Learning Object-Relative Spatial Concepts in the L_0 Project

Terry Regier
Computer Science Division
University of California at Berkeley
and
International Computer Science Institute
1947 Center Street, Berkeley, CA, 94704
(415) 642-4274 x 184
regier@cogsci.Berkeley.EDU

Abstract

This paper reports on the learning of spatial concepts in the L_0 project. The starting point is the identification of a visual primitive which appears to play a central role in the visually-based semantics for terms which express spatial relations between two objects. This primitive is simply the orientation of the imaginary ray connecting the two related objects where they are nearest each other. Given this, an important part of the learning consists of determining which other orientations this particular one should align with (e.g. it should align with upward vertical for "above"). These other orientations may be supplied by an object-centered coordinate frame, as in English "in front of" and Mixtec "čii", as well as by the upright coordinate frame. A central feature of the system design is the use of orientation-tuned Gaussian nodes which can learn their orientation and σ, and which perform the critical task of orientation comparison.

Introduction

The L_0 project (Feldman et al. 1990; Weber & Stolcke 1990) concerns the computational task of acquiring natural language in the visually-based semantic domain of spatial relations between geometrical objects. The goal is to learn to determine, for any natural language, whether a scene description in that language is true of a particular scene. A significant part of this task is learning the perceptually grounded semantics for the individual spatial terms in the language. Thus, as a subtask, we would like to learn to associate scenes, containing several simple objects, with spatial terms describing the spatial relations in the scene. Languages differ widely in the perceptual features encoded in their spatial terms (Talmy 1983; Bowerman 1989), making this subtask a challenging one.

When learning a particular spatial concept, the system is supplied with a scene, and an indication of which object is the reference object (called the *landmark*, or LM) and which is the object located relative to the reference object (called the *trajector*, or TR). This is illustrated in Figure 1.

Earlier work on this subtask (Regier 1990; Regier 1991) used connectionist mechanisms to learn several basic spatial terms in English, and handled the problem of learning the semantics for these in the absence of explicit negative instances. It did not, however, cover *object-relative* terms, i.e. spatial terms which are sensitive to an inherent orientation that a LM may have, such as "in front of" in English. "In front of" makes reference to a coordinate frame centered in the LM itself: if the LM (a person, for example) has an inherent front, "in front of" is determined relative to that front. This is in contrast to terms like "above", which are unaffected by any inherent orientation of the LM.

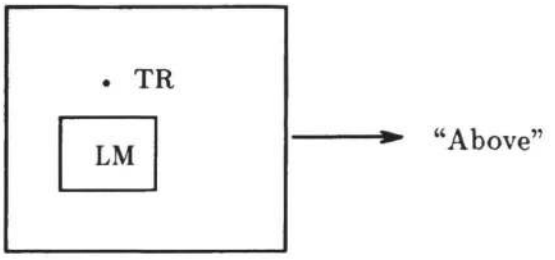

Figure 1: Learning to Associate Scenes with Spatial Terms

This paper presents a way of viewing spatial concepts in which the semantics for object-relative terms and non-object-relative terms are learned in much the same manner. This is based on the identification of a useful visual primitive, and the specification of a mechanism for using it. The system presented here successfully learns the perceptually-based semantics of object-relative terms such as "in front of", as well as that of non-object-relative terms such as "above", "on", and the like.

General Approach

A central assertion of this work is that one of the crucial primitives used in determining spatial relations is the orientation of the (imaginary) ray from the LM to the TR *where they are nearest to each other*. Consider Figure 2. Here we have small circles located relative

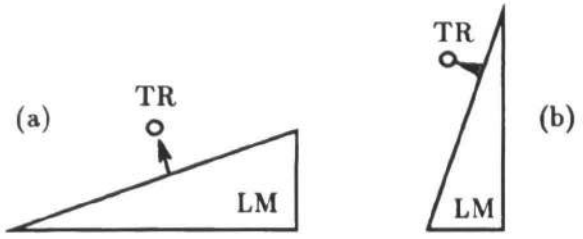

Figure 2: Good and Poor Instances of English "Above"

to two triangles, with the shortest possible connecting line between the LM and TR drawn in. Note that in both cases, the circle is above some part of the triangle. The assertion is that (a) is a better instance of "above" because the orientation of the ray is closer to upward vertical in this case than in (b). This has, at any rate, proven to be a productive working hypothesis. This orientation where the LM and TR are nearest is called the *TR orientation*.

Another potentially useful primitive, to be incorporated in the future, is the orientation of the ray connecting the centers of mass of the two objects (the *CoM orientation*). Figure 2 can actually be seen as supplying evidence that a combination of these two primitives might play a role in the determination of exactly when one object is "above" another. For if the small circle in (a) were moved down the slope of the triangle, it would eventually reach a point at which we would no longer feel comfortable calling the relation between the two objects "above"; however, the TR orientation would have remained unchanged. This can be explained by postulating that both the TR orientation and the CoM orientation must play a part in the determination of "above", and that the CoM orientation in this new case is too far from vertical to allow one to label the scene "above". The CoM orientation is not used in the system presented here; this is primarily to simplify the exposition.

Given orientation primitives of this sort, a large part of the task of learning the perceptually-based semantics for spatial terms is to determine which other orientations in the scene the TR (and/or CoM) orientation aligns with, and to what extent. As a very simple example, if the TR orientation aligns perfectly with upward vertical, we consider this an excellent instance of "above". If it deviates a little from upward vertical, this is a fair instance, and so on. Thus, what is required is an orientation comparison mechanism of some sort. Θ-nodes, to be described in detail below, fulfill this function.

There are in fact two distinct aspects to the process of learning orientations:

- Learning which of several available reference orientations the TR orientation should align with.
- Learning what the values of some of these reference orientations should be. For example, the system starts learning without knowledge of the fact that upward vertical is an important reference orientation. It must learn to tune itself so that upward vertical becomes one of the reference orientations available to the TR orientation for alignment.

These two aspects of the learning occur simultaneously in the system.

Some reference orientations are not learned, however. The orientations of the major and minor axes of the landmark are critical for learning object-relative terms, together with an indication of whether or not one end of each axis is inherently marked as the positive direction (e.g. the positive direction of the major axis for an erect human being is upwards, while the positive direction for the minor axis is toward the ventral side). The values of these orientations clearly change with each new landmark; thus, they are not learned.

Orientation alignment does not suffice to represent such important perceptual features as inclusion and contact. Another crucial primitive is a bitmap representation of the interior of the LM, i.e. a 2-D visual map such that all points that fall in the interior or on the boundary of LM are activated, and all others unactivated. Ideally, a fixed connectionist visual preprocessing stage will produce these primitives for the learning system, given the input image. Currently, they are computed in a non-connectionist fashion.

The system learns spatial concepts by combining evidence from these two forms of representation, orientation-based and bitmap-based. The detailed architecture will be presented below.

The Mixtec Spatial System

As mentioned above, part of the goal of the L_0 project is to build a system that will be able to learn the spatial system of any natural language. For this reason, this paper focuses on the learning of spatial concepts from the Mixtec language, a system radically different from that of English. Both Mixtec and English concepts are learned by the system.

The Mexican Indian language Mixtec has a spatial system based on an extensive body-part metaphor (Brugman 1983), featuring a number of object-relative terms. In general, in Mixtec one would say "The [TR] is located (near/at) the [LM]'s [body-part-name]", even if the LM were not an animate object. Thus, the region above a tree would be described as "(near/at) the tree's head". One intriguing aspect of Mixtec is that it uses human body-part terms to describe relations relative to LMs which are vertically extended, and animal body-part terms for horizontally extended LM objects (since animals such as dogs and horses are generally seen on all fours, horizontally extended). There is a Mixtec term which is used for both humans and animals, however, and therefore, for both vertically and horizontally extended objects. This is the term *čii*, or "belly", which

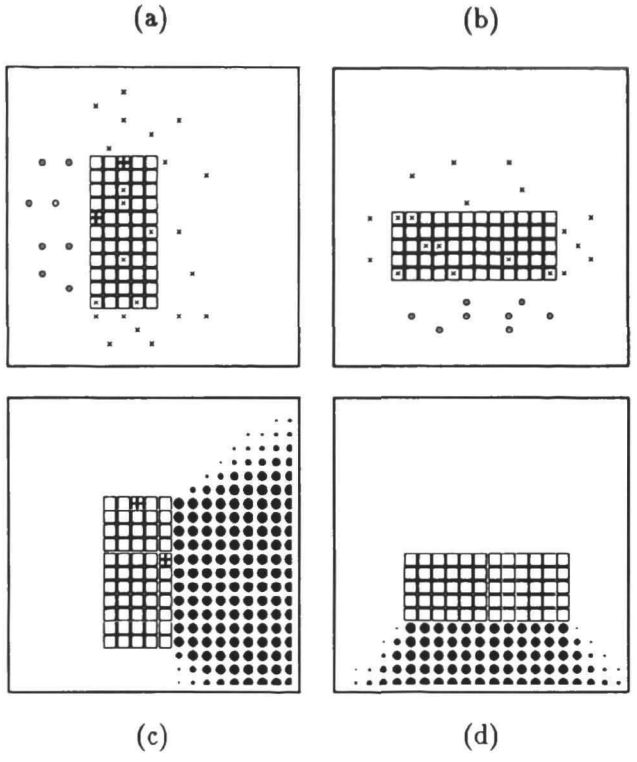

Figure 3: Some Training Data and Results for Mixtec "*čii*" (see text)

indicates the area in front of a vertically extended object (where human bellies are), and the region below a horizontally extended object (where animal bellies are, most of the time). Thus, this term is object-relative in the case of a vertically extended LM, and not object-relative in the horizontally extended case. I.e. the relevant reference orientations are supplied by the LM in the vertically extended case, and by the (learned) upright coordinate system otherwise.

Training and Results

The TR is restricted to be a single point for the time being; current work is directed toward the more general case of an arbitrarily-shaped TR.

Figure 3 presents training data and results for Mixtec *čii* ("belly").

Figure 3 (a) contains some of the training data for Mixtec *čii* ("belly"). It shows an oriented upright LM (the large "+"'s on the LM mark the positive directions of the major and minor axes), and a number of point TRs: the circles indicate positive examples of *čii* with respect to this LM, and the small x-marks indicate negatives. Note that the positive examples are all located in the positive direction of the minor axis, or "in front of" the LM.

Figure 3 (b) also holds training data for *čii*. It shows an unoriented horizontally extended LM (no end of either axis is marked as positive), again with a number of point TRs indicating positive and negative examples. Recall that in the case of horizontally extended LMs, *čii* (or "belly") is associated with that region of space which is found where animal bellies usually are, viz the area below the LM. Thus, for horizontally extended LMs, it does not matter whether the axes of the object are marked for direction or not.

There is training data relative to other LMs as well (upright LMs pointing to the right, horizontal LMs with directed axes, such that the minor axis points downward, etc.), but considerations of space preclude presentation of all of these.

Figure 3 (c) illustrates the results of the learning. The size of the black circles indicates the appropriateness, as judged by the trained system, of using the term *čii* to describe the relation between a point TR at that location, and the LM shown. Note that the LM points to the right here, and that it is the region to the right that is considered to be *čii* the LM. In the case of an upright LM pointing to the left (like the LM presented with the training data in (a)), points to the left of the LM would be considered *čii* the LM.

Figure 3 (d) illustrates the results of the learning relative to an unoriented horizontally extended landmark. As desired, points under the LM are considered to be *čii* the LM. Again, space considerations rule out presentation of the results for all possible LMs, but the concept is successfully learned.

The system has learned the English concepts *in front of, in back of, above, below, left, right, in, out, off,* and *on*. It has also learned the following concepts from Chalcatongo Mixtec: *ini* ("spleen, gut", meaning inside), *šini* ("human head", above an upright LM), *čii* ("belly", already discussed), *ha?a* ("foot", at the base of an upright LM), *siki* ("animal back", above a horizontally extended LM), and *yata* ("human back", in front of an upright LM). Clearly, several of these are object-relative terms.

System Design

One of the central mechanisms used in the system is one that determines to what extent a given orientation matches a reference orientation. A node mechanism is presented here which allows this sort of orientation comparison, and allows the node to learn to tune itself to embody an appropriate reference orientation and tolerance.

Once this node mechanism has been presented, the network as a whole will be presented and discussed.

Θ-Nodes

All orientations in the system are represented in (sin,cos) pairs.[1] The structure of a single Θ-node, which learns to tune itself to a preferred orientation and tolerance using this representation, is presented in Figure 4. Note that sin_θ, cos_θ, and σ_θ are variables internal to

[1]This representation can be viewed as a minimalist version of coarse-coding.

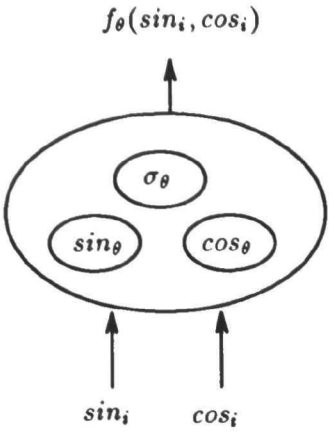

Figure 4: Internal Structure of a Single Θ-node

the Θ-node, defining its preferred orientation and tolerance. These internal values may be learned together with weights in the network as a whole, or may be imposed from the outside to order to "tune" the node.

The function of a Θ-node is simply a Gaussian (see (Moody & Darken 1988) for earlier work using Gaussian nodes in a somewhat different way in connectionist networks):

$$f_\theta(sin_i, cos_i) = \exp\left[-\frac{(sin_\theta - sin_i)^2 + (cos_\theta - cos_i)^2}{\sigma_\theta^2}\right] \quad (1)$$

The partial derivatives of f_θ with respect to each of the internal variables are:

$$\frac{\partial f_\theta}{\partial sin_\theta} = \frac{-2(sin_\theta - sin_i)}{\exp\left[\frac{((sin_\theta - sin_i)^2 + (cos_\theta - cos_i)^2)}{\sigma_\theta^2}\right]\sigma_\theta^2} \quad (2)$$

$$\frac{\partial f_\theta}{\partial cos_\theta} = \frac{-2(cos_\theta - cos_i)}{\exp\left[\frac{((sin_\theta - sin_i)^2 + (cos_\theta - cos_i)^2)}{\sigma_\theta^2}\right]\sigma_\theta^2} \quad (3)$$

$$\frac{\partial f_\theta}{\partial \sigma_\theta} = \frac{2((sin_\theta - sin_i)^2 + (cos_\theta - cos_i)^2)}{\exp\left[\frac{((sin_\theta - sin_i)^2 + (cos_\theta - cos_i)^2)}{\sigma_\theta^2}\right]\sigma_\theta^3} \quad (4)$$

From these, we can easily find $\frac{\partial E}{\partial sin_\theta}$, $\frac{\partial E}{\partial cos_\theta}$, and $\frac{\partial E}{\partial \sigma_\theta}$, which enables us to use back-propagation to train the internal variables of a given Θ-node, together with the weights of the network in which they are embedded.

Every Θ-node will learn its σ_θ, and several, though not all, will learn their orientations as well (sin_θ, cos_θ).

Network Architecture

Figure 5 illustrates the architecture of the network used here. Note that all weights below the dotted line are frozen at 1.0, so learning occurs only above this line.

Recall that the system uses both orientation-based and bitmap-based primitives. We examine the halves of the network handling each of these in turn.

We first consider orientation-based processing. This is done by the left-hand half of the network, which receives input from the TR orientation and the major and minor axis orientations of the LM (here labeled "MAO" and "mAO", respectively). If the LM major axis has no inherent direction, then the two possibilities for that orientation are loaded into the MAO(1) and MAO(2) inputs (e.g. if it is a tall object with neither the top nor the bottom marked as the positive direction, the inputs are loaded with the (sin,cos) representations of 90 degrees and 270 degrees). If, on the other hand, the LM has an inherent direction for its major axis, *both* MAO(1) and MAO(2) are loaded with the representation for that direction. The minor axis orientation inputs (mAO(1) and mAO(2)) are handled analogously.

All hidden nodes marked Θ are Θ-nodes of the sort described above, and all receive their (sin_i, cos_i) inputs (recall Figure 4) from the TR orientation input. Of these nodes, the middle three learn their sin_θ, cos_θ, and σ_θ, while the others learn only σ_θ, and have their (sin_θ, cos_θ) internal variables set by the lines leading to them from the MAO and mAO inputs. Thus, the middle three nodes learn to tune themselves (generally to the upright vertical and to left and right), while the two on the left are always tuned to whatever the major axis orientation is, and the two on the right to whatever the minor axis orientation is, with ambiguously oriented LMs handled as described above.

Consider node H, above the Θ-node layer. It, like the other nodes not explicitly marked with a Θ, computes the usual sigmoid of its weighted and summed input. Notice that the two Θ-nodes associated with the major axis project to H, on links which are *constrained to be of the same weight*, denoted "r" (see (LeCun 1989) for details on this technique of weight-sharing). The two links from the MAO Θ-nodes to node J will also be constrained to be the same, though not necessarily the same as "r". Weights from the mAO Θ-nodes are similarly constrained.

Notice that under these constraints, node H will treat both MAO Θ-nodes identically, so that it can learn to respond to ambiguously directed LMs without worrying about which of the two possible orientations is represented by which of the two Θ-nodes. Note also that if the LM *does* have inherent orientation, then H can receive greater input from these nodes than it would in the case of an ambiguously oriented LM, since in the ambiguous case, at most one of the two Gaussian Θ-nodes will be responding strongly, while in an unambiguously directed case, both may be.

This played a role in the learning of Mixtec *čii*: if there was an inherent minor axis orientation, that direction was the correct one for *čii* as learned by the system. If there was not any inherent directionality to the minor axis however, the system responded to the coincidence with downward vertical of one of the two possibilities for the minor axis. Since a node such as H will receive greater input in the case of an unambigously oriented LM with a TR in that direction, but

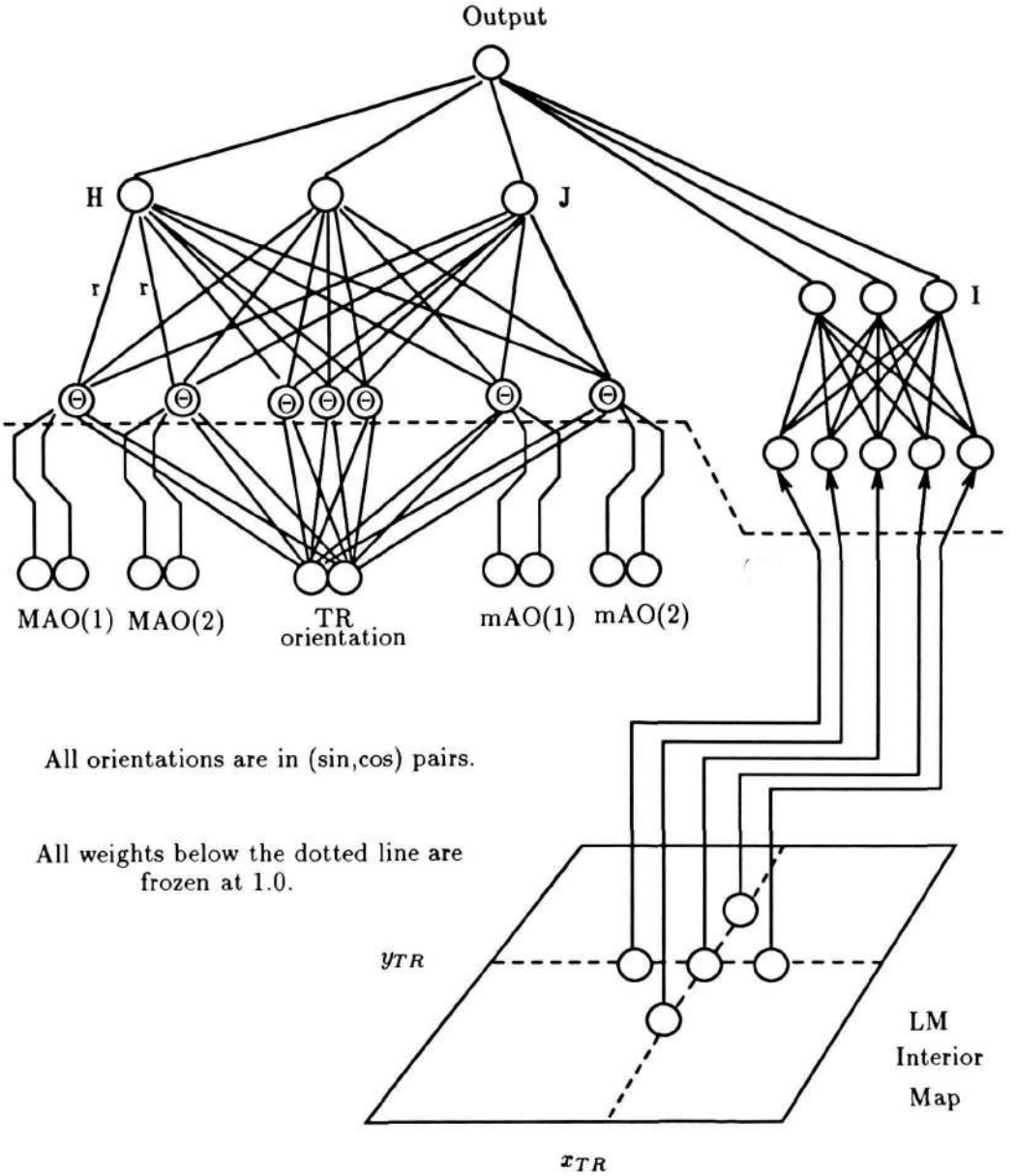

Figure 5: Network Architecture

will still respond somewhat in the case of an ambiguous LM with a TR in one of the two possible directions, this concept was learnable.

We now consider the processing of the bitmap-based LM representation, performed by the right-hand part of the network, which receives input from the LM interior map. Each of the three nodes in the cluster labeled "I" (for interior) has a receptive field of five pixels.

When a TR location is specified, the values of the five neighboring locations shown in the LM interior map, centered on the current TR location, are copied up to the five input nodes. The weights on the links between these five nodes and the three nodes labeled "I" in the layer above define the receptive fields learned. When the TR position changes, five new LM interior map pixels will be "viewed" by the receptive fields formed. This allows the system to detect the LM interior (or a border between interior and exterior) at a given point and to bring that to bear if that is a relevant semantic feature for the set of spatial terms being learned.

Note that the four outer links in this small receptive field are tied to the same value, so that this receptive field is radially symmetric. Thus, this half of the network handles strictly local features such as contact and inclusion, while the rest of the network handles directional features.

The orientation-based and bitmap-based representations are combined at the output level of the network, so that the learned semantics for a given spatial term may involve a mixture of evidence from the two representation types.

Conclusions

A connectionist system has been presented which learns perceptually grounded semantics for both object-relative and non-object-relative spatial terms, from English and Mixtec. The system relies on the use of a particular visual primitive, the orientation of the imaginary ray connecting the two relevant objects where they are nearest to each other. This orientation is compared to various reference orientations, by Gaussian Θ-nodes tuned for a particular orientation and tolerance. These Θ-nodes learn their σs, and some learn their reference orientations as well.

Immediate future work is directed toward extending the system to handle trajectors which are not simply a single point. In addition, moving trajectors, and a means for handling the resulting polysemy, are on the agenda.

References

Bowerman, M. 1989. Learning a Semantic System: What Role do Cognitive Predispositions Play? In et al, M. L. R., ed., *The Teachability of Language*, 133–169. Paul H. Brookes, Baltimore.

Brugman, C. 1983. The Use of Body-Part Terms as Locatives in Chalcatongo Mixtec. in Report No. 4 of the Survey of California and other Indian Languages, pp. 235-90. University of California, Berkeley.

Feldman, J., Lakoff, G., Stolcke, A., and Weber, S. 1990. Miniature Language Acquisition: A Touchstone for Cognitive Science. Technical Report TR-90-009, International Computer Science Institute, Berkeley, CA. also in the Proceedings of the 12th Annual Conference of the Cognitive Science Society, pp. 686–693.

LeCun, Y. 1989. Generalization and Network Design Strategies. Technical Report CRG-TR-89-4, Department of Computer Science, University of Toronto.

Moody, J., and Darken, C. 1988. Learning with Localized Receptive Fields. In *Proceedings of the 1988 Connectionist Models Summer School.* Morgan Kaufmann.

Regier, T. 1990. Learning Spatial Terms Without Explicit Negative Evidence. Technical Report 57, International Computer Science Institute, Berkeley, California.

Regier, T. 1991. Learning Perceptually-Grounded Semantics in the L_0 Project. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (to appear).*

Talmy, L. 1983. How Language Structures Space. Technical Report 4, Institute of Cognitive Studies, University of California at Berkeley.

Weber, S. H., and Stolcke, A. 1990. L_0: A Testbed for Miniature Language Acquisition. Technical Report TR-90-010, International Computer Science Institute, Berkeley, CA.