Learning Spatial Concepts Using a Partially-Structured Connectionist Architecture

Terry Regier*

TR-91-050

October 1991

Abstract

This paper reports on the learning of spatial concepts in the L_0 project. The challenge of designing an architecture capable of learning spatial concepts from any of the world's languages is first highlighted by reviewing the spatial systems of a number of languages which differ strikingly from English in this regard. A partially structured connectionist architecture is presented which has successfully learned concepts from the languages outlined. In this architecture, highly structured subnetworks, specialized for the spatial concept learning task, feed into an unstructured, fully-connected upper subnetwork. The system's success at the learning task is attributed on the one hand to the constrained search space which results from structuring, and on the other hand to the flexibility afforded by the unstructured upper subnetwork.

*The author may be reached by e-mail as **regier@icsi.Berkeley.EDU**

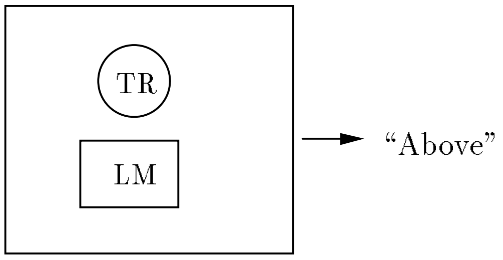


Figure 1: Learning to Associate Scenes with Spatial Terms

1 Introduction

The L_0 project [Feldman *et al.*, 1990; Weber and 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. The training data is a set of pairings of scenes with sentences in the target language, such that the sentence is true of the 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, making this subtask a challenging one. Examples of this crosslinguistic variation will be discussed below.

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, 1991b; Regier, 1991a] used connectionist mechanisms to learn several basic spatial terms in English and some other languages, and handled the problem of learning the semantics for these in the absence of explicit negative instances. This paper provides an overview of the current system, including several features not covered in the earlier papers.

2 Crosslinguistic Variation in Spatial Systems

As pointed out in [Brugman, 1983; Bowerman, 1989; Talmy, 1983], among others, languages differ dramatically in the ways in which they express spatial relations. Several examples of spatial systems exhibiting features which are significantly different from English are described briefly below, to serve as an indication of the breadth of coverage the system must exhibit.

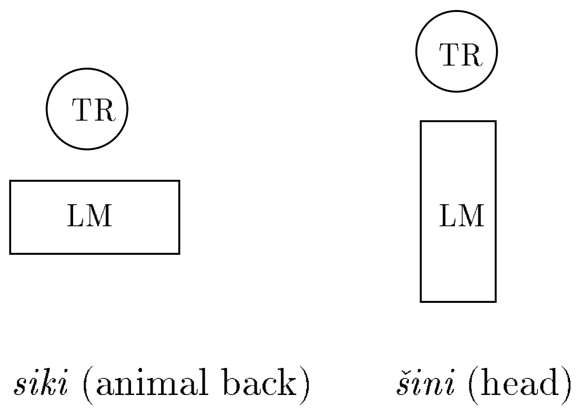


Figure 2: Examples of Mixtec “siki” and “šini”

2.1 Mixtec

[Brugman, 1983] provides an overview of the Chalcatongo Mixtec spatial system, which is based in large measure on a human and animal body-part metaphor. Thus, a trajector above a long, wide landmark is considered to be located at the landmark’s “animal-back”, by analogy to the dorsum of a horizontally-extended quadruped. By contrast, a trajector above a tall, erect landmark is considered to be located at the landmark’s “head”, even if the landmark has no actual head. This distinction is illustrated in Figure 2. Note that both scenes would be classified as “above”, or “over”, in English. In this example, Mixtec is picking up on a visual feature which English is not, namely the orientation of the major axis of the landmark.[1]

2.2 German

Interestingly, even closely related languages can differ substantively in their spatial systems, as pointed out in [Bowerman, 1989]. Figure 3 shows two scenes which would both be classified as “on” in English, but which do not fall in the same category in German. In German, the orientation of the landmark surface which supports the trajector is significant, while it is not for English “on”.

2.3 Bengali

Figure 4 illustrates the fact that Bengali also makes a visually-based distinction which English does not, namely, the distinction between partial and complete inclusion. The term

[1]Note that English is not always insensitive to this feature; it plays a role in terms such as “along”.

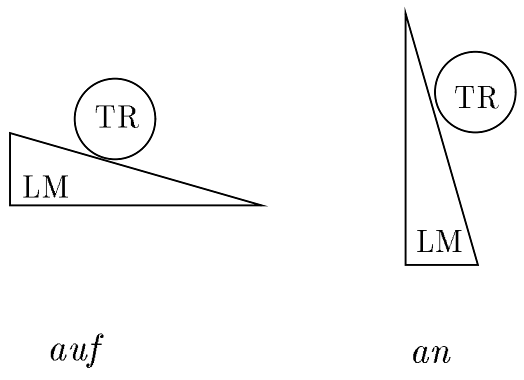


Figure 3: Examples of German "auf" and "an"

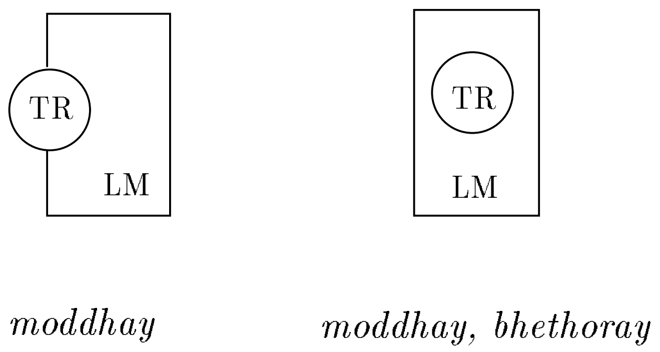


Figure 4: Examples of Bengali "moddhay" and "bhethoray"

"moddhay" is applicable in any situation in which the trajector is at least somewhat inside the landmark, while the term "bhethoray" is reserved for those situations in which the inclusion is complete [Ahmad, 1990].

2.4 Some Others

An exhaustive cataloging of cross-linguistic variations in spatial systems is well beyond the scope of this paper, and indeed probably beyond the scope of any single paper. Nonetheless, there are a number of additional phenomena from other languages which deserve mention. Some of these are listed below.

- Dutch: the preposition "aan" (cognate with but not semantically identical to German "an") seems to involve a notion of hanging, such that earrings are worn "aan" the ear, but a band-aid on one's leg is not "aan" the leg, unless it is mostly loose, and hanging by a corner.
- Korean: the verbs "kki-ta" and "ppay-ta" can be used only when there is a *tight fit* between the landmark and trajector (e.g. a ring on a finger, a finger in a ring, a lid on a jar, a cassette in its case) [Bowerman, 1989].
- Palestinian colloquial Arabic: the term "fawq", usually glossed as "up" or "above", can also be used to refer to situations in which there is contact between trajector and landmark *provided the trajector is located high with respect to the deictic center.* Thus, one could refer to a book on top of a tall refrigerator as being "fawq" the refrigerator, but a book lying on the kitchen table cannot be said to be "fawq" the table (that would imply that it was hovering above the table in mid-air) [Muwafi, 1991].

3 General Approach

A partially structured connectionist network has been designed, and trained under a variant of back-propagation [Rumelhart and McClelland, 1986], to learn spatial concepts from several of the above languages. Figure 5 illustrates the architecture of the network. While there is a good deal of detail shown here, we can begin by pointing out two basic facts about the network:

- **Input and output**: The input scene, with trajector and landmark labeled, is shown at the bottom of the figure. The outline of the trajector is copied into the TR Boundary Map, and the outline of the landmark is copied into the LM Boundary Map. These two boundary maps are kept in register such that if they were to be superimposed, the result would be identical to the input scene.

  If the network has been trained to learn some spatial term, the output node at the top will yield a value indicating how appropriate that term would be when describing the relation shown in the input scene.
- **Partial structuring**: The network consists of three modules, marked "A", "B", and "C". Of these, "A" and "B", the lower two, are highly structured, and were designed

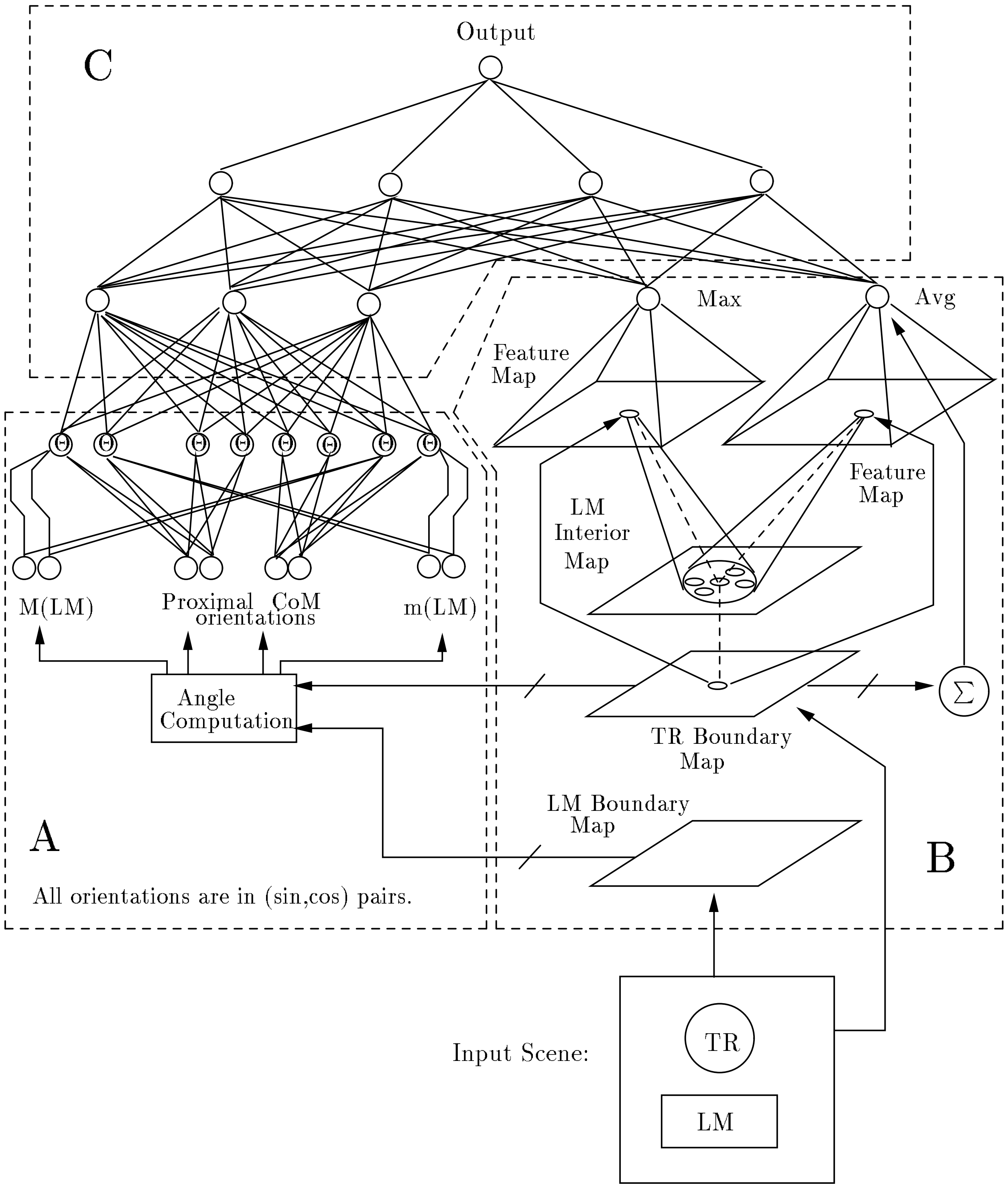


Figure 5: Network Architecture

with the spatial concept learning task very much in mind. Module "C", on the other hand, is an unstructured module, exhibiting full connectivity between hidden layers, as is common in connectionist networks.

This partially structured design is an attempt to capture some of the best features of both structured and unstructured network design, namely:

- the *tractability* in learning that results from structuring, as the dimensionality of the search space is typically dramatically reduced, and
- the *flexibility* that results from an unstructured, fully-connected network design. Flexibility is clearly a critical feature of a system which must be able to adapt itself to the spatial system of any natural language.

4 Directional Features and Θ-Nodes

We consider first the structured module labeled "A". This module is responsible for handling *directional features* in the scene.

4.1 Directional Features

A number of orientations are extracted from the scene, or, in some cases, learned. These orientations are of two types:

- Relational orientations: Orientations which describe the location of one object relative to another. The two which are currently in use are
  - Proximal orientation: The orientation of the line connecting the two objects *where they are closest* to one another.
  - Center-of-mass orientation (CoM orientation): The orientation of the line connecting the centers of mass of the two objects.

  These two relational orientations are illustrated in Figure 6.
- Reference orientations: These are orientations with which relational orientations may align. Examples are the major and minor axes of the landmark, and upright vertical.

All of the orientations mentioned above are extracted from the scene, with the exception of upright vertical, which is learned.

4.2 Θ-Nodes

The central computation performed in this first structured module is *orientation comparison*:

> How well does a given relational orientation align with a given reference orientation?

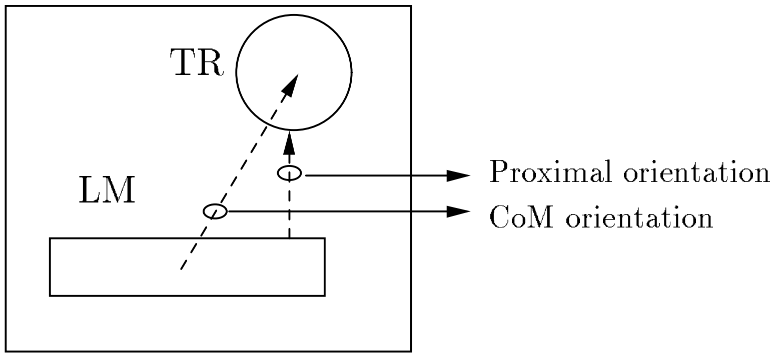


Figure 6: Relational Orientations

For example, we might want to know how well the CoM and proximal orientations align with upright vertical. Clearly, in Figure 6, the proximal orientation is perfectly aligned with upright vertical, while the CoM orientation is not.

All orientations in the system are represented in (sin,cos) pairs. Given this, we can easily measure the degree of alignment of two orientations by using a Gaussian (see [Moody and Darken, 1988] for earlier work using Gaussian nodes in a somewhat different way in connectionist networks):

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

where (sin_i, cos_i) encodes the relational orientation, and (sin_θ, cos_θ) encodes the reference orientation to which it is being compared.

The node which performs this comparison is termed a Θ-node, illustrated in Figure 7. Note that the relational orientation is supplied as input to the node, while the reference orientation is encoded in variables internal to the node. The σ_θ for the Gaussian is also kept as a variable internal to the node.

These internal variables may be trained together with the weights of the network in which the Θ-node is embedded, or may themselves be input to the node, so as to tune it to a particular reference orientation on the fly. In the system being presented, Θ-nodes are used in each of these two ways, as we shall see.

In order to train the internal variables of a Θ-node, we need to determine the partial derivative of the error with respect to each of these variables, i.e. $\frac{\partial E}{\partial sin_\theta}$, $\frac{\partial E}{\partial cos_\theta}$, and $\frac{\partial E}{\partial \sigma_\theta}$. These are easily obtained once we find the partial derivative of f_θ (recall Equation 1) with respect to each of the internal variables:

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

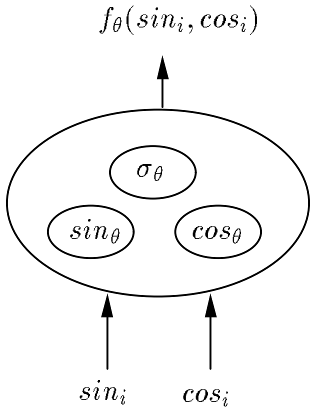


Figure 7: Internal Structure of a Single Θ-node

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

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

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

Recall Figure 5. As can be seen, module "A" contains a layer of eight Θ-nodes. Of these, the leftmost four accept the proximal orientation as relational orientation input, while the rightmost four take the CoM orientation. In addition, the major and minor axis orientations of the landmark (marked **M(LM)** and **m(LM)**, respectively) provide input to some of the nodes. Each of these nodes will have its reference orientation set to the orientation of the corresponding landmark axis. In contrast, the middle four nodes do not accept reference orientation input. These nodes may train their internal variables to arrive at reference orientations which are useful for the task.

5 Non-Directional Features and Feature Maps

5.1 Feature Maps

Returning momentarily once again to Figure 5, we now consider module "B", which handles non-directional features such as inclusion and contact, among others.

To begin with, notice that in this module, above the two maps that receive input from the scene (namely the TR and LM boundary maps), there is a map labeled the *LM interior*

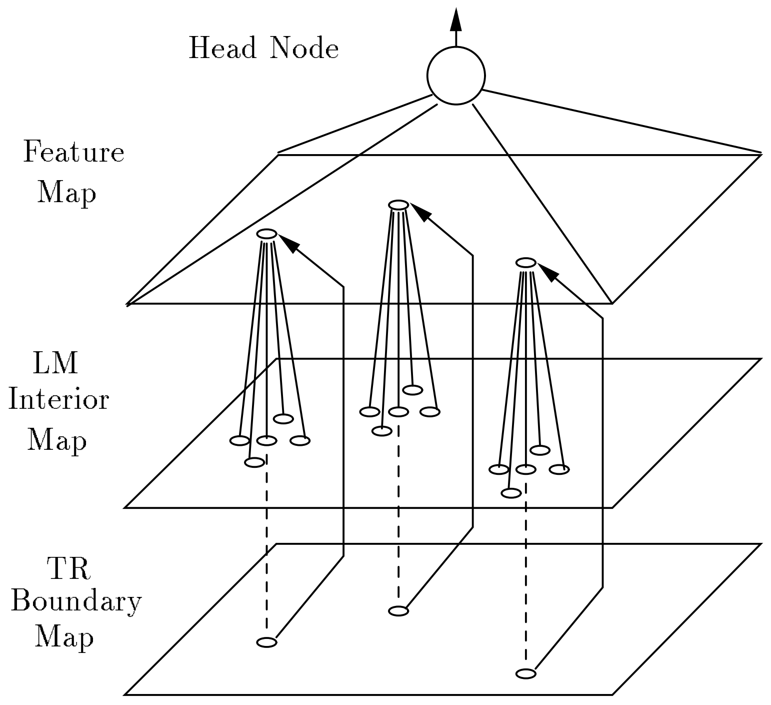


Figure 8: Feature Map Architecture

map. This map contains a copy of the landmark with all of the interior of the landmark activated, as well as the boundary.[2] The LM interior map will play a critical role in this module.

Above the LM interior map, and receiving input from it, are two *feature maps*. These are the central structural devices of this module.

The design of a single feature map is shown in Figure 8. It is simply a map of units, with a head node at the top taking some function of the entire map; this function is either the maximum or the average value of the nodes in the feature map.

There are three forms of structuring built into the feature map:

- Each node has a very highly localized receptive field, examining only the node directly below in the LM interior map, and its four nearest neighbors. This yields a very simple center-surround receptive field.
- The weights of corresponding links at different positions are constrained to be identical (see [LeCun, 1989] for details on this technique of weight-sharing). In addition, all links in the surround are constrained to be of the same weight. This implies that there are actually only two *weights* to be adjusted for a single feature map, despite all

[2]This is accomplished using a simple spreading activation mechanism.

the links: that for all center links, and that for all surround links.

- Each node in the feature map is gated by the node in the corresponding position in the TR boundary map, such that if the node in the TR boundary map is not activated, the feature map node will not respond.

Thus, the function of a given feature map node at position i is

$$F_i = [\sigma(\sum_{j \in N(i)} l_j w_{ij}) \times t_i], \tag{5}$$

where σ is the usual sigmoidal squashing function, and $N(i)$ is that set of visual field positions which constitutes the neighborhood of a unit at position i, namely, position i together with its four nearest neighbors, as mentioned above. l_i and t_i are, respectively, the LM interior map unit at position i, and the TR boundary map unit at the same position.

Thus, the function of a feature map unit at position i is the usual sigmoid of the weighted sum of its inputs (from i's neighborhood in the LM interior map), but gated by the TR boundary map unit at position i. The effect of this is that whatever function the receptive field is trained to compute, given input from the LM interior map, that function is computed at every point in the visual field which is a part of the trajector boundary, and only at those points.

Finally, the "head node" shown in Figure 8 computes some function over all nodes in the feature map below it. There are currently two types of head nodes:

- **Max:** This node returns the maximum response over the nodes in the map below.

$$F = \max_i F_i$$

- **Avg:** This node returns the average response of the nodes in the map below it, averaged over only those positions corresponding to TR boundary points.

$$F = [\sum_i F_i]/size(\text{TR boundary})$$

  By definition, F_i is zero when there is no TR boundary point at i (recall Equation 5).

The overall architecture shown in Figure 5 contains two feature maps, one headed by a node returning the maximum over the map, the other headed by an averaging node.

The intuition behind having a node which takes the maximum of the responses of the nodes in the feature map below is that there are cases in which semantically significant perceptual features (such as contact) may occur on just a small portion of the TR boundary. Examples of this are English "on" and German "auf". The system should be constructed so as to be able to respond to such localized events.

At the same time, there are also cases in which the response along the entire TR boundary should be taken into account. Examples of this are the distinction in Bengali between "moddhay" and "bhethoray" (partial vs. total inclusion), and the Korean sensitivity to tightness-of-fit.

The idea here is that averaging nodes will allow the system to learn "bhethoray", for the fraction of TR boundary points that lie within the LM gives a good indication of the extent to which the input scene should be classified as "bhethoray".[3] A node which simply takes the maximum value of all the nodes in the feature map below would not be able to do this.

In contrast, just a single TR boundary point lying inside the LM would be enough to cause the scene to be classified as "moddhay"; this could be done using a head node taking the maximum over the map below it.

5.2 Learning in Feature Maps

As mentioned above, the system is trained under a variant of back-propagation. In order to train the weights within the feature maps, several changes to the usual back-propagation setup were necessary. These changes are described in this section.

Learning proceeds differently for feature maps headed by *Max* and *Avg* head nodes.

- If m is the *Max* head node of a feature map, the feature map nodes compete to determine which will impose the weight changes for its incoming weight vector on the weight vectors of the other nodes. The map node with the maximum response will update its incoming weights, and the weight vectors for all other nodes in the map are constrained to take the same values as those of the winner. This is done as follows:

  The value $\delta_m = -\frac{\partial E}{\partial net_m}$ is computed for node m as if it were an ordinary sigmoid unit. This value is then propagated down to each feature map node f such that $\delta_f = \delta_m$ for all local feature units f in the feature map. Given this, we can compute the weight updates for the links entering the feature map node which had the greatest response, update its weights, and constrain the incoming weight vectors for all other feature map nodes to be identical to that.

- Things are somewhat more complicated in the case of *Avg* nodes. Figure 9 illustrates a schematized version of the structure of the map below an *Avg* node, here called a. Only two feature map nodes are shown (numbers 3 and 8), and they are shown with only two input lines each (c for center and p for periphery). In reality, there is a full map of such nodes, and each has four peripheral inputs and one central one. This version is used in order to avoid an excessively cluttered exposition.

  Here, n is the number of TR boundary points, F_i is the output of feature map unit i, and o_p^i and o_c^i are the nodes in the LM interior map which connect to node i via the peripheral and central links, respectively.

  Recall from the derivation of back-propagation that

$$-\frac{\partial E}{\partial w_{ji}} = -\frac{\partial E}{\partial net_j}\frac{\partial net_j}{\partial w_{ji}} = \delta_j o_i \tag{6}$$

[3]This of course assumes that the individual nodes in the feature map are able to learn to detect inclusion at a given point.

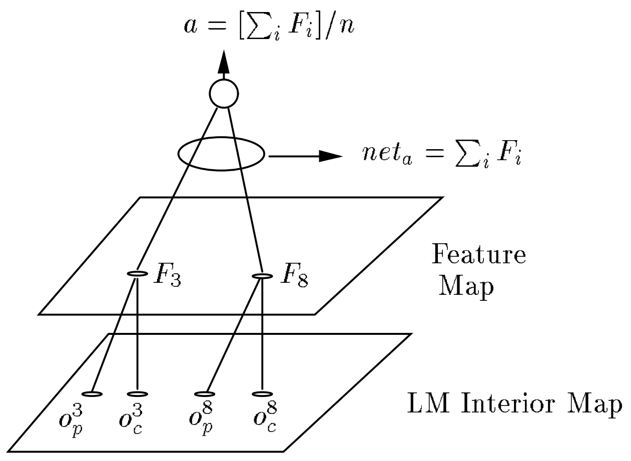


Figure 9: Learning Using *Avg* Nodes (see text)

---

where w_{ji} projects from i to j, and that

$$\delta_j = -\frac{\partial E}{\partial net_j} = -\frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial net_j} = -\frac{\partial E}{\partial o_j} f'(net_j). \tag{7}$$

Now, for an averaging node a,

$$\frac{\partial o_a}{\partial net_a} = \frac{1}{n}, \tag{8}$$

so that

$$\delta_a = -\frac{\partial E}{\partial o_a}\frac{1}{n}. \tag{9}$$

Since the function of a is to take the average of all the nodes in the map below it, the weights connecting the map nodes to a are all frozen at 1.0. Thus, for a feature map node f, since $-\frac{\partial E}{\partial o_j} = \sum_k [\delta_k w_{kj}]$, and since $w_{af} = 1.0$,

$$\delta_f = -\frac{\partial E}{\partial o_f}\frac{\partial o_f}{\partial net_f} = \delta_a \frac{\partial o_f}{\partial net_f} = \delta_a f'(net_f). \tag{10}$$

This means that, if w_i^f is the weight on the ith input link to feature node f,

$$-\frac{\partial E}{\partial w_i^f} = \delta_a f'(net_f) o_i^f = -\frac{\partial E}{\partial o_a} \times \frac{1}{n} \times f'(net_f) \times o_i^f, \tag{11}$$

where o_i^f is the output of the node feeding into input line i of the feature map node at position f (see Figure 9).

Gradients are computed at all feature map nodes. Finally, all weights w_i^f are updated according to the sum

$$\sum_k -\frac{\partial E}{\partial w_i^k}. \tag{12}$$

Using these modifications to the usual back propagation algorithm, links within these feature maps can be trained along with those in the rest of the network. The result is that the head nodes of the feature maps learn to detect non-directional features such as contact, inclusion, tightness of fit, and the like, when needed. The architecture being presented here has two feature maps, one headed by a **Max** node, and one by an **Avg** node (recall Figure 5).

6 The Architecture in Review

The two structured modules in the design, modules "A" and "B", have been discussed in detail. The design of module "C" does not merit much discussion in and of itself, as it consists simply of standard, fully connected hidden layers of sigmoidal units, as are typical of connectionist networks.

However, it is worthwhile mentioning the role that module "C" plays during the learning of spatial concepts, relative to the other two modules,

The two structured modules, as mentioned, are responsible for the learning of particular directional and non-directional *features*; the various output nodes of these modules will provide an indication of the extent to which a given feature is present in the scene. Thus, after training, we might have a Θ-node (in module "A") indicating the degree to which the proximal orientation aligns with upright vertical, and a feature map head node (in module "B") indicating the presence or absence of contact.

These learned features can then be combined through module "C", in more or less "classic" connectionist fashion, to yield an output value. This reflects one of the central design decisions embodied in the network: the division of labor between highly structured feature-detecting modules on the one hand, and an unstructured feature-combining module on the other. As mentioned earlier, the motivation behind this is the desire to capture, in a single network, the learning tractability afforded by structuring, and the flexibility in feature combination which results from unstructured hidden layers.

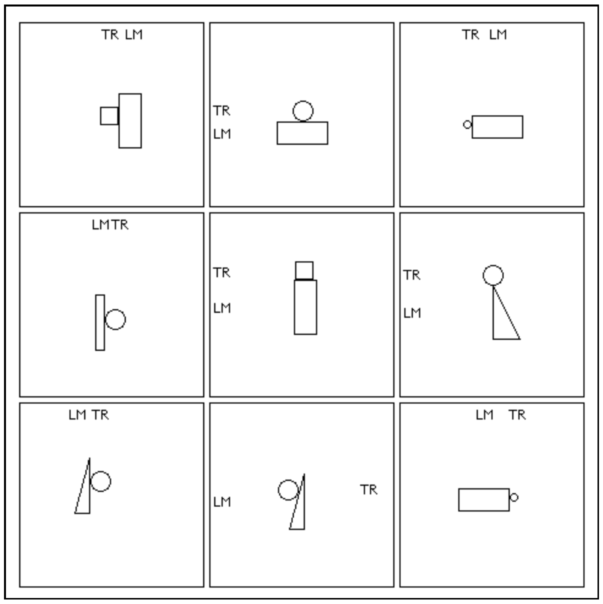
7 Results

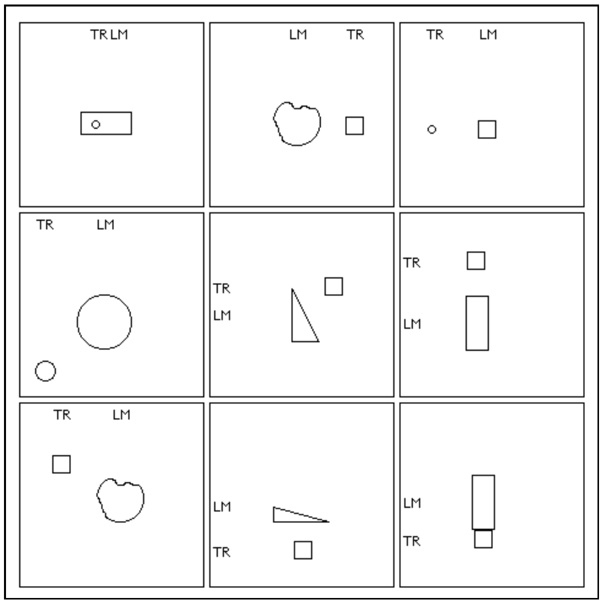
The system was trained under quickprop [Fahlman, 1988], a variant of back-propagation which exhibits fast convergence. Figure 10 presents a typical training set, that for the English preposition "on". Here, nine positive instances of "on" are presented in (a), and nine negative instances in (b).

The system as presented has been trained on the following spatial concepts:

- **English:** *above, in, on, outside*

(a)





(b)

Figure 10: Positive and Negative Examples for English "On"

- **Mixtec:** *šini*
- **German:** *an, auf*
- **Bengali:** *bhethoray, moddhay*
- **Japanese:** *ue ni* (similar to English "above", but insensitive to contact between TR and LM; thus some scenes which would be classified as "on" in English are described using "ue ni" in Japanese.)

Training was in all cases very fast, requiring under 100 quickprop epochs to attain error rates under 0.0001.

Figure 11(a) illustrates the difference between the German term "auf" and English "on", by showing the system's response, once it was trained to recognize "auf", on a series of positive examples of "on". The number between 0.0 and 1.0 at the bottom of each scene indicates how appropriate the system deems the use of "auf" for the scene. As shown here, "auf" requires not only that the TR be in contact with (and presumably supported by) the LM, but also that the proximal orientation be roughly upright vertical.[4]

Similarly, Figure 11(b) illustrates the difference between the Mixtec term "šini" and German "auf", by showing the system's response to positive examples of "auf" after it was trained to recognize "šini". As reflected by the system's response in each case here, Mixtec "šini" requires that the major axis of the LM be oriented vertically, while "auf" has no such requirement.

[4]The notion of support, which is in fact central to English "on", and possibly German "auf" as well, is not currently implemented, but will be in future versions of this system.

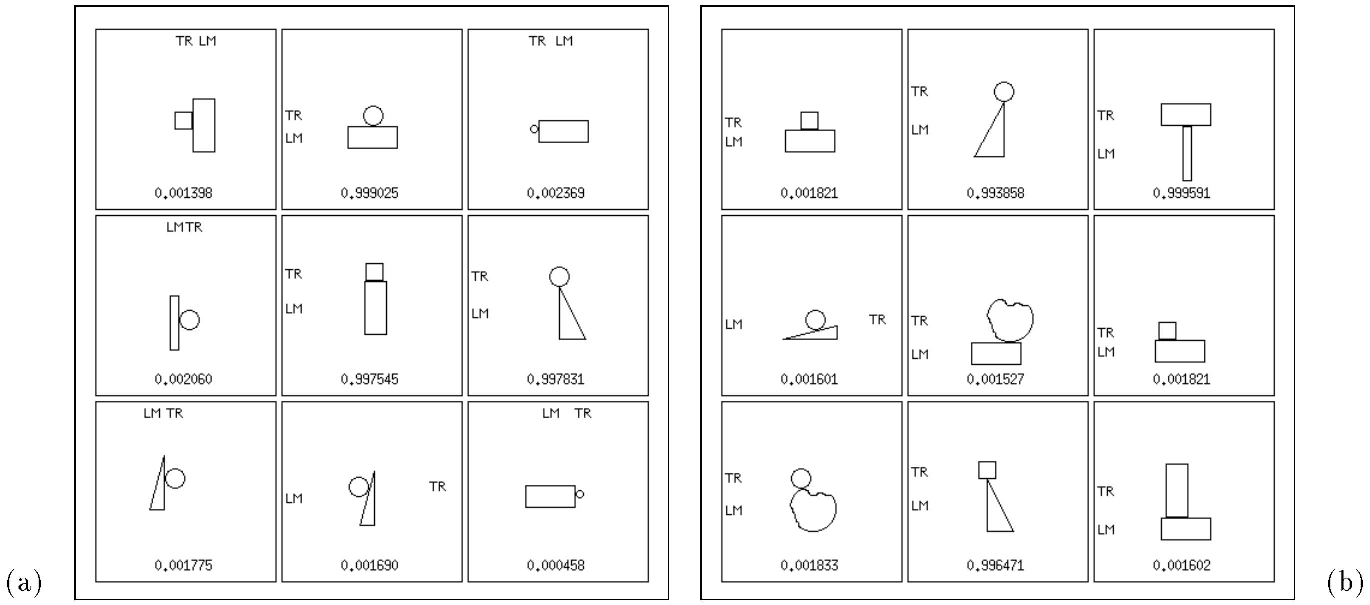


Figure 11: German "Auf" vs. English "On", and Mixtec "Šini" vs. German "Auf"

Finally, Figure 12 illustrates the difference between the Bengali terms "bhethoray" and "moddhay", by showing the system's response to positive examples of "moddhay" after it was trained to recognize "bhethoray". Recall that "bhethoray" encodes complete inclusion, while "moddhay" encodes partial or complete inclusion. This is reflected in the examples shown here.

8 Current and Future Work

Current work is focused on two separate tasks. The first is a constantly ongoing task, namely the testing of the system on new spatial systems, seeing whether or not it is in fact capable of learning them.

The second task is extending the system to handle motion. Under this extension, the input to the system will no longer be a single scene, but rather a sequence of scenes, a movie. Preliminary work has been completed on learning such concepts as "into" and the motion sense of "through", and the approach is being tested on other concepts.

One issue which arises at this point is *polysemy*, the phenomenon of several distinct but related meanings for a single lexeme. For example, the English lexeme "in" may be used to denote either static inclusion:

The ball is *in* the box.

or motion into a situation of inclusion, synonymously with "into":

He walked *in* the room.

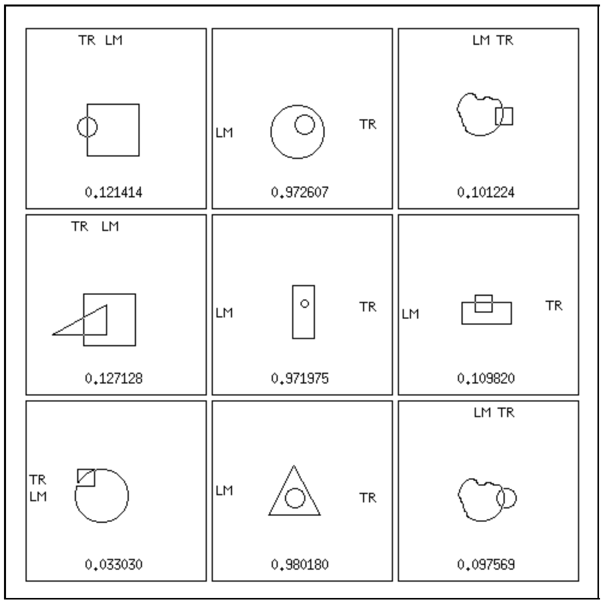


Figure 12: Bengali "Bhethoray" vs. "Moddhay"

The structure of some polysemous locative terms have been worked out in detail (e.g. [Brugman, 1981] on English "over"), and accounting for such data will be a focus of work in the near future.

9 Conclusions

The connectionist system presented in this report was designed to learn spatial concepts from arbitrary natural languages. It has been shown to work on a number of spatial concepts from languages with quite different spatial systems. Ongoing work will be devoted to testing the system on various spatial systems from the world's languages, and extending the system to handle motion and the resulting polysemy.

The system exhibits a partially structured architecture, with highly structured subnetworks, designed with the spatial concept learning task in mind, feeding into an unstructured, fully-connected subnetwork. The unstructured module serves to combine the features learned by the structured subnetworks. This partially structured design is an attempt to capture both the tractability in learning which results from structuring, due to the reduced size of the search space, and the flexibility which results from an unstructured design.

References

[Ahmad, 1990] Subutai Ahmad, (personal communication), 1990.

[Bowerman, 1989] Melissa Bowerman, "Learning a Semantic System: What Role do Cognitive Predispositions Play?," In M. L. Rice et al, editor, *The Teachability of Language*, pages 133–169. Paul H. Brookes, Baltimore, 1989.

[Brugman, 1981] Claudia Brugman, "Story of *Over*," M.A. thesis, University of California, Berkeley. Available from the Indiana University Linguistics Club., 1981.

[Brugman, 1983] Claudia Brugman, "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, 1983.

[Fahlman, 1988] Scott Fahlman, "Faster-Learning Variations on Back Propagation: An Empirical Study," In *Proceedings of the 1988 Connectionist Models Summer School.* Morgan Kaufmann, 1988.

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

[LeCun, 1989] Yann LeCun, "Generalization and Network Design Strategies," Technical Report CRG-TR-89-4, Department of Computer Science, University of Toronto, June 1989.

[Moody and Darken, 1988] John Moody and Christian Darken, "Learning with Localized Receptive Fields," In *Proceedings of the 1988 Connectionist Models Summer School.* Morgan Kaufmann, 1988.

[Muwafi, 1991] Jumana Muwafi, (personal communication), 1991.

[Regier, 1990] Terry Regier, "Learning Spatial Terms Without Explicit Negative Evidence," Technical Report 57, International Computer Science Institute, Berkeley, California, November 1990.

[Regier, 1991a] Terry Regier, "Learning Object-Relative Spatial Concepts in the L_0 Project," In *Proceedings of the 13th Annual Meeting of the Cognitive Science Society*, 1991.

[Regier, 1991b] Terry Regier, "Learning Perceptually-Grounded Semantics in the L_0 Project," In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, 1991.

[Rumelhart and McClelland, 1986] David Rumelhart and James McClelland, *Parallel Distributed Proccessing: Explorations in the microstructure of cognition*, MIT Press, 1986.

[Talmy, 1983] Leonard Talmy, "How Language Structures Space," Technical Report 4, Institute of Cognitive Studies, University of California at Berkeley, January 1983.

[Weber and Stolcke, 1990] Susan Hollbach Weber and Andreas Stolcke, "L_0: A Testbed for Miniature Language Acquisition," Technical Report TR-90-010, International Computer Science Institute, Berkeley, CA, 1990.