1994.

[14] K. Plunkett, C. Sinha, MF MøLLER, and O. Strandsby. Symbol grounding or the emergence of symbols? vocabulary growth in children and a connectionist net. *Connection Science*, 4(3-4):293–312, 1992.

[15] T. Regier. A model of the human capacity for categorizing spatial relationships. *Cognitive Linguistics*, 6(1):63–88, 1995.

[16] T. Regier and L. Carlson. Grounding spatial language in perception: An empirical and computational investigation. *Journal of Experimental Psychology: General*, 130(2):273–298, 2001.

[17] Terry Regier. *The Human Semantic Potential*. The MIT Press, 1996.

[18] D. K. Roy. Learning visually grounded words and syntax for a scene description task. *Computer Speech and Language*, 16(3):353–385, 2002.

[19] Deb Roy. *Learning Words from Sights and Sounds: a Computational Model*. PhD thesis, MIT, 1999.

[20] A. Sankar and A. Gorin. Visual focus of attention in adaptive language acquisition. In *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-93*, volume 1, pages 621–624, 1993.

[21] P. G. Schyns. A modular neural network model of concept acquisition. *Cognitive Science*, 15(4):461–508, 1991.

[22] J. Siskind. Grounding lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of AI Research*, 15:31–90, 2001.

[23] J. M. Siskind. A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, 61(1–2):39–91, 1996.

[24] Luc Steels. Perceptually grounded meaning creation. In *Proceedings of the International Conference on Multi-Agent Systems*, 1996.

[25] C. A. Thompson. *Automatic Construction of Semantic Lexicons for Learning Natural Language Interfaces*. PhD thesis, University of Texas, Austin, 1998.

[26] T. Winograd. *Understanding Natural Language*. Academic Press, Inc. Orlando, FL, USA, 1972.