The Acquisition of Lexical Semantics for Spatial Terms: A Connectionist Model of Perceptual Categorization

Terry Regier*

TR-92-062

September 1992

Abstract

This thesis describes a connectionist model which learns to perceive spatial events and relations in simple movies of 2-dimensional objects, so as to name the events and relations as a speaker of a particular natural language would. Thus, the model learns perceptually grounded semantics for natural language spatial terms.

Natural languages differ – sometimes dramatically – in the ways in which they structure space. The aim here has been to have the model be able to perform this learning task for terms from any natural language, and to have learning take place in the absence of explicit negative evidence, in order to rule out *ad hoc* solutions and to approximate the conditions under which children learn.

The central focus of this thesis is a connectionist system which has succeeded in learning spatial terms from a number of different languages. The design and construction of this system have resulted in several technical contributions. The first is a very simple but effective means of learning without explicit negative evidence, taking positive instances of other concepts as implicit negative instances of the concept being learned, and deliberately weakening the evidence from these implicit negatives so as to reduce the effect of false implicit negatives in cases where concepts overlap. This method is shown to be effective even in situations involving a high degree of overlap among concepts. In addition, this thesis presents the notion of partially-structured connectionism, a marriage of structured and unstructured network design techniques

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

capturing the flexibility afforded by unstructured networks, and the tractability in learning and improved generalization ability that result from highly structured network design. Finally, the idea of learning within highly specialized structural devices is introduced, with the purpose of restricting the search during learning to a set of options known in advance to be of possible relevance.

Scientifically, the primary result of the work described here is a computational model of the acquisition of visually grounded semantics. This model is shown to successfully learn terms for spatial events and relations from a range of languages with widely differing spatial systems, including English, Mixtec (a Mexican Indian language), German, Bengali, and Russian. The model exhibits prototype effects which roughly match human intuitions, and extensions to the core system model the linguistic phenomena of polysemy and deixis. While no claims are made regarding structure-to-structure correspondences between architectural elements of the model and structures in the brain, the model is informed by a number of neuroscientific results. And perhaps most importantly, the model does more than just recapitulate the data; it also generates a number of falsifiable linguistic predictions regarding the sorts of semantic features, and combinations of features, one might expect to find in lexemes for spatial events and relations in the world's natural languages.

Acknowledgements

I have flipped and flopped a number of times as regards my general outlook on the work reported here. At times I've been frustrated purple by my inability to make any substantive headway through the morass; at other times I was convinced that any idiot could do it. And in the worst of times, these two outlooks co-occurred. More recently, though, I have settled into something approaching a very modest satisfaction. There are rough edges, but all in all I'm reasonably happy with the end result.

For this, I owe thanks to Jerry Feldman, who has been a truly superlative thesis advisor. Through a combination of genuine involvement in the subject matter, judicious advice, a well-developed ability to ferret out inconsistencies, and an equally well-developed sense of fun, Jerry has been instrumental in the development of this thesis. He also knows a lot. I could not have asked for a better advisor. I also owe thanks to George Lakoff, who has in very many ways been a co-advisor — and it would be difficult to imagine a more linguistically well-informed one. It was his work on the use of spatial metaphor in conceptual systems that motivated me to start looking into the linguistic structuring of space in the first place. This thesis is in large measure an outgrowth of my interactions with him, as well as with Jerry. I would also like to thank Robert Wilensky, for introducing me to the computational study of language when I first arrived at Berkeley and was still getting my sea-legs, for agreeing to be on my committee, and for a number of helpful comments on the thesis. He and Donald Glaser each supervised me for a few years, each providing me with advice, a stimulating academic environment, and financial support. Thanks also to Steve Palmer for discussions on related work of his, and for serving on my qualifying exam committee. And thanks to the International Computer Science Institute in Berkeley for funding the last several years of my grad school career.

Adele Goldberg, Andreas Stolcke, Susan Weber, Jordan Zlatev, Tom Dietterich, Jonathan Bachrach, Subutai Ahmad, Dan Jurafsky, Dekai Wu, and Valeriy Nenov have all contributed to this thesis, through helpful conversations. And thanks are especially due to the members of my thesis seminar. Read Chapter 7; you'll understand. Andreas Stolcke, Subutai Ahmad, Susan Weber, David Bailey, Ben Gomes, and Srini Narayanan helped make ICSI an enjoyable and productive environment.

And now, as we leave ICSI and stroll down the street toward the rest of my life, the list of those deserving of my gratitude grows to unwieldy proportions. I hereby thank, with embarrassing effusiveness: Dan Jurafsky, Dekai Wu, Erin Dare, Eric Enderton, and Pearl Chow, for being the rest of Nervous for Nigel, and thereby giving me the chance to live out the dream of every young American male: playing lead guitar for a rock-n-roll band at the 29th annual meeting of the Association for Computational Linguistics. Jimi, eat yer heart out. And most recently, Jordan, Iva, and Kamen Zlatev, Mike Schiff and Nina Amenta, for Blagoevgrad and very large trees. Вот это существительное! Also Alan and Iz, Königspinguin extraordinaire Marshall Bern, Nigel Ward, and Colleen Cotter. Finally, there is of course my Levantine

crowd, my relatives and friends with that vague but distinctly Middle Eastern aroma of coffee and za'tar about them. My brother Chris, my cousin Rima Hanania, and my friends Jumana Muwafi, Imad Ftouni, Madeleine Biskintawi, Basil Ayish, Salim Yaqub, Clay Scott, Lisa Smorto, and Golijeh Golarai have all helped grace my days during my stay here in Berkeley.

And of course, my most heartfelt gratitude and love go to Mom and Dad Regier. If it weren't for them, you wouldn't be reading this.

Contents

1 Introduction **1**

1.1 Overview 1
1.2 Two Approaches to the Thesis 3
1.3 Technical Contributions 4
1.3.1 Learning Without Explicit Negative Evidence 4
1.3.2 Partially-Structured Connectionism 4
1.3.3 Learning Within Structural Devices 5
1.4 Scientific Contributions 7
1.5 Task, Algorithm, and Implementation 8
1.6 The L_0 Project 10
1.7 A Guide to the Remainder of the Thesis 11

2 Linguistic Issues **13**

2.1 The Linguistic Categorization of Space 14
2.2 Cross-Linguistic Variation 14
2.2.1 Mixtec 15
2.2.2 Bengali 16
2.2.3 German 17
2.2.4 Others 17
2.3 A Close Look at English 18
2.3.1 Potential Motion in Static Scenes 18
2.3.2 A Closer Look at *Above* 20
2.3.3 Further Complications 22
2.4 Cognitive Linguistics 24
2.4.1 An Overview 24
2.4.2 Trajectors and Landmarks 25
2.4.3 Prototype Effects in Categorization 25

2.4.4 Deixis 26
2.4.5 Polysemy 28

3 Connectionism **31**
3.1 Overview 31
3.2 Structured Connectionism 34
3.3 Back-Propagation 37
3.4 Learning Sequences Using Back-Propagation 38
3.4.1 Back-Propagation Through Time 38
3.4.2 Time-Delay Neural Networks 41
3.4.3 Back-Propagation with State Units 42
3.4.4 Discussion 43

4 Learning Without Explicit Negative Instances **44**
4.1 The Problem 45
4.2 A Solution: Mutual Exclusivity 47
4.3 Difficulties with Mutual Exclusivity 49
4.4 Salvaging Mutual Exclusivity 51
4.5 Implementation 51
4.6 Results 54
4.6.1 Uniform Attenuation 54
4.6.2 Incorporation of Prior Knowledge 62

5 Static Scenes **66**
5.1 The Problem 66
5.2 Outline of a Solution 69
5.2.1 Directional Features 69
5.2.2 Non-directional Features 70
5.3 Implementing the Solution 74
5.3.1 Preliminaries 76
5.3.2 Directional Features 78
5.3.3 Non-directional Features 81
5.3.4 The Architecture in Review 88
5.3.5 Motivations from Neuroscience 89
5.4 Results 91
5.4.1 Prototype Effects 94

5.4.2 Cross-Linguistic Variation 95

6 Motion **104**

6.1 The Problem 105
6.1.1 Training Data 105
6.1.2 Assumptions 106
6.2 Outline of a Solution 108
6.3 Implementing the Solution 111
6.3.1 Three Potential Methods 112
6.3.2 Source and Path Buffers 117
6.4 Results 127
6.4.1 Back Propagation with State Units 127
6.4.2 Source and Path Buffers 130
6.5 Learning Movies without Explicit Negative Evidence 134
6.6 A Fictitious Spatial Concept: *in/out-of* 145
6.7 Discussion 147

7 Extensions **152**

7.1 Ongoing Work 152
7.1.1 Polysemy 153
7.1.2 Deixis 163
7.1.3 Force Dynamics 168
7.1.4 Reified Paths and Key Events 172
7.2 Possibilities for Future Work 176
7.2.1 Distance 176
7.2.2 Regions as Objects 178
7.2.3 Convex Hulls 180
7.2.4 Implicit Paths 181

8 Perspective and Conclusions **183**

8.1 Related Computational Research 183
8.1.1 Related Non-Learning Systems 184
8.1.2 Related Learning Systems 187
8.2 The Thesis in Retrospect 189

Chapter 1

Introduction

1.1 Overview 1
1.2 Two Approaches to the Thesis 3
1.3 Technical Contributions 4
1.3.1 Learning Without Explicit Negative Evidence . . 4
1.3.2 Partially-Structured Connectionism 4
1.3.3 Learning Within Structural Devices 5
1.4 Scientific Contributions 7
1.5 Task, Algorithm, and Implementation 8
1.6 The L_0 Project 10
1.7 A Guide to the Remainder of the Thesis 11

1.1 Overview

This thesis concerns the connectionist modeling of the acquisition of lexical semantics for spatial terms in a simple visually-grounded domain. This perhaps somewhat abstract-sounding declaration of scope and aims can be easily made more concrete: Imagine a set of movies of simple 2-dimensional objects moving relative to one another, such that each movie has been correctly labeled as a positive instance of some spatial concept from a natural language. For example, Figure 1.1 presents such a movie, a positive example of the Russian preposition *iz-pod*, which has no single-word English counterpart, but translates to "out from underneath". This thesis concerns the design and construction of a system which takes a set of such movies, each labeled as a positive example of some natural language spatial term from some language, and learns the association between words and the events or relations which they describe. Once such a system has successfully accomplished this task, it should be able to indicate which of the natural language terms would be appropriate for describing a movie not previously seen.

Also, very importantly, the system should be able to perform this learning task for any natural language. Since languages differ in the ways in which they structure space, this rules out *ad hoc* approaches. Thus, the task as a whole can be viewed as:

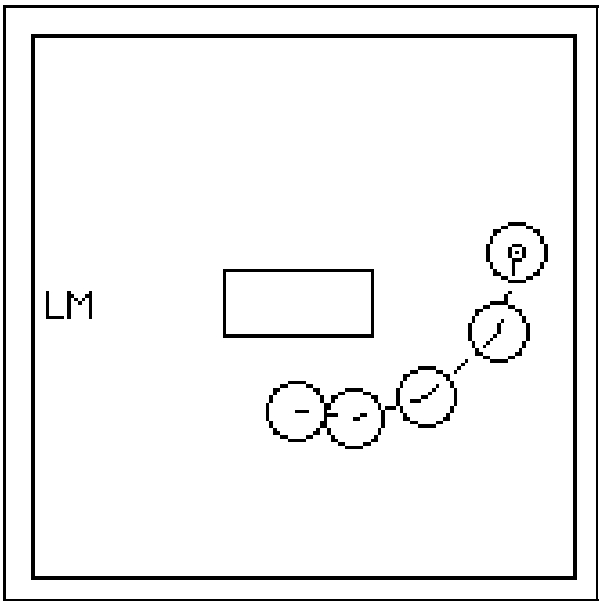


Figure 1.1: A movie: Russian *iz-pod*

Learning how to perceive simple spatial relations, both static and dynamic, so as to name them as a speaker of a particular natural language would.

Figure 1.1 was one of many movies used in training the system described in this thesis. Each movie contains a static object known as the *landmark*, or *LM*; this is the reference object with respect to which other objects are located. In this movie, it is the horizontally extended rectangle in the middle of the scene. Each movie also contains another object, here in motion, known as the *trajector*, or *TR*; this is the object located relative to the landmark. In this movie, the trajector is a small circle moving from the region beneath the landmark, to the right and upwards. The dashed lines connect successive positions of the trajector as it moves. The final frame of the movie is indicated by a tiny circle located inside the trajector. This convention will be used throughout this thesis.

There are two aspects of this task which are worth highlighting at the outset, as they provide much of the challenge. Firstly, as alluded to above and discussed in greater detail in Chapter 2, there is dramatic cross-linguistic variation in the spatial systems of the world's languages. Thus, any system which is to be able to learn the spatial system of an arbitrary natural language must exhibit great flexibility, more perhaps than might be imagined at first.

In addition to this, the system must learn without the benefit of explicit negative instances of spatial concepts.[1] This is reflected in the task description above, which

[1] I.e. it must learn without ever being told, while viewing a movie, that this movie is *not* a good

stipulates that the system is to be exposed only to positive instances of concepts. The motivation for this is that it appears to be the case that children learn language generally, and the semantics for spatial terms specifically, in the absence of explicit negative evidence (see Chapter 4 for further discussion of this point). Thus, any system which purports to be a cognitive model of the acquisition process – and as we shall see, the system described here does in fact have such pretensions – will need to learn under at least roughly the same conditions as those organisms it is seeking to model: children. This constraint of learning without explicit negative evidence raises problems that might not otherwise be an issue, which are covered in detail in Chapter 4. This is thus another source of challenge. It is worth stressing that this is a very general problem, not at all restricted to the spatial concept learning domain. Similarly, the solution proposed in Chapter 4 is a very general one, and could be applied elsewhere.

1.2 Two Approaches to the Thesis

There are two general approaches one might take to the work described here. Under one approach, the learning task is viewed primarily *technically*, as a testing ground for particular connectionist techniques of more or less general applicability. Thus, the work would be evaluated in terms of any technological contributions it may make to the field of connectionist modeling. Under this view, valid contributions might be such things as novel learning techniques, the use of architectural structures which can be seen to be of general value, and the like. In sum, the emphasis here would be on whatever widely applicable tools the work presented adds to the connectionist toolbox.

The other approach views the work not so much as a forum in which to demonstrate the utility of particular computational mechanisms, but rather, *scientifically*, as an explanatory model of a process which exists in the world, and which we are attempting to understand, namely, the acquisition of visually-based semantics. The criteria for evaluating the work under this view are quite different from the criteria highlighted by the technical viewpoint: here, what counts as a result is an explanation of some sort of the process under study. Science, after all, is about the construction of models for the purpose of explaining the world, models which can generate falsifiable predictions about the processes they claim to explain. If such a model is constructed along purely conventional lines, using conventional technology, that is immaterial to the scientist, although to the technologist, it robs the work of its interest. The emphasis under this viewpoint then is on the specific explanations and predictions produced by the model, rather than on the techniques used to arrive at them.

In this dissertation, I adopt both of these viewpoints, as twin frameworks within which to situate the work presented. The two are by no means independent of one

example of some spatial term.

another, however: as we shall see, each drives the other. The modeling effort undertaken here, that is, the specific scientific task at hand, has necessitated a number of technical innovations in connectionist modeling. Similarly, the explanation and predictions that fall out of the model, concerning the acquisition of semantics, i.e. the scientific results of the effort as a whole, are critically dependent on the particular architectural elements and training regimen adopted.

This interplay between the two frameworks will appear throughout this dissertation. It is in fact a central function of the work as a whole to highlight both the utility, for the purposes of linguistic inquiry, of technically detailed computational models, and the fact that such a modeling effort may result in connectionist architectural features and techniques which will find applicability far outside the particular domain for which they were originally conceived.

We proceed now to consider first the technical, and then the scientific, contributions made by the work described here.

1.3 Technical Contributions

The system whose design and construction form the central focus of this thesis is a connectionist network, trained under error back propagation [Rumelhart *et al.*, 1986].[2] The architecture and training regimen associated with this system feature a number of novel techniques, which are potentially of general use across a wide range of applications. These are the following: (a) a method for learning without explicit negative evidence, (b) the notion of *partially-structured* connectionism, and (c) learning within structural devices. Each of these is briefly described in turn below.

1.3.1 Learning Without Explicit Negative Evidence

A method is presented for learning in the absence of explicit negative evidence. The basic idea behind the solution is extremely simple and appears to be quite effective: the system learns a set of concepts simultaneously, and each positive example for one concept is taken as weak implicit negative evidence for all other concepts. This approach is shown to work even in cases in which there is considerable overlap among concepts. This issue of learning without explicit negative evidence is the focus of Chapter 4.

1.3.2 Partially-Structured Connectionism

A balance is struck between unstructured architectures, such as are common in connectionist networks trained under back propagation, and highly structured ones, with

[2] For those not familiar with this computational paradigm, Chapter 3 presents a brief introduction.

the intent of capturing the best of both methods of network design. Specifically, the goal has been to capture the *flexibility* afforded by unstructured networks, together with the *tractability* in learning and improved generalization ability that results from highly structured network design.

Figure 1.2 should serve to give a feel for the overall appearance of such a partially structured connectionist network. This figure presents a simplified version of the system which forms the focus of this thesis, a version which classifies static scenes rather than movies. Here, the input scene is fed in as shown, and a given labeled output node at the top becomes strongly activated if the input scene constitutes a good example of the spatial term with which the node is labeled. Here, three terms appropriate for describing the scene are *above, out,* and *off*, which correspond to the activated output nodes.

For our purposes here, we may ignore much of the detail shown in the figure – this will be explained in later chapters. The critical aspect of the architecture for now is its *partially structured* design. The section of the architecture which is surrounded in dashed outline contains highly structured devices which were designed with the spatial semantics learning task very much in mind, and whose inclusion here embodies the incorporation into the architecture of knowledge regarding the domain. At the same time, the remainder of the network, that is, the hidden layer shown above the dashed box, is an unstructured perceptron hidden layer of the traditional sort, with full interconnection from the layer immediately below. This marriage of two methods of network design constitutes an attempt to capture the best of each: the structural devices provide a means to build knowledge of the domain into the architecture, effecting the straightforward extraction of types of features known to be relevant, while the the unstructured hidden layer allows for flexibility in the combination of the features detected at previous layers.

1.3.3 Learning Within Structural Devices

The structural devices indicated in dashed outline in Figure 1.2 are composed of particular unit types which were designed specifically for the spatial semantics learning task – these units are trained along with the units in the rest of the network in which they are embedded. In contrast, much prior work in structured connectionism has focused either on non-learning systems [Cooper, 1989; Olson, 1989; Weber, 1989a; Shastri and Ajjanagadde, 1990], or on systems in which the structures were composed of essentially the same sort of units as the rest of the network [Hinton, 1986; Fanty, 1988; Harris, 1991]. The idea here has been to build structures out of units whose activation functions are particularly well-suited to the task at hand, but to allow the back propagation of error to reach these units and to adjust a set of parameters associated with them, thereby restricting the search during learning to a range of options known in advance to be of possible relevance. This results in a dramatic reduction of the dimensionality of the search space, facilitating the learning process.

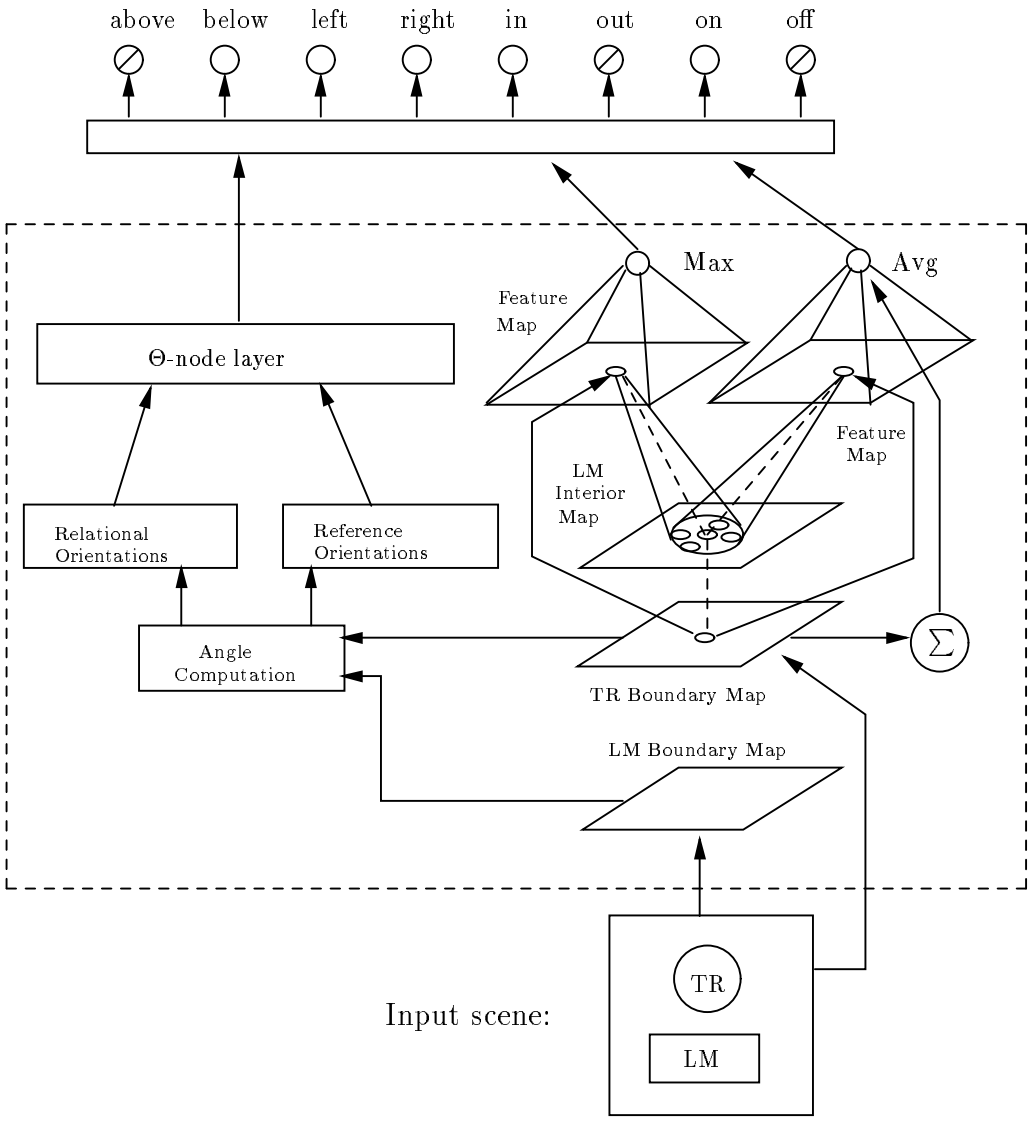


Figure 1.2: Simplified architecture overview (see text)

1.4 Scientific Contributions

The primary scientific contribution of this work is the construction and analysis of a computational model of one particular aspect of the linguistic categorization of space. The particular aspect focused on is what can be viewed as the central conceptual structure of space, as manifested in the spatial systems of particular languages. This is an aspect that has been identified with *closed-class*[3] linguistic forms by Leonard Talmy. He expresses the function of closed-class forms as follows:

> They [closed-class forms] represent only certain categories, such as space, time (hence, also form, location, and motion), perspective point, distribution of attention, force, causation, knowledge state, reality status, and the current speech event, to name some main ones. And, importantly, they are not free to express just anything within these conceptual domains, but are limited to quite particular aspects and combinations of aspects, ones that can be thought to constitute the "structure" of those domains. Thus, the closed-class forms of a language taken together represent a skeletal conceptual microcosm. Moreover, this microcosm may have the fundamental role of acting as an organizing structure for further conceptual material...

There are also open-class forms which capture the same sort of skeletal conceptual structure as closed-class forms; for example, English *enter*, a verb and thus an open-class form, describes roughly the same spatial event as English *into*, which is a preposition, and thus closed-class. In addition, there are a number of languages whose basic structuring of space takes place largely through open-class forms (for example, see the discussion of Mixtec in Chapter 2). The emphasis here then has been not on closed-class forms themselves, but rather on the skeletal conceptual structure that they often carry, whether that in fact appears in closed-class or open-class forms.

As a model of a linguistic process, the system described here presents the following features:

- Its primary responsibility is to match linguistic data concerning the categorization of space, by learning to associate movies of simple 2-dimensional objects with spatial terms from a range of natural languages. Apart from the ability to handle cross-linguistic variation in spatial systems, the model learns without explicit negative evidence (as mentioned above), as children appear to. In addition to the data matched by the core system, extensions to this system are shown to handle the phenomena of polysemy and deixis, among others.[4] The

[3] "The linguistic term *open-class* refers to any set of elements, e.g., noun stems, that is quite large in number and can rather readily add new members. *Closed-class* is applied to a set of elements – e.g. verbal inflections for tense, pronouns, prepositions – that are relatively small in number and fixed in membership." [Talmy, 1983]

[4] These phenomena are described in Chapter 2, and the computational experiments which address them may be found in Chapter 7.

model is able to learn to detect a range of features, such as contact, inclusion, and tightness-of-fit, using the same architectural structures.

- The implementation is neurally plausible. No claim is made regarding structure-to-structure correspondences between the model and actual neural circuitry in the brain, but the model is, as the current phrase has it, neurally-inspired, and features structures which are similar in overall character to particular neural structures.[5]

- The model does more than just recapitulate the data; it also generates falsifiable predictions, regarding the sorts of semantic features, and combinations of features, one might expect to find in lexemes for spatial relations in natural languages. The two central predictions are the following:

  - In any language, any semantically significant static feature which appears in mid-event for some spatial term in the language will appear at end-event for at least one spatial term in that language. For example, the fact that the static feature of inclusion appears in mid-event for English *through* leads us to predict that there will exist a term in English denoting an event ending in inclusion of the trajector in the landmark (such as *in* or *into*).
  - Some languages use the same word to denote static location in some configuration and motion into that configuration. For example, English *in* can be used to mean either static inclusion, or motion into a state of inclusion, synonymously with *into*. The model predicts that this usage will be more likely to appear than the use of the same word to denote either static location in some configuration or motion *out of* that configuration.

  These predictions fall out of the particular architecture and training regimen chosen, as we shall see in Chapter 6.

1.5 Task, Algorithm, and Implementation

There exists a by-now classic philosophical framework for viewing computational modeling efforts due to [Marr, 1982], in which the *computational task*, the fundamental *algorithm* for performing that task, and the *implementation* of that algorithm are viewed separately. These are three distinct levels of explanation at which the functioning of any computational model may be examined. To borrow Marr's example,

> "trying to understand perception by studying only neurons is like trying to understand bird flight by studying only feathers: It just cannot be done.

[5] A discussion of neuroscientific motivation for the architectural structures used here may be found in §5.3.5.

> In order to understand bird flight, we have to understand aerodynamics; only then do the structure of the feathers and the different shapes of birds' wings make sense."

Following Marr's tri-level approach, what we would look for in the case of a model of bird flight is first of all a statement of the task: getting the bird aloft and keeping it there. Next we would ask for an exact specification of how this would be done, at a conceptual level, independent of actual implementation details: this is the algorithmic level, and an explanation in terms of aerodynamics would be in order here, independent of whether this was to be implemented using feathers or balsa wood. Finally, we would ask for a specification of an implementation; in the case of bird flight, feathered wings.

Applying the same framework to the movie-learning task which which this thesis is concerned, we can arrive at the following analysis of the modeling effort described here:

> **Computational Task:** We are given two sets of movies, a training set and a test set. Each movie in the training set is labeled as a positive instance of one of n spatial terms. Each movie in the test set is labeled as either a positive or a negative instance of one of these n terms. The system is presented with the training set, and is to learn to classify movies so that the *error* on the test set is less than some constant ϵ. The error is defined as $E = \frac{1}{2}\sum_i (t_i - o_i)^2$, where i is an index over the movies. Here, t_i is the system's *desired output*, given movie i as input, for the spatial term with which movie i is labeled. This value is 1.0 if movie i is a positive instance of that spatial term, and 0.0 if it is a negative instance. In addition, o_i (where $0 \leq o_i \leq 1$) is the system's *actual output*, given movie i as input, for the spatial term with which movie i was labeled; this can be interpreted as the degree to which the system finds movie i to be a positive instance of that spatial term: if movie i is in fact a positive instance of this spatial term, o_i should be 1.0, and if it is a negative instance, this quantity should be 0.0. Clearly, as the system learns and the actual outputs approach the desired outputs, the error E drops toward zero.
>
> **Algorithm:** The algorithm for accomplishing this task is, very simply, gradient descent in the error measure, for each of a set of complexly interrelated variable parameters. Thus, each variable parameter is repeatedly updated so as to bring about an incremental decrease in the overall error measure.
>
> Notice that while the algorithm itself is trivial, the data structures upon which it operates – i.e. the variable parameters and their interconnection patterns – are not. In fact, most of the "structure" to be explicated at

the algorithmic level resides in these data structures, rather than in the algorithm itself. Unfortunately, these structures do not lend themselves to simple exposition; much of Chapter 5 and Chapter 6 is in fact devoted to this end.

Implementation: The gradient descent algorithm is implemented using error back-propagation in a partially structured connectionist network. The weights on the links in the network correspond to the variable parameters referred to above in the algorithm description, and the structuring of the network corresponds to the interconnection patterns of these parameters.

1.6 The L_0 Project

This thesis is a part of the L_0 project, headed by Jerry Feldman at the International Computer Science Institute in Berkeley. L_0 brings structured connectionist techniques to bear on the issue of miniature language acquisition – more specifically, on the acquisition of nontrivial portions of natural language in the visually-based domain of spatial relations. Thus, the task of the project as a whole is very similar to the task description outlined above for this thesis, except that in the integrated L_0 task specification, the system is presented with entire sentences describing the scene or movie, rather than single words. Thus, L_0 as a whole must deal with cross-linguistic variation not only in semantic structure, but in syntactic structure as well, as well as the complex ways in which one maps onto the other. This is, of course, a much more ambitious project than the one outlined above, and has thus been broken down into parts. This thesis forms one part, and Andreas Stolcke's thesis will form another. His research focuses on precisely those issues that this work does not: acquisition of syntax, and the mapping of syntax onto semantics.

While much of the L_0 work is described in these two theses, there are other persons involved in the project as well, who have contributed in great measure to the progress that has been made. Apart from Jerry Feldman, who heads the project, are George Lakoff, Susan Weber (now at Mills College), and Adele Goldberg. There is, in addition, a group of psychologists and linguists at UC Berkeley who are interested in linguistic representations of space, with whom we have collaborated fruitfully. And perhaps most significantly for the work reported here, in the spring of 1992 a seminar course was offered at UC Berkeley, covering the ideas presented in this thesis. The ideas and system extensions presented in Chapter 7 are the result of work done by the students in that course.

1.7 A Guide to the Remainder of the Thesis

The remainder of this thesis breaks down as follows:

- Chapter 2: **Linguistic Issues** This chapter reviews previous work in linguistics and psychology which raises issues relevant to the work presented here. This chapter is critical for a proper understanding of the linguistic issues which are addressed in this dissertation, but it can be skimmed by the reader interested primarily in the technical aspects of the work.

- Chapter 3: **Connectionism** This chapter supplies a brief overview of the field of connectionism, primarily for the benefit of the previously uninitiated. The central ideas which form the field are discussed, and the idea of error back propagation is presented, in order to orient the newcomer. A number of connectionist approaches to sequence learning are also discussed, and their merits compared, so as to set the stage for the discussion, in Chapter 6, of what the desiderata are for a system which is to learn to classify movies (sequences of frames), and the eventual choice of mechanism for that purpose.

- Chapter 4: **Learning Without Explicit Negative Instances** This chapter presents the problem of learning without explicit negative evidence, and a solution which relies on the use of implicit negative evidence, and the deliberate attenuation of the strength of evidence from these implicit negatives.

- Chapter 5: **Learning Static Scenes** This chapter describes the static feature detecting module which forms a central part of the system described in this thesis, and which, by itself, is capable of learning to linguistically categorize single movie frames, i.e. static scenes, but not movies. Several structural devices which were devised particularly for the spatial semantics learning task are presented here, together with a discussion of learning in such structures. We demonstrate that this module is capable of learning static spatial terms from a variety of languages.

- Chapter 6: **Motion** This chapter describes the system which classifies movies, i.e. the central model whose design and analysis constitute the focus of this dissertation. This system comprises the static feature detecting module described in Chapter 5, along with a means to handle sequences of movie frames over time. The system's performance on a number of spatial terms from different languages is presented, and its implications discussed.

- Chapter 7: **Extensions** This chapter describes a number of extensions to the central system described in Chapter 6, which were undertaken as projects in a seminar course offered at UC Berkeley in the spring of 1992. Among the issues addressed are deixis, polysemy, force dynamics, and alternative means for

handling motion. In addition, remaining open issues on which future research could focus are covered here.

- Chapter 8: **Conclusions** This final chapter begins by reviewing computational work which is similar in goal and scope to the work described here, and concludes by reviewing a number of the main points of the dissertation.

Chapter 2

Linguistic Issues

2.1 The Linguistic Categorization of Space 14
2.2 Cross-Linguistic Variation 14
2.2.1 Mixtec 15
2.2.2 Bengali 16
2.2.3 German 17
2.2.4 Others 17
2.3 A Close Look at English 18
2.3.1 Potential Motion in Static Scenes 18
2.3.2 A Closer Look at *Above* 20
2.3.3 Further Complications 22
2.4 Cognitive Linguistics 24
2.4.1 An Overview 24
2.4.2 Trajectors and Landmarks 25
2.4.3 Prototype Effects in Categorization 25
2.4.4 Deixis 26
2.4.5 Polysemy 28

This chapter describes a number of linguistic issues which are relevant to this thesis, and also presents earlier linguistic work on these issues. The primary function of the chapter as a whole then is to motivate the detailed discussion of the computational model itself which will follow in later chapters, and to give the reader a sense of perspective regarding this model-building enterprise.

There have also been a number of earlier attempts at building computational models whose aim and scope are similar, if not quite identical, to those of the model proposed here. Since these are most easily compared and contrasted with the current model after a detailed exposition of the model, the discussion of this work has been deferred until Chapter 8.

2.1 The Linguistic Categorization of Space

The linguistic categorization of space is a topic that has captured the attention of linguists for a number of years. There is good reason for this: space is an attractive semantic domain in that it is concrete and tangible, and yet still affords a good deal of subtlety in its semantic structure. In this sense it resembles the domain of color, another objectively measureable domain which has attracted work in semantics [Berlin and Kay, 1969].

In addition, as mentioned in Chapter 1, space (along with time) has a privileged position as a fundamental conceptual structuring device in language, while most other domains, color included, do not. [Lakoff, 1987] points out that

> [m]any scholars believe that some parts of a conceptual system are more fundamental than others. Concepts like space and time are usually taken as the most fundamental. Concepts like *chutzpah* in Yiddish or *agape* in Ancient Greek are taken as more superficial. Discussions of conceptual relativity, therefore, tend to be about fundamental concepts like space and time, rather than about less fundamental concepts ...
>
> There are two intuitions behind such a conception of what is fundamental and what is superficial. The first is that concepts that are fundamental, like space and time, are used in many other concepts throughout the system, while concepts like *chutzpah* and *agape* are localized to isolated domains of experience, and therefore don't affect much else. The second intuition is that fundamental concepts tend to be grammaticized, that is, to be part of the grammar of the language. As such, they are used unconsciously, automatically, and constantly.

This chapter presents some of the linguistic issues related to the linguistic categorization of space which are most immediately relevant to this thesis. These issues serve to motivate the work described here, and form an environment which gave rise to the work and within which the work is best understood.

2.2 Cross-Linguistic Variation

The notion of cross-linguistic variation in conceptual systems and modes of thought has long been a subject of fascination for linguists (see [Whorf, 1956] for an early and extremely influential example). Variation of this sort appears in the spatial domain, as even a cursory glance at systems of spatial concepts from a variety of languages will reveal. These differences in spatial systems are sometimes quite dramatic, but more often than not, they are rather subtle, particularly when one compares closely related languages. Examples of both striking and subtle cross-linguistic differences

in spatial categorization are given here. The English spatial system is used as the standard against which others are compared.

In general, cross-linguistic variation in spatial systems provides one motivation for the work addressed by this thesis, by implicitly posing the question "What sort of system could adapt itself to so many different structurings of space, as the human capacity for categorizing space does?"

2.2.1 Mixtec

One language exhibiting a spatial system which is profoundly different from that of English is Mixtec, an Otomanguean language spoken mostly in the state of Oaxaca, Mexico. [Brugman, 1983] presents a semantic analysis of spatial terms in Mixtec,[1] spelling out the manner in which spatial locations are referred to as metaphorical body-parts. For example, to express the notion *The stone is under the table* in Mixtec, one would say

yuù	wã	híyaà	čìì-mesá
stone	the	be-located	belly-table

which, if translated word for word, would yield *The stone is located the table's belly.* Speakers of Mixtec thus appear to view the table metaphorically as a four-legged animal, and the space under the table as the region by the animal's belly. Similarly, to express the notion *The bird is above the tree*, one would say the Mixtec equivalent of *The bird is located the tree's head.* Here, the tree is viewed as a biped, with its head at the top. Brugman supplies many more examples of this sort.

The idea of viewing objects as animate beings, and using body-part names for regions of space near parts of objects, is by no means unique to Mixtec. In English *in back of* we have an instance of the same phenomenon. However, the metaphor is carried far further in Mixtec than in English or related languages. There is in fact no other way in Mixtec to express spatial relations.

It may seem at first that learning this spatial system would require a good amount of world knowledge, e.g. basic anatomy of quadrupeds and bipeds. However, it seems to be the case that much if not all of the categorization is based on primitive, purely perceptual features of the landmark object in question, such as the orientation of its major axis. 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 2.1. Note that both scenes would be classified as *above*, or *over*, in English.

[1] Brugman focuses on the dialect spoken in the village of Chalcatongo, Oaxaca.

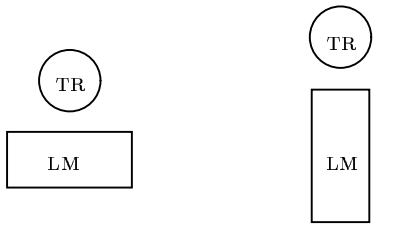


Figure 2.1: Examples of Mixtec *siki* and *šini*

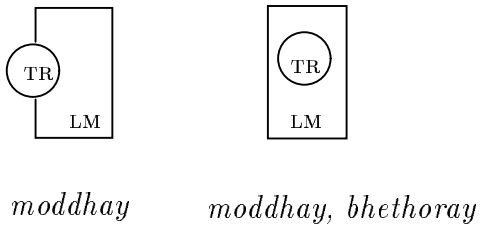


Figure 2.2: Examples of Bengali *moddhay* and *bhethoray*

2.2.2 Bengali

Bengali, which is, like English, an Indo-European language, differs from English in its structuring of space, though much less dramatically than does Mixtec. The notion of "partial inclusion" is significant in Bengali: there are separate lexemes denoting complete and partial inclusion of a trajector in a landmark. This is illustrated in Figure 2.2. The term *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].

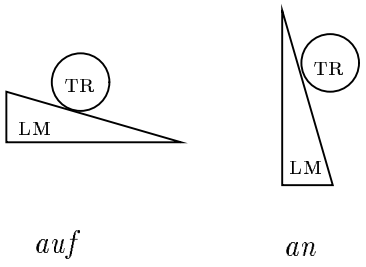


Figure 2.3: Examples of German *auf* and *an*

2.2.3 German

Interestingly, even closely related languages can differ substantively in their spatial systems, as pointed out in [Bowerman, 1989]. Figure 2.3 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*. If the supporting surface is roughly horizontal, as in the scene on the left in Figure 2.3, the preposition *auf* is used to describe the relation. However, if the supporting surface is roughly vertical, as in the scene on the right, the preposition *an* is used instead [Stolcke, 1990a].

2.2.4 Others

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

- Atsugewi: [Talmy, 1983] presents a set of verbal suffixes from Atsugewi, a California Indian language. The suffixes presented mark distinctions which are much finer than those made by English prepositions. For example, there is a suffix expressing such highly specific spatial relations as "over-the-rim into a volume enclosure (e.g. a gopher hole, a mouth)", among many others.

- Cora: Speakers of Cora, a Mexican Indian language, live in a mountainous region, and basic hill shape has been incorporated into their spatial system. They have closed-class morphemes for such aspects of hill shape as "top", "slope", and "bottom" [Casad, 1982].

- 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 [Goldberg, 1991].

- 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 speaker.* 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; Hanania, 1990].

It should be clear from these examples that spatial systems vary cross-linguistically in non-trivial ways, and that the challenge in building a system that will be able to adapt itself to any one of them, (or, more realistically, any of a large number of them) is a significant one. Further examples of cross-linguistic variation will also come up in the discussion, below, of specific linguistic phenomena in various languages. In addition, Chapter 5 discusses the computational modeling of spatial systems from a variety of natural languages.

2.3 A Close Look at English

Another motivation for the work described in this thesis is the fact that spatial terms, including ones we are very familiar with, possess a semantics which is far more subtle than might be expected at first glance. In this section, we examine some evidence for this contention.

(**Note:** In the examples here, and in the bulk of the remainder of the thesis as well, we restrict our attention to simple *convex* objects. Chapter 7 discusses the use of convex hulls in handling non-convex objects.)

2.3.1 Potential Motion in Static Scenes

Figure 2.4 presents two scenes; the one on the left is generally taken as a good example of English *above*, while the one on the right is not. The only difference between the two is the absence of the "supporting" triangle in the scene on the left. Thus, it is clear that the physical context in which a spatial relation occurs can profoundly influence the way it is categorized linguistically (see [Herskovits, 1986] for further examples of the effect of physical context in English linguistic categorization of spatial relations).

This example is interesting because it helps to weed out some naive first-cut characterizations of *above*, such as "the center of mass of the trajector is located higher

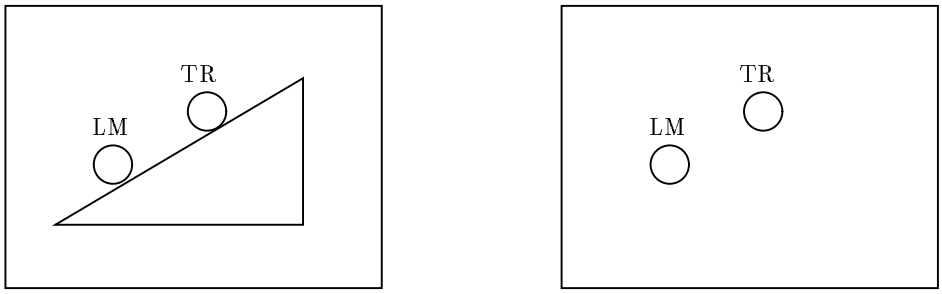


Figure 2.4: The role of context

---

in the visual field than the center of mass of the landmark". This characterization is clearly not correct, since it is unaffected by physical context.

A more satisfying approach can be motivated by some recent work in perceptual psychology. [Freyd *et al.*, 1988] present intriguing evidence that humans viewing static scenes (such as the ones in Figure 2.4) mentally represent the forces that are acting on the objects, keeping them where they are. They presented subjects with three pictures of a potted plant, first supported either from below (sitting on a table) or from above (hanging from a hook), then hanging in mid-air without support at the same height as the first picture, and finally in mid-air without support in either the original position or slightly above or below the original position. Subjects were asked whether the final frame showed the plant at the same height as the first two. The results indicated that subjects had a tendency to misjudge the position of the plant in the final frame, often judging it to be at the original position when it was in fact slightly lower. This result was not contingent on whether the plant was originally supported from above or below. The authors take this, along with some similar experiments, as evidence that "subjects represent the underlying dynamics of coherent static scenes, so that when the forces are suddenly unbalanced, the mental representation includes an unfreezing of the potential motion implicit in the scene."

This notion of unfreezing of potential motion is helpful in considering the case of Figure 2.4. For if potential motion is part of what humans mentally represent when they perceive static scenes, then English *above* could well have a component of such potential motion in it. One possibility is that part of what *above* denotes is the possibility of the trajector striking the landmark if it (the trajector) is allowed to fall under the influence of gravity. More specifically, if the *direction of potential motion*, (see Figure 2.5) tends to lead the trajector to the landmark, this could be taken as evidence for *above*. This would explain Figure 2.4, since the direction of potential motion leads the trajector to the landmark in the scene on the left, but not in the scene on the right, and can also explain some more straightforward instances

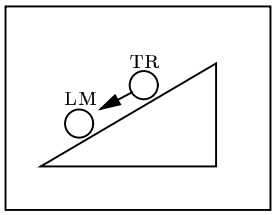


Figure 2.5: Direction of potential motion

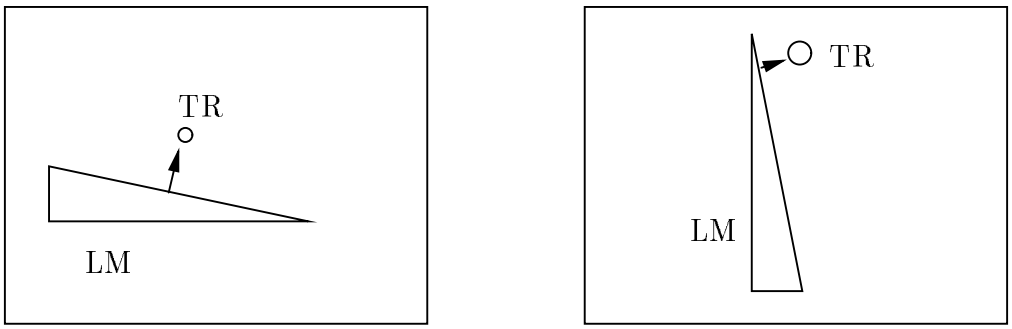


Figure 2.6: Good and poor examples of English *above*

of *above*.[2]

As we shall see, the direction of potential motion does not suffice for characterizing even as simple a term as *above*, but it almost certainly does play an important part. Further elements are presented below. For further discussion of the direction of potential motion, see §7.1.3, which presents experiments focusing on the role of this directional primitive feature in the semantics of another English preposition, *on*.

2.3.2 A Closer Look at *Above*

Figure 2.6 presents one good and one poor example of English *above*. Note that in both cases, the direction of potential motion of the trajector, were it to suddenly

[2]Further evidence for the imputation of motion to static scenes, albeit under somewhat different circumstances, comes from a phenomenon which [Talmy, 1990] refers to as "fictive motion". This appears in such sentences as "Those rods *go* through the ceiling" and "This road *goes* through Red Bluff", where a verb of motion and a preposition are used to describe the static relationship of an elongated trajector to a landmark, as if the speaker were mentally scanning along the length of the trajector, and describing the relationship of his/her region of focus to the landmark. See also the discussion of implicit search paths in §7.2.4.

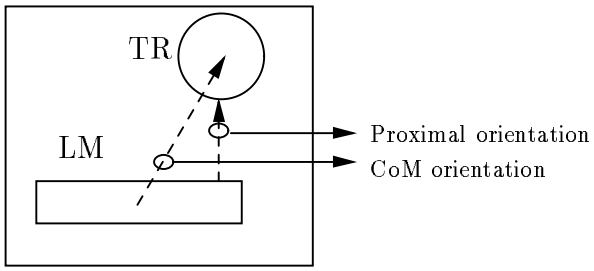


Figure 2.7: Proximal and center-of-mass orientations

---

be released from its position, would tend to lead it toward the landmark. The fact that these two cases are not equally good examples of *above* thus indicates that there is more to the concept than the simplistic characterization outlined above, relying exclusively on the direction of potential motion of the trajector.

The point of view taken by this research is that even very simple spatial concepts like *above* in fact involve a combination of evidence from various sources, the direction of potential motion being only one of these. We proceed now to outline two other such sources of evidence.

Two perceptual primitives which seem to play a role in the linguistic categorization of space are the *proximal orientation* and *center-of-mass orientation* of a scene. These both specify the location of the trajector with respect to the landmark, but in different ways:

- **Proximal orientation** is the orientation of the (imaginary) directed line segment connecting the landmark to the trajector where the two objects are closest.
- **Center-of-Mass orientation** is the orientation of the (imaginary) directed line segment connecting the center of mass of the landmark to that of the trajector.

Figure 2.7 illustrates both the proximal and center-of-mass orientations in a single scene.

A motivation for introducing the proximal orientation as a primitive comes from Figure 2.6. Recall that the direction of potential motion did nothing to differentiate these two cases, one of which is a good example, and one a poor one, of English *above*. The proximal orientation, on the other hand, is very different in the two cases: in the scene on the left, it is very nearly upright vertical (and this corresponds to our intuitive notion that *above* is intimately related to upright vertical), while in the scene on the right, the proximal orientation is far from upright vertical. Thus it is at least possible that this difference is what causes the scenes to be judged differently as examples of English *above*.

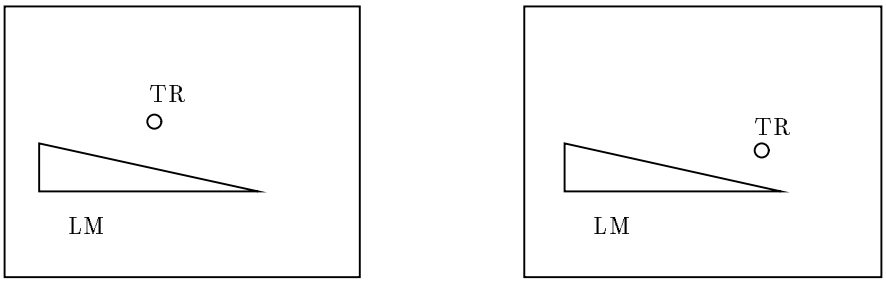


Figure 2.8: More good and poor examples of English *above*

However, this cannot be the entire story either. Figure 2.8 presents two more examples of English *above*, again, one good and one poor. The one on the left is essentially identical to the left-hand scene in Figure 2.6. The one on the right, on the other hand, was produced from the one on the left by sliding the trajector down the slope of the triangle until it was near the bottom. Notice that the direction of potential motion and the proximal orientation do not vary across these two scenes. Thus, some other feature must account for the difference in judgment. The center-of-mass orientation does vary, and may well be the deciding factor.

While some of the perceptual primitives proposed here are in fact used in the system described in this thesis, the overall idea here has been not so much to propose specific primitives and claim that they provide the only means of characterizing simple spatial concepts like English *above*, but rather, to point out that apparently simple spatial concepts possess a semantics more subtle than might be imagined, and to indicate a number of features that might be relevant to a complete characterization of such concepts. While this list is by no means meant to be exhaustive, it is meant to give a flavor for the sorts of features that may play a role in the linguistic categorization of spatial relations. In addition, it is meant to highlight the very general notion that the categorization of spatial relations appears to rely upon the combination of evidence from a number of perceptual features.

2.3.3 Further Complications

The examples presented here, unfortunately, barely begin to scratch the surface of the semantics of spatial prepositions in English (let alone those of other languages). [Herskovits, 1986] presents an array of further subtleties, such as the use of *in* to describe the relation of the pear to the bowl in Figure 2.9, despite the fact that the pear is not physically contained within the interior of the bowl.

One of Herskovits' central claims is that

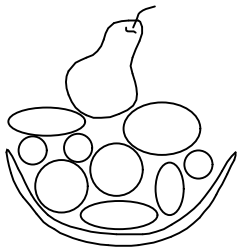


Figure 2.9: "The pear is *in* the bowl." (from Herskovits)

> a large number of special conventions occurs in the spatial uses of the prepositions, but the motivation for the choice of preposition is generally quite clear, and lies with a link to some geometric ideal – the essence of the preposition's meaning. Since the number of special conventions must of necessity be finite, this led to asserting that the set of uses of a preposition divides into *use types* derived from a geometric ideal. ... The metaphor of a musical theme (the ideal meaning) with variations (use types) is instructive, in that, though there is no question that a resemblance is seen between the theme and a variation, there are no simple rules that transform one into the other. There are many different ways the ideal meaning can be manifested in a use type. So, though it is clear that it makes sense to use a given preposition in some situation, it would often also make sense to use another. *Only convention justifies the correct choice.* (Emphasis added).

This is similar in spirit to the viewpoint, voiced by [Lakoff, 1987], that it is unhelpful to view the relation between the "core" or central sense of a word to other senses as either, on the one hand, entirely predictable, or, on the other hand, entirely arbitrary. Instead, there is usually a *motivation* for the particular senses found, even though other ones might make as much sense. The role played by such a notion of non-arbitrary convention in the use of spatial terms is, as Herskovits amply demonstrates, a central one.

It is, unfortunately, not one that this thesis will be able to address fully, although Chapter 7 contains descriptions of some work in that direction (see §7.1.1). In general, of course, the approach taken here has been to restrict the domain to simple 2-dimensional objects; this simplifies matters considerably, in that many (though by no means all) of the problems that Herskovits points out do not arise. Despite this simplification, however, the resulting domain is still rich in semantic subtleties, as the examples prior to this section should indicate.

2.4 Cognitive Linguistics

The work presented in this thesis springs partly from recent work in *cognitive linguistics*. The purpose of this section is to acquaint the reader with the general philosophy of cognitive linguistics, and to thus provide a fuller understanding of the context within which the current work takes place. On a more specific level, the purpose is to present a number of issues that cognitive linguists have concerned themselves with, and which are dealt with in this thesis, mostly as extensions to the central architecture, in Chapter 7.

2.4.1 An Overview

[Langacker, 1987], in introducing the field of cognitive linguistics, begins by stating that

> [l]anguage is an integral part of human cognition. An account of linguistic structure should therefore articulate with what is known about cognitive processing in general, regardless of whether one posits a special language "module" [Fodor, 1983], or an innate *faculté de langage*. If such a faculty exists, it is nevertheless embedded in the general psychological matrix, for it represents the evolution and fixation of structures having a less specialized origin. Even if the blueprints for language are wired genetically into the human organism, their elaboration into a fully specified linguistic system during language acquisition, and their implementation in everyday language use, are clearly dependent on experiential factors and inextricably bound up with psychological phenomena that are not specifically linguistic in character. Thus we have no valid reason to anticipate a sharp dichotomy between linguistic ability and other aspects of cognitive processing. Instead of grasping at any apparent rationale for asserting the uniqueness and insularity of language, we should try more seriously to integrate the findings of linguistics and cognitive psychology.

Given that this is the general thrust of the field of cognitive linguistics, it should be no surprise that one of the topics cognitive linguists have focused on is the relation of language to perception, specifically, the linguistic categorization of spatial relations [Lindner, 1982; Casad, 1982; Vandeloise, 1991; Matsumoto, 1989; Janda, 1984; Brugman, 1981; Brugman, 1983; Talmy, 1983]. Apart from outlining the structure (and relation to other cognitive processes) of a number of non-Western spatial systems, as recounted above, these and other researchers have addressed a cluster of issues which impinge on the work described here. These are described below.

2.4.2 Trajectors and Landmarks

The terms *trajector* and *landmark*, introduced in Chapter 1, are drawn from [Langacker, 1987], who characterizes these two roles that entities may play in a relation as follows:

> In virtually every relational predication, an asymmetry can be observed between the profiled participants. One of them, called the **trajector** *(tr)*, has special status ... The term trajector suggests motion, and in processual predications describing physical activity (presumably the prototype for relations) the trajector generally does move through a spatial trajectory. Note, however, that the definition makes no reference at all to motion, either physical or abstract, so this schematic description is applicable to both static and dynamic relations. Other salient entities in a relational predication are referred to as **landmarks** *(lm)*, so called because they are naturally viewed (in prototypical instances) as providing points of reference for locating the trajector.

These characterizations are are essentially the same as the concepts of *figure* and *ground* described in [Talmy, 1983], such that figure corresponds to trajector, and ground corresponds to landmark.

Every movie dealt with in this thesis contains a single trajector, either moving or statically located relative to a single landmark object.

2.4.3 Prototype Effects in Categorization

Eleanor Rosch's work on the phenomenon of *prototypicality* in categorization [Rosch, 1973; Rosch, 1977; Rosch, 1978] has attracted the attention of several cognitive linguists [Lakoff, 1987; Langacker, 1987]. The attraction which prototype-based categorization has held can be seen as stemming from an awareness of serious problems encountered with the idea of defining natural language categories by lists of necessary and sufficient conditions for membership. The basic issue here is that some members of a category are better exemplars of the category than others; they are in some manner more central, in a word, more *prototypical* of the category. For example, most native English speakers would agree that a robin is a better example of a bird than is an ostrich or a kiwi, just as a lion is a better example of a mammal than is a platypus. Under a theory of categorization which considers membership in a category to be contingent only on meeting a set of necessary and sufficient conditions, there is no room for such a notion of gradation of membership in a category.

The phenomenon of prototypicality appears in the domain of spatial relations as well, as can be seen in Figure 2.10. Here, the scene on the left is a prototypical instance of English *above*, while the one on the right, while clearly an instance of

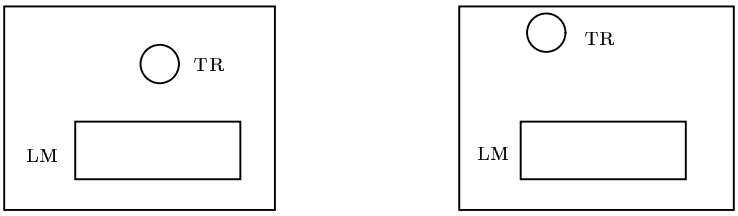


Figure 2.10: Prototypical and non-prototypical cases of *above*

above, is not nearly as prototypical an instance of the concept. There will be more to say on the subject of prototypicality effects later on (see §5.4.1); for the time being, it suffices to point out that a prototypical case of *above* appears to be one for which there is strong evidence from each of the various factors that might contribute to "above-ness". For example, notice that in the prototypical case on the left in Figure 2.10, the center-of-mass orientation and the proximal orientation are both perfectly aligned with upright vertical,[3] while the direction of potential motion of the trajector, were it to be released, is straight down toward the landmark. In the non-prototypical case on the right, the proximal orientation is still upright vertical, and the direction of potential motion of the trajector is still directly downward, but the center-of-mass orientation is no longer perfectly aligned with upright vertical. This is a possible cause for the scene not being judged as a prototypical case of English *above*.

2.4.4 Deixis

Deixis is the effect that the physical setting of a speech event may have on the way in which an utterance is interpreted. For example, anyone in Berkeley, which is located just across the Bay Bridge from San Francisco, would be able to truthfully say

San Francisco is just across the Bay Bridge

while someone in Los Angeles would not, generally speaking. Notice that Berkeley is never explicitly mentioned in the sentence. It is, however, implicitly the *deictic center*, or center of discourse, for the sentence, and this is why the sentence makes sense when spoken in Berkeley. Note, however, that if the speaker were in fact in Los Angeles, but were describing the layout of the San Francisco Bay Area, it would be perfectly legitimate to, in the process, "anchor" the description in Berkeley by saying

[3] Recall that the center-of-mass orientation is defined as the orientation of the (imaginary) directed line segment connecting the center of mass of the landmark to that of the trajector, while the proximal orientation is the orientation of the (imaginary) directed line segment connecting the landmark to the trajector where the two objects are closest.

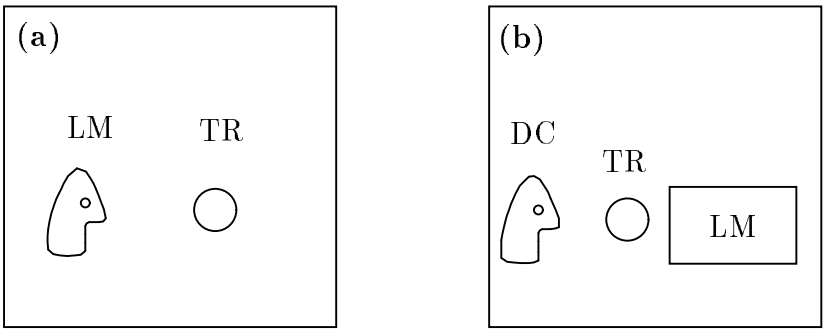


Figure 2.11: Deixis: English *in front of*

"Berkeley is a city on the east side of the Bay", and then use the above sentence. In this latter case, the deictic center has been shifted to Berkeley, despite the fact that the speaker is actually in Los Angeles.

Consider Figure 2.11. Here, the face-like figure denotes an object with an inherent front and back, such as a human being.[4] Both (a) and (b) are examples of English *in front of*. In (a), the inherent front of the landmark object is facing the trajector, therefore the TR is *in front of* the LM. This is the non-deictic sense of *in front of*, the one encountered in sentences such as

Kim is standing *in front of* Pat.

In (b), however, something different is happening. Here, the landmark object has no inherent front and back of its own. Instead, the fact that the deictic center *(DC)* is facing the landmark imputes a front and back to the landmark, such that the front of the landmark is the side facing the deictic center. Now, since the trajector is also on that side of the landmark, the TR is considered to be *in front of* the LM. This is the deictic sense of the term, such as one might encounter in sentences like

The ball is lying *in front of* the tree

provided there was a speaker on the scene to provide an orientation to the tree.

It is interesting to note that there are languages which handle deixis differently. For example, Hausa, a west African language, treats deictic effects such that Figure 2.11(b) would be considered an example of *in back of* [Hill, 1977]. This happens since in Hausa, the side of the landmark object which is facing away from the deictic center is implicitly taken as the front, rather than the side facing towards the DC, as

[4]This convention is adopted from [Vandeloise, 1991].

Figure 2.12: The polysemy of *in*

is done in English. There are, in addition, languages with far more elaborated and intricate deictic systems than that found in English. An example of such a language is the Eskimo language Inuktitut [Denny, 1980; Denny and Issaluk, 1976], which has deictic terms denoting such specific notions as "down there (used for a large trajector)", "down there (used for a small trajector)", along with many others.

For the time being, it suffices to note that deixis is a phenomenon that can affect the construal of even very simple scenes, such as those in Figure 2.11; it is thus a relevant issue for this thesis. §7.1.2 presents computational work concerning deixis, using an extension to the system whose design we shall be reviewing in the upcoming chapters.

2.4.5 Polysemy

Polysemy is a linguistic phenomenon whereby a single word has a cluster of distinct but related senses. Let us consider a very simple example to begin with: the polysemy of *in*, as illustrated in Figure 2.12. Here, (a) depicts a small circular trajector within a rectangular landmark, and (b) depicts the same trajector moving into that landmark. These two pictures correspond to the senses of *in* found in the following sentences:

(a) "The circle is *in* the rectangle."

(b) "The circle went *in* the rectangle."[5]

Notice that there is a simple generalization, a simple abstraction over these two senses, which will allow us to differentiate positive from negative instances of *in*: if the trajector ends up inside the landmark, the scenario is an instance of *in*, regardless

[5]This usage sounds odd to some people, who would rather use *into*. It is certainly acceptable to my ears, however, and in any event, an analogous usage of the word in the sentence "He walked *in* the room" meets with near universal approval.

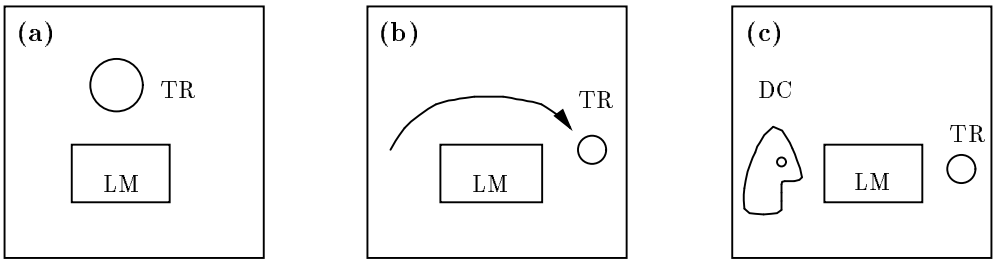


Figure 2.13: The polysemy of *over* (after Brugman)

---

of whether or not the trajector started there – in fact, regardless of whether or not it moved at all. This means that in this case there is no principled reason to consider the two senses distinct, since the phenomena can be accounted for using a simpler explanation, namely the above abstraction over the two senses.

On the other hand, there are a number of prepositions for which no such obvious abstraction over the various senses exists. For example, [Brugman, 1981] presents a semantic analysis of the English preposition *over*, detailing literally dozens of different senses. These do not lend themselves to the straightforward sort of analysis we offered for *in*. Rather, there seems to be a cluster of demonstrably distinct senses of this word. Figure 2.13 illustrates three of these senses, the ones that correspond to the use of *over* in the following sentences:

(a) "The circle is *over* the square."

(b) "The circle went *over* the square."

(c) "The circle is *over* the square (from here)."[6]

Notice that there is no obvious abstraction covering these three senses. One possible abstraction that comes to mind takes a scenario to be an instance of *over* if the trajector is ever above the landmark. But this is clearly not correct, since a trajector which moves from the left side of the landmark to the region above the landmark, and then back to the left of the landmark again cannot be said to have gone *over* the landmark, even though it fits the abstraction. In addition, (c) above does not fit this generalization (although the imagined path in (c) passes above the landmark). So a

[6]Brugman views this latter sense, an *end-point focus* sense of *over*, as deriving from sense (b), in that the trajector is located at the end of an imagined path originating at the deictic center, and passing over the landmark, just as the actual path does in (b). See §7.2.4 for further discussion of implicit paths.

more detailed analysis is required, pointing out the various senses and their relations with one another, along the lines of [Brugman, 1981].

Polysemy is widespread throughout the lexicon. More to the point, it occurs even in the highly simplified visual domain we are studying here, which indicates that it is an issue that we will have to address. This has been done, at least to a degree: §7.1.1 presents preliminary computational experiments in acquiring various senses of the English prepositions *in, over*, and *under*.

As we have seen in this chapter generally, there are a number of interesting semantic issues that crop up in our simplified domain, perhaps more than might be expected. The upcoming chapters describe a modeling effort directed at addressing these issues, beginning with an overview of the general paradigm adopted – connectionism – and then moving on to the details of the model itself.

Chapter 3

Connectionism

3.1 Overview 31
3.2 Structured Connectionism 34
3.3 Back-Propagation 37
3.4 Learning Sequences Using Back-Propagation 38
3.4.1 Back-Propagation Through Time 38
3.4.2 Time-Delay Neural Networks 41
3.4.3 Back-Propagation with State Units 42
3.4.4 Discussion 43

This chapter presents an introduction of the field of connectionism, intended primarily to orient the otherwise uninitiated. After a necessarily brief overview of the field as a whole, focusing in particular on the back-propagation algorithm, we move on to discuss traditional connectionist methods for sequence-recognition. This issue of sequence-recognition is of relevance to this thesis since the movies which the central system will learn to categorize are sequences of static frames.

3.1 Overview

Connectionism is a neurally-inspired style of computation in which many simple interconnected processors cooperatively perform some computational task [Feldman and Ballard, 1982; Ballard, 1987a]. This field, also known as the field of neural networks, has had a profound impact in recent years on the way in which computation is conceived of, primarily by providing an alternative to the classic von Neumann model of a program residing in memory dictating the actions of a central processing unit. Under the connectionist view, there is no single sophisticated central processor; rather, there are a large number of quite simple ones, which correspond roughly to biological neurons, interconnected by links, which correspond roughly to axons and dendrites – the fibers by means of which actual neurons transmit and receive signals.

Figure 3.1 presents a very simple connectionist network of a particular sort known as a *two-layer perceptron*. This network consists of three layers of units, or nodes

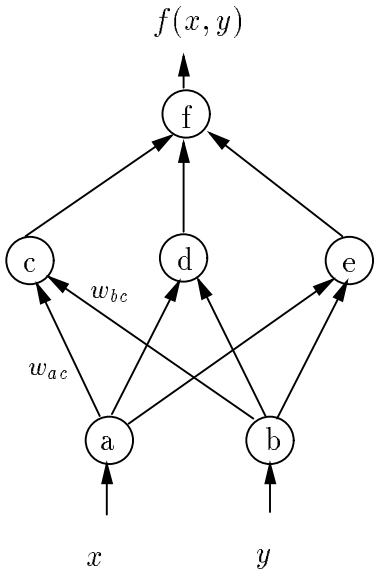


Figure 3.1: A simple connectionist network – a 2-layer perceptron

---

(somewhat unhelpfully, input units are generally not considered when counting perceptron layers). The bottom layer, consisting of units **a** and **b**, consitutes the *input layer*; these two units receive external inputs x and y respectively. Units **c**, **d**, and **e** make up the intermediate *hidden layer*; these units are neither input units nor output units, but receive signals from the input units, perform some computation on them, and pass signals on to the output unit. Finally, unit **f** constitutes the *output layer*, and produces some function $f(x,y)$ of the input by virtue of its connections to the rest of the network. At a given point in time, each unit in the network will have an activation level, which will be affected by the activation levels of those units whose outgoing links lead into it, and which will in its turn affect the activation levels of those units to which its outgoing links are connected. The links which connect units in the network are typically weighted; for example, the link connecting unit **a** to unit **c** in the figure is shown with weight w_{ac}. Under this scheme, each unit receives as input the weighted sum of the activations of the units that have links leading into it. For example, unit **c** in the figure will receive as input $w_{ac}a + w_{bc}b$, where a and b are the activation levels of nodes **a** and **b**, respectively. Unit **c** will then produce as its activation level some function of these two inputs.[1] The other two hidden units will similarly compute their activation levels, and finally, the output unit will compute its activation as a function of the activations of the hidden units.

[1]Commonly, the function of a unit is the sigmoidal squashing function $f(x) = 1/1 + e^{-x}$.

This two-layer perceptron is an instance of a particular sort of network known as a *feed-forward* network, since the flow of activation is always in one direction, from input units to output units. The alternative form is a *recurrent* network, one in which there exists a path from a unit back to itself, either directly or through other units.

The field of connectionism has had a checkered history. Early work [McCulloch and Pitts, 1943; Hebb, 1949] held out the promise of a relatively simple and easily understood biological grounding for our understanding of intelligence, and [Rosenblatt, 1962] in particular excited interest with a proof of convergence for an algorithm to train the weights of simple one-layer perceptrons (i.e. perceptrons without hidden units), so that a given perceptron could learn to adapt itself to various tasks. This excitement was quashed by the publication of the book *Perceptrons* [Minsky and Papert, 1969], which pointed out that some very simple computations were in principle unlearnable by one-layer perceptrons. The most famous example of such a function is the *exclusive-or*, a function of two (binary) arguments which returns 1 when the inputs are different, i.e. one is 0 and the other is 1, and returns 0 if they are the same. A two-layer perceptron (such as the one in Figure 3.1) is required for this function. And since no training algorithm for multi-layer perceptrons was known at the time, this served to dampen a good deal of the earlier enthusiasm engendered by Rosenblatt's proof.

This state of affairs continued until the recent resurgence of interest in multi-layer perceptrons, brought about by the development of a training algorithm which is able to train weights in such a perceptron. This is the *back-propagation* algorithm [Rumelhart *et al.*, 1986].[2] This, together with somewhat earlier work in neurally-inspired computational models based on ideas from statistical mechanics [Hopfield, 1982; Hopfield, 1984], led to a renewal of interest in the field of connectionist or neural network models generally. Recent work in the field has seen the application of connectionist networks to a wide variety of tasks, such as neurobiological modeling [Lehky and Sejnowski, 1988; O'Reilly *et al.*, 1990], speech recognition [Waibel *et al.*, 1987; Morgan and Bourlard, 1989; Renals *et al.*, 1991; Waibel *et al.*, 1991; Osterholtz *et al.*, 1992], natural language processing and inference [Waltz and Pollack, 1985; Cottrell, 1985; Elman, 1988; Shastri, 1988; Fanty, 1988; Weber, 1989b; Miikkaulainen, 1990; Stolcke, 1990b; Jain *et al.*, 1992], vision [Ballard, 1987b; Sejnowski and Hinton, 1987; Olson, 1989; LeCun, 1989; Hummel and Biederman, 1990; Poggio and Edelman, 1990; Ahmad and Omohundro, 1990; Ahmad, 1991; Mozer *et al.*, 1991; Keeler *et al.*, 1991] and purely theoretical work probing the limits of connectionist networks as computational devices [Cybenko, 1989; Kruglyak, 1990; Siegelman and Sontag, 1991]. Computational mechanisms similar in flavor to connectionist networks, such as Markov random fields, have also enjoyed widespread attention [Cross and Jain, 1983; Geman and Geman, 1984; Chou and Raman, 1987;

[2] It is interesting to note that the algorithm had actually been discovered long before this [Werbos, 1974], but not publicized widely. The rediscovery of the algorithm over 10 years later had a much greater impact than the original discovery.

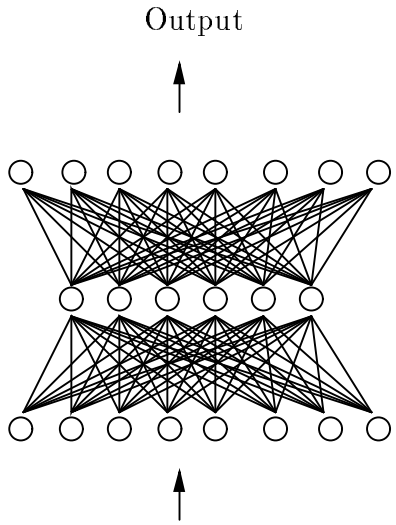


Figure 3.2: An unstructured 2-layer perceptron

Cooper, 1989; Regier, 1991e; Regier, 1991d]. An excellent overview of the field of connectionism as a whole can be found in [Hertz *et al.*, 1991], while [Hinton, 1990] presents a review of connectionist learning mechanisms.

3.2 Structured Connectionism

One may distinguish two general views on the pre-structuring of connectionistist networks. One viewpoint holds that with learning algorithms such as back-propagation at our disposal, we need not put much effort into network design, but may rather let the algorithm induce whatever structure is inherent in the training set. Recent work in neuroscience, demonstrating the extreme plasticity and flexibility of structures found in the brain provides a possible source of inspiration for this view [Merzenich and Kaas, 1982; Métin and Frost, 1989]. For if neural structures in the brain change upon exposure to training stimuli, the argument would run, why should connectionist networks be so rigid as to disallow that possibility? Figure 3.2 presents an unstructured connectionist network which one might use if one were an advocate of this position. Note that there is full connectivity between layers here; this unrestricted inter-layer connectivity leaves the representations formed in the intermediate hidden layer free

to adapt themselves to the training set in any of a very large number of ways, thus capturing the flexibility that is central to the appeal of this viewpoint.

The opposing viewpoint, the one adopted in this thesis, is that of *structured connectionism* [Feldman *et al.*, 1988]. While not denying the appeal of the notion of flexible representations, this approach holds that it is unrealistic to expect unstructured networks to be able to be able to handle large, complex problems in inherently structured domains, such as language and vision, and that a more promising route is the incorporation into the network architecture of knowledge regarding the target domain. [LeCun, 1989] notes that such incorporation of prior knowledge will tend to lead to superior generalization from the training set:

> Considering [back-propagation] as a general learning rule that can be used as a black box for a wide variety of problems is, of course, wishful thinking. Although some moderate sized problems can be solved using unstructured networks, we cannot expect an unstructured network to generalize correctly on every problem. ... [G]ood generalization performance *can* be obtained if some *a priori* knowledge about the task is built into the network. Although in the general case specifying such knowledge may be difficult, it appears feasible on some highly regular tasks such as image and speech recognition.
>
> Tailoring the network architecture to the task can be thought of as a way of reducing the size of the space of possible functions that the network can generate, without overly reducing its computational power. Theoretical studies [Denker *et al.*, 1987; Patarnello and Carnevali, 1987] have shown that the likelihood of correct generalization depends on the size of the hypothesis space (total number of networks[3] being considered), the size of the solution space (set of networks that give good generalization), and the number of training examples.

Thus if we were to hold everything else constant, better generalization from the training set could be obtained through decreasing the number of networks being considered, i.e. reducing the number of free parameters in a given architecture — and this means pre-structuring the architecture. So the argument for structured connectionism essentially rests on the relative impracticality of the unstructured approach, in that it is unreasonable to expect good generalization performance from a network in a complex domain without building in knowledge of the domain so as to constrain the search for solutions. Of course, this need for prior structuring, or biasing, in generalizing learning systems is by no means restricted to connectionist learners – it affects any machine learning system, as pointed out in [Mitchell, 1980].

Figure 3.3 presents a structured architecure of the sort described in this thesis, for comparison with the unstructured network of Figure 3.2. The structural devices

[3]By the term "network", LeCun means an architecture with a specific set of weights on its links.

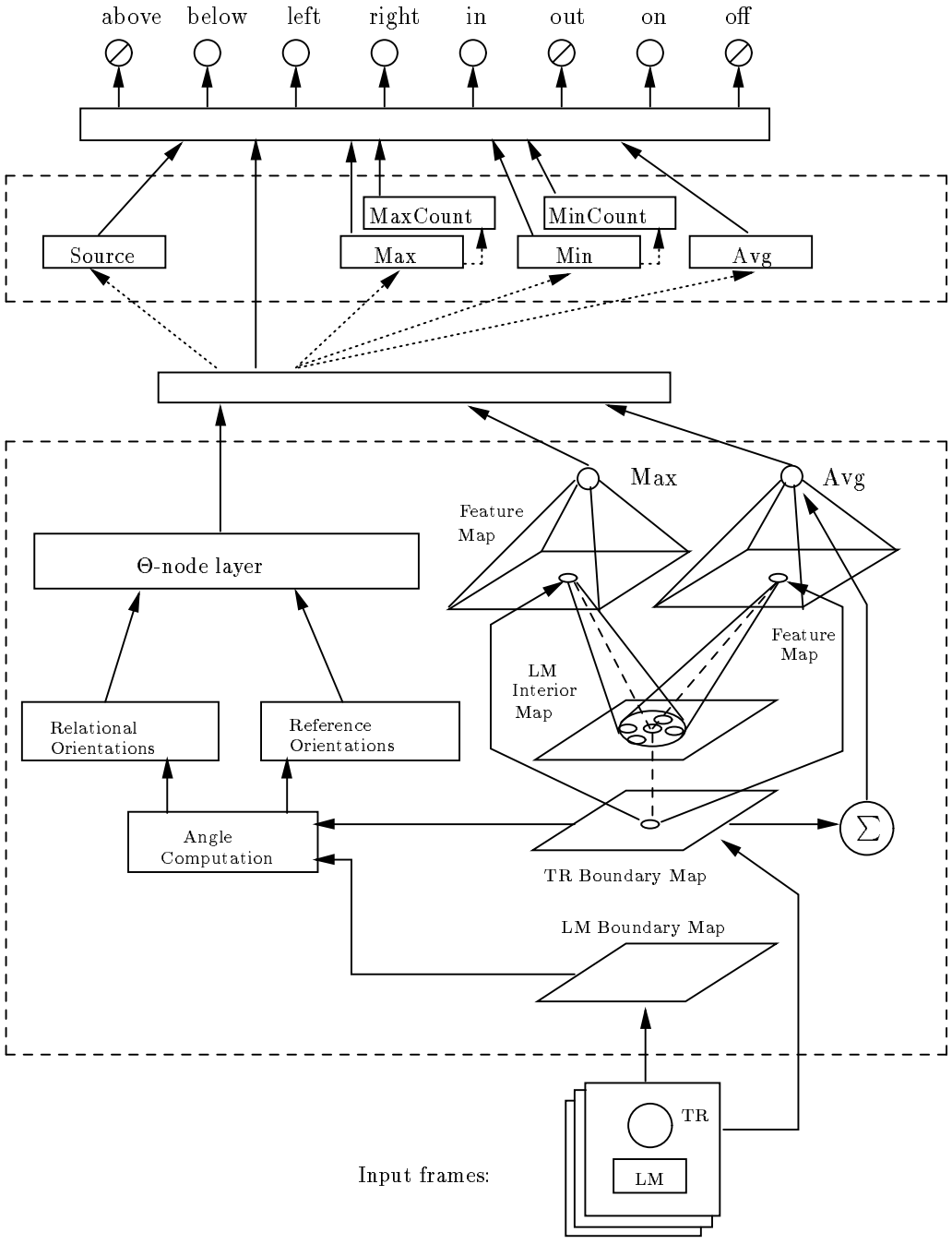


Figure 3.3: A structured network

in those sections of the architecture displayed in dotted outline serve precisely the role outlined above by LeCun: they reduce the dimensionality of the space which the network must search through, in the process of learning, to find a solution. Thus, the way in which these structural devices are designed is critical to the success or failure of such a network. If they are designed well, accurately incorporating knowledge of the domain into the network architecture, the hypotheses considered during the learning process will be restricted to those that are consistent with the knowledge of the domain which the network designer was trying to capture. This will simplify the learning process, and, as mentioned above, enhance generalization from the training set. The motivation and design of these structures is the central focus of Chapter 5 and Chapter 6.

Many connectionist learning systems are structured to one degree or another. Examples of connectionist systems which incorporate knowledge of the target domain into their architectures are the hand-written digit recognizers of [LeCun, 1989; Keeler *et al.*, 1991], the equilateral triangle recognizer of [Ahmad and Omohundro, 1990], the parser of [Jain *et al.*, 1992], the object segmentation and recognition systems of [Mozer *et al.*, 1991; Hummel and Biederman, 1990; Poggio and Edelman, 1990], and the perceptually-grounded lexical semantics acquisition systems of [Nenov, 1991; Nenov and Dyer, 1988; Regier, 1991b; Regier, 1991c]. In addition, [Nowlan, 1990; Jacobs *et al.*, 1990] have investigated the use of modular structure in general-purpose learning networks. These lists of course do not approach being exhaustive; they are merely meant to be indicative of the variety of connectionist models which embody this notion of prior structuring of the architecture.

3.3 Back-Propagation

This section very briefly outlines the basic ideas behind the back-propagation algorithm. For a full exposition, see [Rumelhart *et al.*, 1986].

When being trained under back-propagation, a network takes an input vector as input, and produces an output vector at the output node(s). This output pattern is then compared with the desired output for that input pattern; the overall goal is to reduce the difference between the desired and actual outputs, i.e. to reduce the *error*. The formula for the error is

$$E = \frac{1}{2}\sum_{i,p}(t_{i,p} - o_{i,p})^2 \tag{3.1}$$

where i indexes over output nodes, and p indexes over input patterns. Here, $t_{i,p}$ is the desired output, and $o_{i,p}$ is the actual output, for node i on pattern p. Thus, the above quantity is simply the difference between the observed and desired outputs, squared, and summed over all output nodes and all patterns.

The essence of the back-propagation algorithm is that it reduces error by performing *gradient descent* in this error measure. This means that for each weight w_{ij} in the network, it computes the gradient $\frac{\partial E}{\partial w_{ij}}$, i.e. the partial derivative of the error with respect to that weight,[4] and then changes the weight by a small amount in the opposite direction, so as to decrease the error:

$$\Delta w_{ij} = \epsilon \times -\frac{\partial E}{\partial w_{ij}} \tag{3.2}$$

where ϵ is the *learning rate*, a constant factor.

Thus, with each weight update, the overall error will decrease, eventually reaching an amount so small as to be negligible. At this point, the network has learned the training set.

The back-propagation algorithm gets its name from the fact that in order to compute the gradient, it must *back-propagate* an error term backwards through the net. In order for it to do this, the activation functions of the nodes in the network must be differentiable. This is of relevance for us here, since in Chapter 5 we will be discussing nodes with activation functions that were designed specifically for the spatial concept learning task of this thesis; it will critical that these activation functions be differentiable if learning is to take place on the weights that lead into these nodes.

3.4 Learning Sequences Using Back-Propagation

The issue of learning sequences using back-propagation is of relevance for this thesis: a movie portraying an event is after all simply a sequence of static frames. More specifically, it is of relevance since the system described in this thesis uses back-propagation for the learning of these sequences.

The actual method used to learn these movies will be presented in Chapter 6. For the time being, we present three more traditional sequence learning methods, so as to have something with which to compare the method eventually chosen. These three common methods of using back-propagation in the learning of sequences are *back-propagation through time, time-delay neural networks,* and *back-propagation with state units.* They are discussed in turn below.

3.4.1 Back-Propagation Through Time

Consider Figure 3.4. This is the simplest possible recurrent network, and while it is exceedingly limited in computational power, with the appropriate weight on its single link, it would be able to distinguish certain sorts of sequences. For example, if the

[4] This is the amount that the error E would increase given an incremental increase in w_{ij}, with everything else held constant.

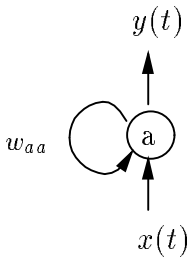


Figure 3.4: The simplest possible recurrent network

---

weight were set correctly and the activation function of the net shown were the usual sigmoidal squashing function, the network would be able to discriminate between arbitrary-length binary sequences that have at least one 1 in them, and those that do not.

Since this is a recurrent network, it cannot be trained using a straightforward application of the back-propagation algorithm, which assumes a feed-forward architecture. Given this, one approach to training such a network to recognize sequences is to "unfold the network over time" [Rumelhart *et al.*, 1986], as shown in Figure 3.5. Here, the recurrent network of Figure 3.4 has been transformed into a feed-forward network of depth $t+1$, where $t+1$ is the length of the input sequence, by replicating node a once for each step in the sequence. Thus, we have node a_0 corresponding to node a in the recurrent net at time step 0, node a_1 corresponding to node a at time step 1, and so on. Notice that each node a_i accepts input $x(i)$, which corresponds to input x at time i in the original recurrent network. Finally, note that in the feed-forward case, the weights on the links connecting the nodes corresponding to different time steps are all the same: w_{aa}. This is the counterpart of the recurrent link in the original network. This network will respond as the recurrent network would to a sequence of length t, but since it is feed-forward in nature, we can apply back-propagation to it, with the constraint that the links between nodes must change in unison. This constraint is easily enforced using the technique of *weight-sharing* [Rumelhart *et al.*, 1986; LeCun, 1989].

Back-propagation through time has the disadvantage that it consumes a good deal of memory by replicating network nodes. Furthermore, the resulting "unfolded" networks can be quite deep[5], particularly for long sequences. This is a disadvantage since back-propagation tends to be more successful at learning in relatively shallow networks. However, the unfolded version of the network is required only for training;

[5]I.e. they may have many layers

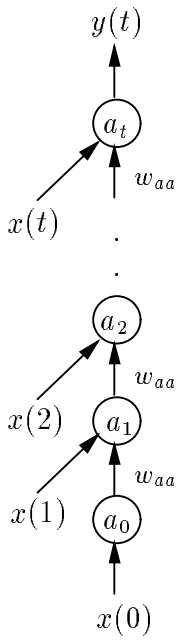


Figure 3.5: Back-propagation through time

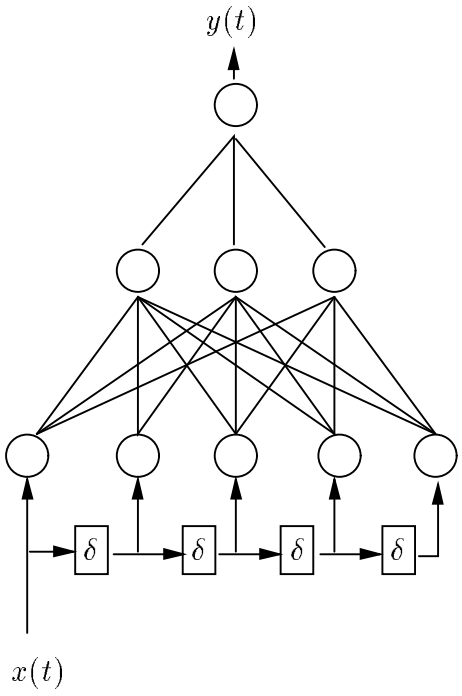


Figure 3.6: A time-delay neural network

---

once this has been completed, the weights learned may be transferred to a functionally equivalent, and much smaller, recurrent network.

3.4.2 Time-Delay Neural Networks

Another approach to the learning of sequences using back-propagation is the use of time-delay neural networks (TDNNs) [Waibel *et al.*, 1987; Waibel, 1989; Weigend *et al.*, 1990; Guyon *et al.*, 1991]. Figure 3.6 presents an example of a network of this sort. Here, the current input $x(t)$ is fed directly to the leftmost input node, while successive input nodes in the network receive the input signal after it has passed through a number of delays. Thus, at a given point in time, the input nodes at the bottom of the network will receive as input, from left to right, $x(t), x(t-\delta), x(t-2\delta), x(t-3\delta)$, and $x(t-4\delta)$. This is similar in spirit to the manner in which input at successive time steps is fed to successive layers under back-propagation through time, so that time is unfolded over space here as well, but there are important differences between the two schemes. Unlike back-propagation through time, the use of TDNNs does not allow one to collapse the resulting network down to an equivalent, smaller

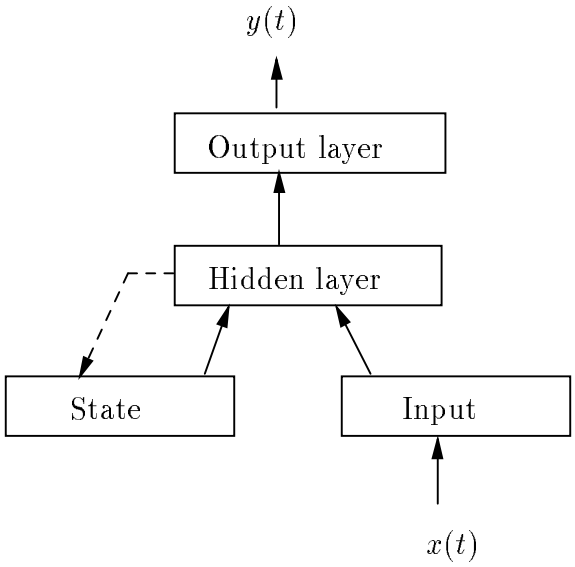


Figure 3.7: The Elman architecture

---

recurrent network, and the use of time-delays in this manner means that only a given time window of the sequence is provided as input to the network at any given time. However, time-delay neural networks are generally much shallower than networks which have been unfolded under back-propagation through time. Since learning under back-propagation tends to be more successful in shallow networks, this is an advantage that time-delay neural networks have over back-propagation through time.

3.4.3 Back-Propagation with State Units

A popular approach to connectionist sequence learning is the use of state units [Elman, 1988; Jordan, 1986]. This is perhaps best exemplified by the work of Elman; the network architecture suggested in [Elman, 1988] is presented in Figure 3.7. In this figure, each box represents a layer of units. Thus, this is a standard 2-layer perceptron, with one addition, namely, the state units. The solid arrows connecting layers in the network represent full connectivity between layers, while the dashed arrow from the hidden layer down to the state layer represents one-to-one connections, each of weight 1, such that the state layer at time t contains an exact copy of the contents of the hidden layer at time $t-1$.

The architecture is then trained using back-propagation, keeping the one-to-one downward connections fixed, and treating the contents of the state buffer as if they were external input to the network. The one-to-one downward links are thus essen-

tially ignored by the training algorithm. Technically, ignoring these links means that only an approximation to the true gradient $-\frac{\partial E}{\partial w_{ij}}$ is obtained, and that "gradient" descent is then performed using that approximation. This method yields convergence if the learning rate is relatively low, since the algorithm will never commit itself to a large step in the wrong direction. However, second-order methods,[6] which do take large steps, will be misled by the approximate gradient; these tend to perform poorly on Elman networks.

Despite these issues, networks of this sort have become a standard connectionist tool for the learning of sequences. They have in their favor simplicity, and the ability to learn sequences without unfolding input in time out over space, as both TDNNs and back propagation through time do. Thus they are somewhat less memory-expensive than these other two methods.

3.4.4 Discussion

The three methods outlined above are by no means the only ways in which back-propagation, or indeed the general idea of gradient descent, can be applied to the learning of temporal sequences. In fact, there exists a wide array of methods; the interested reader may find reviews in [Hertz *et al.*, 1991; Pearlmutter, 1990].

Nonetheless, these are three of the most commonly used connectionist approaches to sequence learning. In Chapter 6, we shall review how each of the three methods outlined here could be applied to the spatial concept learning task, and shall test a variation on the method of back propagation with state units, applied to that task. Eventually, however, we shall be adopting a sequence learning mechanism that is more highly structured, and more specifically geared for the movie learning task. The design of that mechanism, together with a discussion of its suitability as compared with these more traditional approaches, form a central focus of Chapter 6.

[6] I.e. methods based on the second derivative of the error with respect to the weight

Chapter 4

Learning Without Explicit Negative Instances

4.1 The Problem 45
4.2 A Solution: Mutual Exclusivity 47
4.3 Difficulties with Mutual Exclusivity 49
4.4 Salvaging Mutual Exclusivity 51
4.5 Implementation 51
4.6 Results 54
4.6.1 Uniform Attenuation 54
4.6.2 Incorporation of Prior Knowledge 62

Researchers in child language acquisition have often observed that the child learns language apparently without the benefit of explicit negative evidence [Braine, 1971; Bowerman, 1983; Pinker, 1989]. This introduces the following problem: if the child is never told that a particular utterance is incorrect, how does he or she learn not to utter it, while still learning to produce correct sentences that have also never been heard? How will he or she know which of these sentences that have never been heard are correct, and which are incorrect? In other words, how does the child know not to *overgeneralize* from the utterances heard, if nothing has been explicitly ruled out?

Pinker states the problem as follows:

> ... it is commonly assumed that children do not depend on negative evidence to acquire a language. ... [E]ither they never entertain any hypothesis that is a superset of the target language, or, if they do, some endogenous force must impel them to abandon it, because the world will never force them to. On the other hand, children cannot simply stick with the exact sentences they hear, because they must generalize to the infinite language of their community. This tension, between the need to generalize and the need not to generate supersets, characterizes many of the toughest problems in explaining human language acquisition.

While the abovementioned researchers have focused on the "no negative evidence"

problem as it relates to the acquisition of grammar, the problem is a general one, and appears in several different aspects of language acquisition. This chapter approaches the "no negative evidence" problem specifically in the context of learning the semantics of lexemes for spatial relations; recall from Chapter 1 that learning in the absence of explicit negative evidence is a part of the overall task specification for the work described in this thesis.

Although the focus here is on the very specific domain of spatial relations, it is worthwhile mentioning the fact that the problem of learning in the absence of negative evidence is a very general one, not at all restricted to this domain, or to the domain of language generally. And just as the problem is a general one, the methods of solving the problem which are presented in this chapter are general, and could well prove to be applicable in other domains.

4.1 The Problem

This section illustrates the "no negative evidence" problem as it arises in the learning of perceptually-grounded semantics for spatial terms.

To simplify exposition, only *static* terms are considered, i.e. terms that do not involve motion, and can thus be represented visually in a single frame. Furthermore, only *punctate* trajectors are used, i.e. trajectors consisting of a single point each. Techniques for handling non-punctate trajectors are covered in Chapter 5, while techniques for handling sequences of frames ("movies") are covered in Chapter 6. The integration of these techniques with the techniques developed in this chapter for learning in the absence of negative evidence is covered in §6.5.

Imagine a system which is presented with point trajectors located relative to non-punctate landmarks, as illustrated in Figure 4.1. Here, the placement of each of the three point trajectors (indicated by small dotted circles) constitutes a positive example of English *above*, relative to the tall landmark shown. The system is to learn to associate data such as this with natural language terms for spatial relations. Thus, once it has learned, the system should be able to indicate, for a point trajector at any location relative to a given landmark, how good an example of *above* the relation between that point trajector and the landmark is.

The "no negative evidence" problem arises here, as it is not clear just how to generalize from the positive examples seen. To help illustrate this point, Figure 4.2 shows three possible generalizations from the training data shown in Figure 4.1. (The size of each of the black circles indicates how good an example of *above* the relation between a point at that position and the landmark would be considered.) The very small region shown in (a) consists roughly of the convex hull of the training points.[1]

[1]The convex hull of a set X of points is the smallest set of points that is convex and contains X. Intuitively, if one were to stretch a rubber band around a set of points, the rubber band would

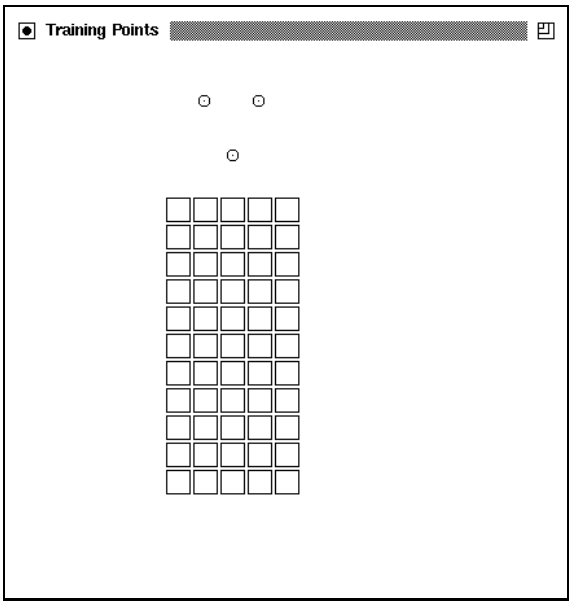
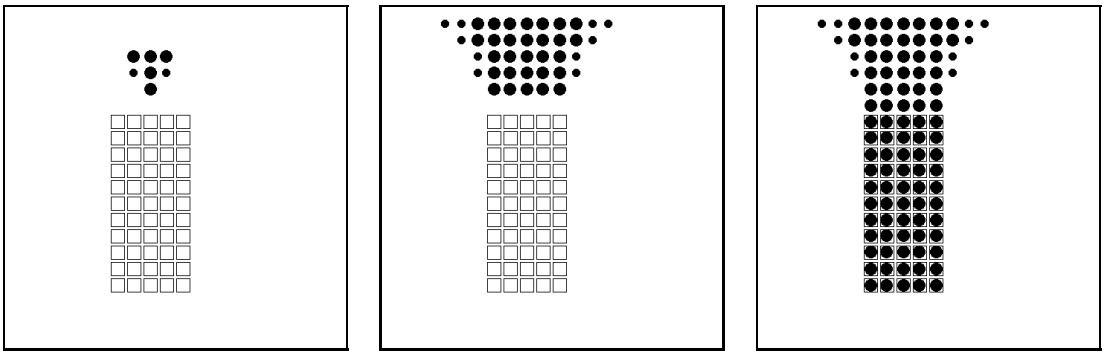


Figure 4.1: Three point trajectors *above* a landmark



(a) (b) (c)

Figure 4.2: Three possible generalizations from the training set for *above*

The region shown in (b), which generalizes further from the training data than the region in (a), corresponds fairly closely to our notion of *above*. Finally, yet another valid generalization from the training data is shown in (c), which generalizes still further. Thus, the three figures show undergeneralization, correct generalization, and overgeneralization, all of which are valid given the training data shown in Figure 4.1.

The essence of the problem is that without negative evidence, the system will have no way to know where to stop generalizing. Clearly, some generalization from the data itself is called for, as the correct answer (Figure 4.2(b)) does involve generalizing from the training data. However, since there is no negative evidence to explicitly rule out inclusion of the interior of the landmark in the region considered to be *above* the landmark, the system may well include that region as well, as in Figure 4.2(c).

And yet, despite this problem, humans do learn these concepts, apparently in the absence of negative evidence. The computational work presented in this chapter indicates how that learning might take place.

The basic ideas are presented first, and are followed by technical details. After that, the results of using the techniques put forth here on the task of learning the semantics for spatial terms are presented, and discussed.

4.2 A Solution: Mutual Exclusivity

One solution to the "no negative evidence" problem which suggests itself is to take every *positive* instance for one concept to be an *implicit negative* instance for all other spatial concepts being learned. There are problems with this approach, as we shall see, but they are surmountable.

Figure 4.3 illustrates this idea by showing how the principle might be applied to a learning system of the sort presented in this thesis. The input scene is shown at the bottom; this particular scene has been classified by the teacher as a positive instance of English *above*. Therefore, in this one scene, we have an explicit positive instance of *above* and implicit negative instances for all other terms being learned. This is indicated at the output nodes at the top of the figure.

There are related ideas present in the child language literature, which support the work presented here. [Markman, 1987] posits a "principle of mutual exclusivity" for object naming, whereby a child assumes that each object may only have one name. This is to be viewed more as a learning strategy than as a hard-and-fast rule: clearly, a given object may have many names (an office chair, a chair, a piece of furniture, etc.). The method being suggested really amounts to a principle of mutual exclusivity for spatial relation terms: since each spatial relation can only have one name, we take a positive instance of one to be an implicit negative instance for all others. Clearly, both the problem and the posited heuristic solution are very general in nature, and could

provide the outline of the convex hull of the set of points.

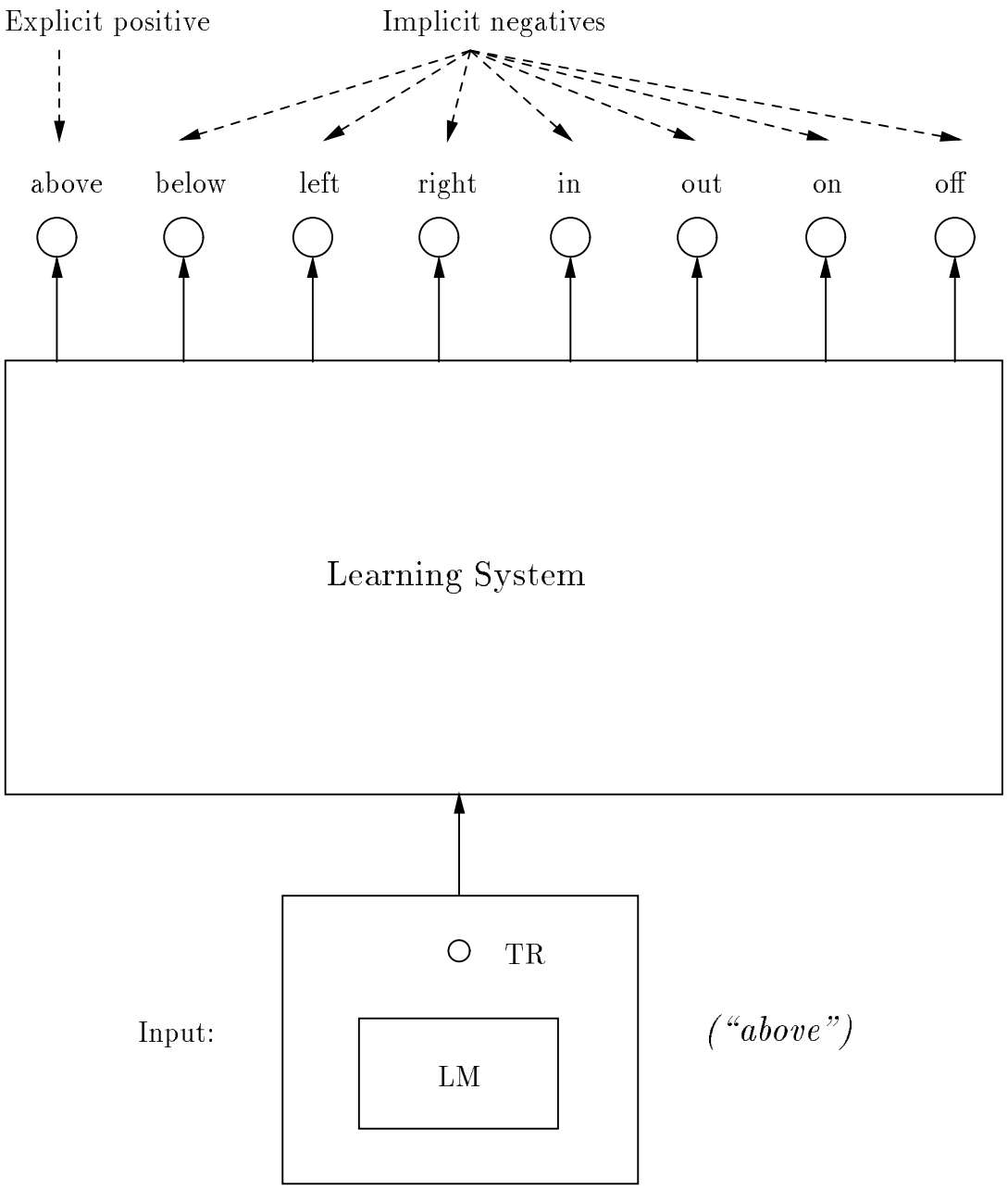


Figure 4.3: The Principle of Mutual Exclusivity

be applied to domains other than the naming of either objects or spatial relations.

In a related vein, [Johnston and Slobin, 1979] point out that in a study of children learning locative terms in English, Italian, Serbo-Croatian, and Turkish, terms were learned more quickly when there was little or no synonymy among terms. They note that children seem to prefer a one-to-one meaning-to-morpheme mapping; this is similar to, although not quite the same as, the mutual exclusivity notion put forth here.[2] Johnston and Slobin's principle of one-to-one meaning-to-morpheme mapping also appears as the principle of contrast in [Clark, 1987] and the principle of uniqueness in [Pinker, 1989]; these are thus also close relatives of Markman's principle of mutual exclusivity. Similar ideas can also be found in [Bolinger, 1965; Chomsky and Lasnik, 1977; MacWhinney, 1989].

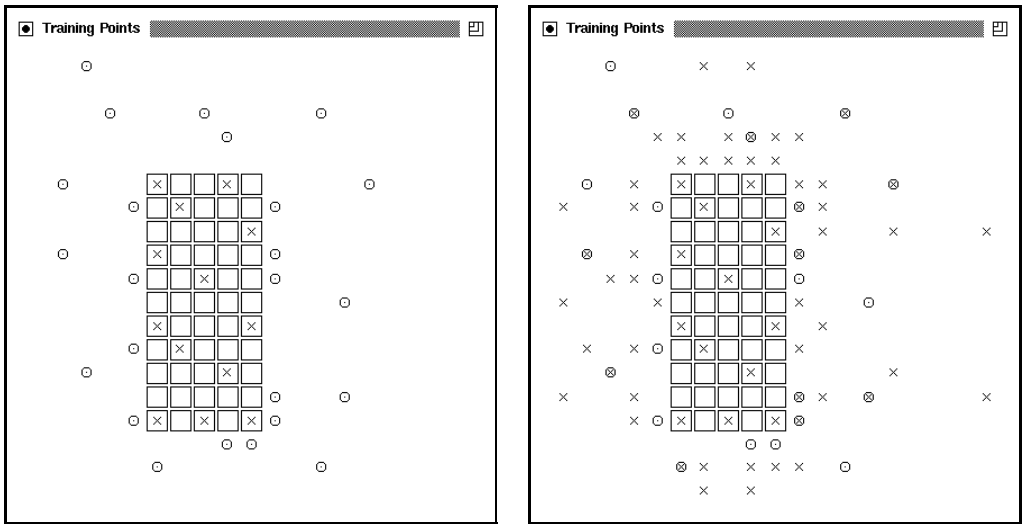
4.3 Difficulties with Mutual Exclusivity

For purposes of exposition, this chapter focuses on the learning of the eight English spatial terms listed below. Note that all eight are static, i.e. they do not involve motion. These correspond to the terms which label the output nodes in Figure 4.3.

- *above*
- *below*
- *to the left of*
- *to the right of*
- *inside*
- *outside*
- *on*
- *off*

Bearing this set of terms in mind, consider Figure 4.1 again. Under mutual exclusivity, if the three dotted circles are considered as positive instances of *above*, they are implicit negatives for all other spatial terms. It is certainly the case that a TR cannot be both *above* and *in* a LM; thus, it is reasonable for a positive instance of *above* to yield an implicit negative for *in* (and for *below*, among other terms). However, a TR can be both *above* and *outside* a LM, or both *above* and *off* a LM. As an example,

[2] They are not the same since a difference in meaning need not correspond to a difference in actual reference. When we call a given object both a "chair" and a "throne", these are different meanings, and this would thus be consistent with a one-to-one meaning-to-morpheme mapping. It would not be consistent with the principle of mutual exclusivity, however.



(a) (b)

Figure 4.4: Ideal and realistic training sets for *outside* (see text)

all three positive instance for *above* shown in the figure could also be perfectly good positive instances of *outside*, and labeling them as implicit negatives through mutual exclusivity leaves us with a training set that has many *false negatives* in it, i.e. implicit negative instances which really should be positives.

> *More generally, any time the principle of mutual exclusivity is applied to a domain in which there is overlap among concepts, the problem of false implicit negatives will arise.*

Outside is a term that is particularly badly affected by this problem of false implicit negatives: all of the spatial terms listed above except for *in* (and *outside* itself, of course) will supply false negatives to the training set for *outside*.

The severity of this problem is illustrated in Figure 4.4. In the two figures shown, which represent training data for the spatial concept *outside*, we have tall, rectangular landmarks, and training points[3] relative to the landmarks. Positive training points (instances) are marked with circles, while negative instances are marked with X's. In both (a) and (b), the positive instances were supplied by a teacher. In (a), the negative instances were placed there by the teacher, showing exactly where the region *not* outside the landmark is. This gives us a "clean" training set, but the use of teacher-supplied explicit negative instances is precisely what we are trying to get away from. In (b), the negative instances shown were derived from positive instances

[3]I.e. trajectors consisting of a single point each

for the other spatial terms listed above, through the principle of mutual exclusivity. Thus, this is the sort of training data the system will actually be exposed to. Note that in (b) there are many false negative instances among the positives, to say nothing of the positions which have been marked as both positive and negative (these are the positions which have been marked with both a circle and an X).

This issue of false implicit negatives is the main problem with the principle of mutual exclusivity. It is this problem that the following section addresses.

4.4 Salvaging Mutual Exclusivity

The basic idea used here, in salvaging the idea of mutual exclusivity, is to treat positive instances and implicit negative instances differently during training:

Implicit negatives are viewed as supplying only *weak* negative evidence.

The intuition behind this is as follows: since the implicit negatives are arrived at through the application of a fallible heuristic rule (mutual exclusivity), they should count for less than the positive instances, which are all assumed to be correct. Clearly, the implicit negatives should not be seen as supplying excessively weak negative evidence, or we revert to the original problem of learning in the (virtual) absence of negative instances. But equally clearly, the training set noise supplied by false negatives is quite severe, as seen in the figure above. So this approach is to be seen as a compromise, so that we can use implicit negative evidence without being overwhelmed by the noise it introduces in the training sets for the various spatial concepts.

The details of this method, and its implementation under back-propagation, are covered in the following section. Note, however, that this is a very general solution to the "no negative evidence" problem, and can be understood independently of the actual implementation details. Any learning method which allows for weakening of evidence should be able to make use of it (although non-evidential symbolic systems cannot). In addition, it could serve as a means for addressing the "no negative evidence" problem in other domains. For example, a method analogous to the one suggested here could be used for object naming, the domain for which Markman suggested mutual exclusivity.

4.5 Implementation

The actual architecture used in the experiments reported in this chapter (the one portrayed as a box labeled "Learning System" in Figure 4.3) is a simplified version of the architecture for learning static terms which will be presented fully in Chapter 5. For our purposes here, we need not consider its internal structure in detail; it suffices

to note simply that it accepts input in the form of training points (i.e. point trajectors) relative to landmarks, as shown previously, that it must classify its input into one or more of the spatial categories which label the output nodes shown at the top of Figure 4.3, and finally, that it is trained under error back-propagation [Rumelhart *et al.*, 1986].

At attempt has been made to make this presentation as general as possible, so that the ideas involved may be applied to other domains.

We present here the means by which the intuitive notion of "weakening of evidence from implicit negative instances" is implemented. It is assumed that training sets have been constructed using mutual exclusivity as a guiding principle, such that each negative instance in the training set for a given concept results from a positive instance for some other concept.

- Evidence from implicit negative instances is weakened simply by attenuating the error caused by these implicit negatives.
- Thus, an implicit negative instance which yields an error of a given magnitude will contribute less to the weight changes in the network than will a positive instance of the same error magnitude.

This is done as follows:

Referring back to Figure 4.3, note that output nodes have been allocated for each of the concepts to be learned. For a network such as this, the usual error term in back-propagation is

$$E = \frac{1}{2}\sum_{i,p}(t_{i,p} - o_{i,p})^2 \tag{4.1}$$

where i indexes over output nodes, and p indexes over input patterns. Here, $t_{i,p}$ is the desired output, and $o_{i,p}$ is the actual output, for node i on pattern p.

We modify this by multiplying the error at each output node by a value $\beta_{i,p}$, dependent on both the node and the current input pattern. In general, $\beta_{i,p}$ corresponds to the amount by which the error signal from node i on pattern p is to be attenuated.

$$E = \frac{1}{2}\sum_{i,p}((t_{i,p} - o_{i,p}) \times \beta_{i,p})^2 \tag{4.2}$$

For positive instances, $\beta_{i,p}$ will be 1.0, so that the error is not attenuated. For an implicit negative instance of a concept, however, $\beta_{i,p} < 1.0$, such that error signals from implicit negatives are attenuated.

For example, if we are currently viewing input pattern p, which is a positive instance of concept j, then the target value for output node j will be 1.0, while the target values for all others will be 0.0, as they are implicit negatives. In addition, $\beta_{j,p} = 1.0$, and $\beta_{i,p} < 1.0, \forall i \neq j$.

The question of how to set the $\beta_{j,p}$ values in the case of implicit negatives now arises. Two approaches to this which have been investigated are:

- **Uniform attenuation:** If j is the current positive concept, set all $\beta_{i,p}$ where $i \neq j$ to the same value, B. This value B will then be an external parameter that can be set prior to training, much the way learning rate, momentum, and others usually are.

- **Incorporation of prior knowledge:** If there is prior knowledge regarding which concepts are disjoint, this can be built in and used in assisting the learning. This is easily done by setting $\beta_{i,p} = 1.0$ for all concepts which are completely disjoint with the current positive concept j in pattern p. Thus, these concepts will receive strong, rather than attenuated, negative evidence from positive instances of concept j. Analogous treatment is easily arranged if one concept is known to include another. This approach can of course be used in conjunction with the one above, such that certain pairs of concepts will have their β values set according to prior knowledge regarding their distribution relative to one another, while the remainder will take $\beta_{i,p}$ where $i \neq j$ to be the uniform value B, as a default.

An appealing idea which has not been fully investigated is that of adapting the values $\beta_{i,p}$ during learning. Note that this is a delicate business: it would be possible to learn the β values under back-propagation, but simply following the gradient given by $-\frac{\partial E}{\partial \beta_{i,p}}$ (following Equation 4.2) will tend toward a solution in which $\beta_{i,p} = 0.0, \forall i \neq j$, where j is the positive concept in pattern p.[4] This essentially takes us back to the original situation, in which we had no negative evidence whatsoever (here, we have negative evidence, but it is being given no weight at all). Thus, this will tend to yield overgeneralization from the positive examples. A possible fix is to add a term in the cost function which penalizes trivial solutions of this sort.

Another approach is to set all $\beta_{i,p}$ where $i \neq j$ to a uniform value B, as above, have the system learn the concepts to the best of its ability, and then use observed output correlations between concepts to estimate what the actual overlap in concepts is, and thus to estimate what the optimal β values would be. These estimates would then be used in further training, to "clean up" the initial results obtained with uniform attenuation. As learning progresses, β estimates closer and closer to the ideal would be obtained from inter-concept output correlations, and used in further learning.

[4] This assumes that $\beta_{j,p}$ is fixed at 1.0, so that error from positive evidence is never attenuated. It is simple to show that following the gradient tends toward $\beta_{i,p} = 0$: $-\frac{\partial E}{\partial \beta_{i,p}} = -\beta_{i,p}(t_{i,p} - o_{i,p})^2$. Since $0 < (t_{i,p} - o_{i,p})^2 < 1$, following the gradient will always result in taking a step of size smaller than $\beta_{i,p}$ in the negative direction if $\beta_{i,p}$ is positive, and in the positive direction if $\beta_{i,p}$ is negative. This will cause $\beta_{i,p}$ to approach 0.

4.6 Results

The system was trained, using the techniques described in this chapter, on the eight English spatial terms mentioned above: *above, below, to the left of, to the right of, inside, outside, on,* and *off*. Figure 4.5 and Figure 4.6 show positive training instances for the eight concepts being learned, relative to a vertically extended landmark, while Figure 4.7 and Figure 4.8 show positive instances for these concepts relative to a horizontally extended landmark; these constitute the positive training data used for the concepts. The principle of mutual exclusivity was used to obtain implicit negative instances in addition to the explicit positive instances supplied by the training sets, and the system was trained on all concepts in parallel.

4.6.1 Uniform Attenuation

The first method used was uniform attenuation of error signals from all implicit negative instances. That is, we set all $\beta_{i,p} = B$, for some B, where i is not the concept of which pattern p is a positive instance.

Figure 4.9 shows the results of learning the spatial term *outside*, under three different conditions. (We focus on *outside* because of its particular susceptibility to the problem of false implicit negatives under mutual exclusivity; recall Figure 4.4(b).) The size of the black circles indicates the appropriateness, as judged by the trained network, of using the term *outside* to refer to a particular position, relative to the landmark shown. Figure 4.9(a) shows the results of learning without any negative instances whatsoever. As one might expect, this yields gross overgeneralization from the positive instances, such that all points inside as well as outside the LM are considered to be *outside*. Figure 4.9(b) shows the results of learning with implicit negatives, obtained through the mutual exclusivity heuristic, but without weakening the evidence from these negatives – i.e. without attenuating the error signal resulting from them. Clearly, the concept is learned very poorly, as the noise from the false implicit negatives hinders the learning of the concept.[5]

Finally, Figure 4.9(c) shows the results of learning with implicit negative instances, obtained through mutual exclusivity, such that the evidence from the negatives is weakened.[6] The concept *outside* is learned far more accurately in this case than in the other two cases, demonstrating the utility of the techniques used here. Having implicit negatives supply only weak negative evidence greatly alleviates the problem of false implicit negatives in the training set, enabling us to learn without using explicit, teacher-supplied negative instances.

[5]It should be noted that in general, when using mutual exclusivity without weakening the evidence given by implicit negatives, the results are not always identical with those shown in Figure 4.9(b), but are always of approximately the same quality.

[6]I.e. the error is attenuated following Equation 4.2. The uniform attenuation value $B = 0.03$ was used in the experiment reported here.

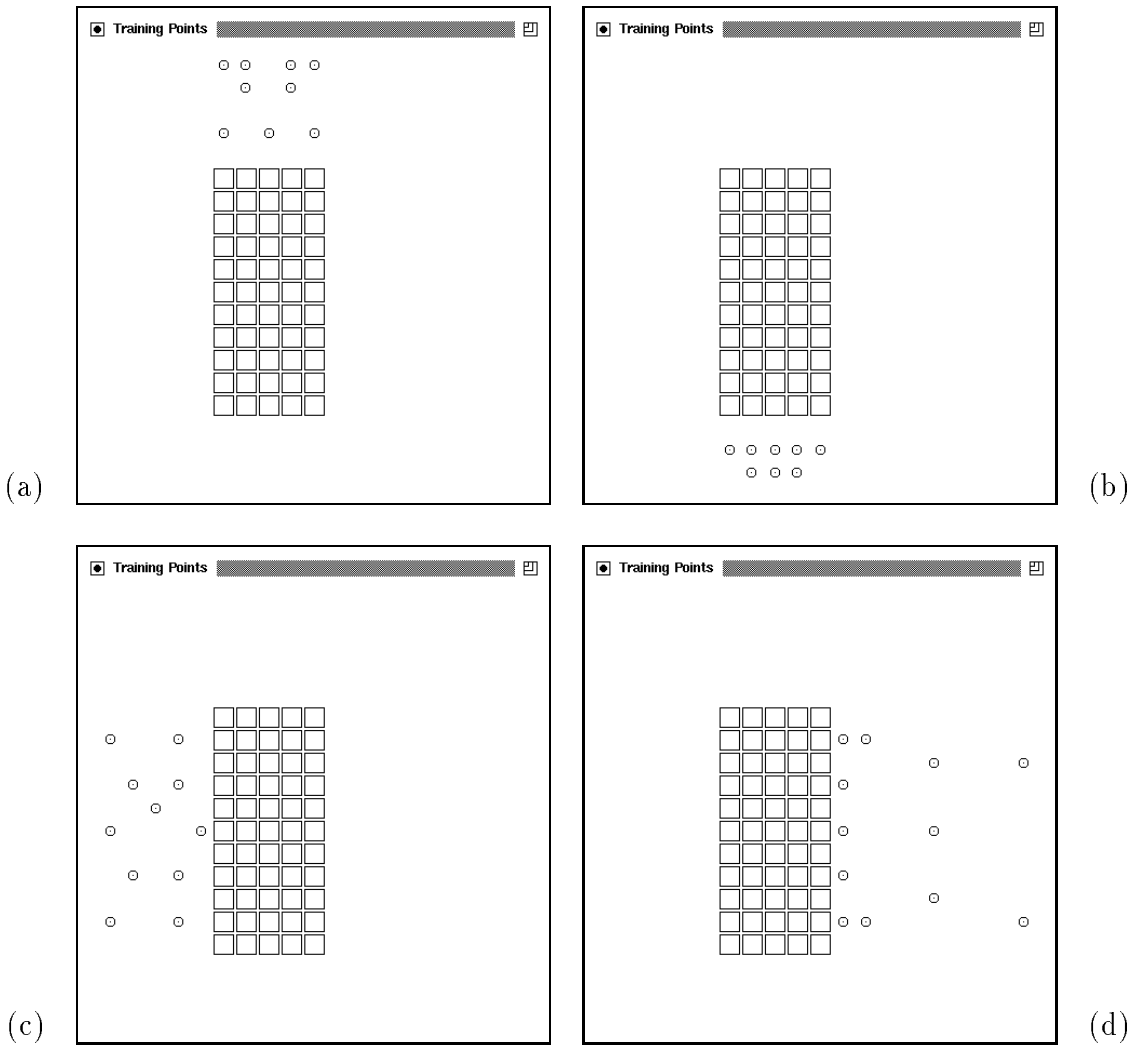


Figure 4.5: Positive instances for *above, below, to the left of,* and *to the right of* relative to a vertically extended landmark

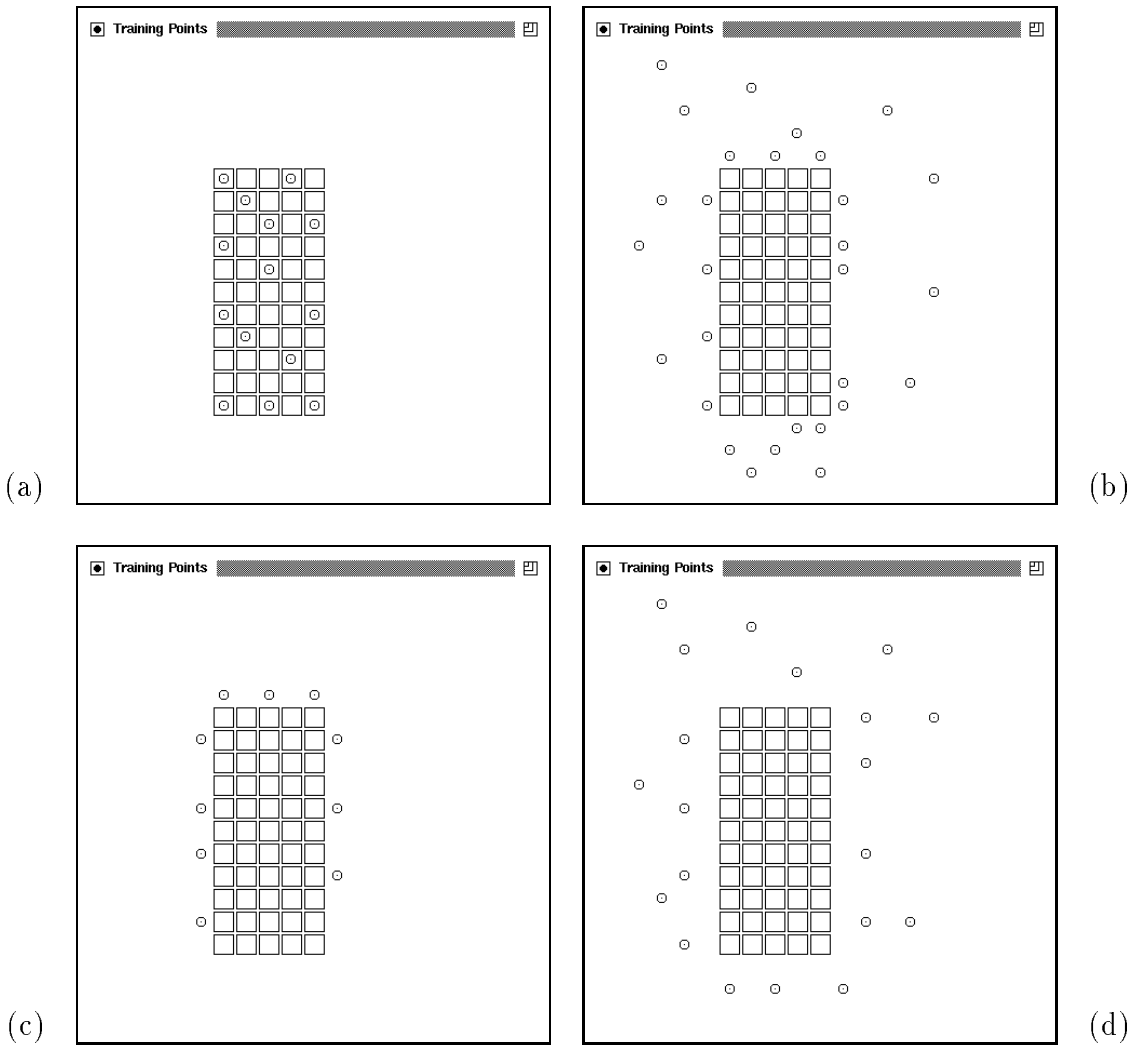


Figure 4.6: Positive instances for *inside, outside, on,* and *off* relative to a vertically extended landmark

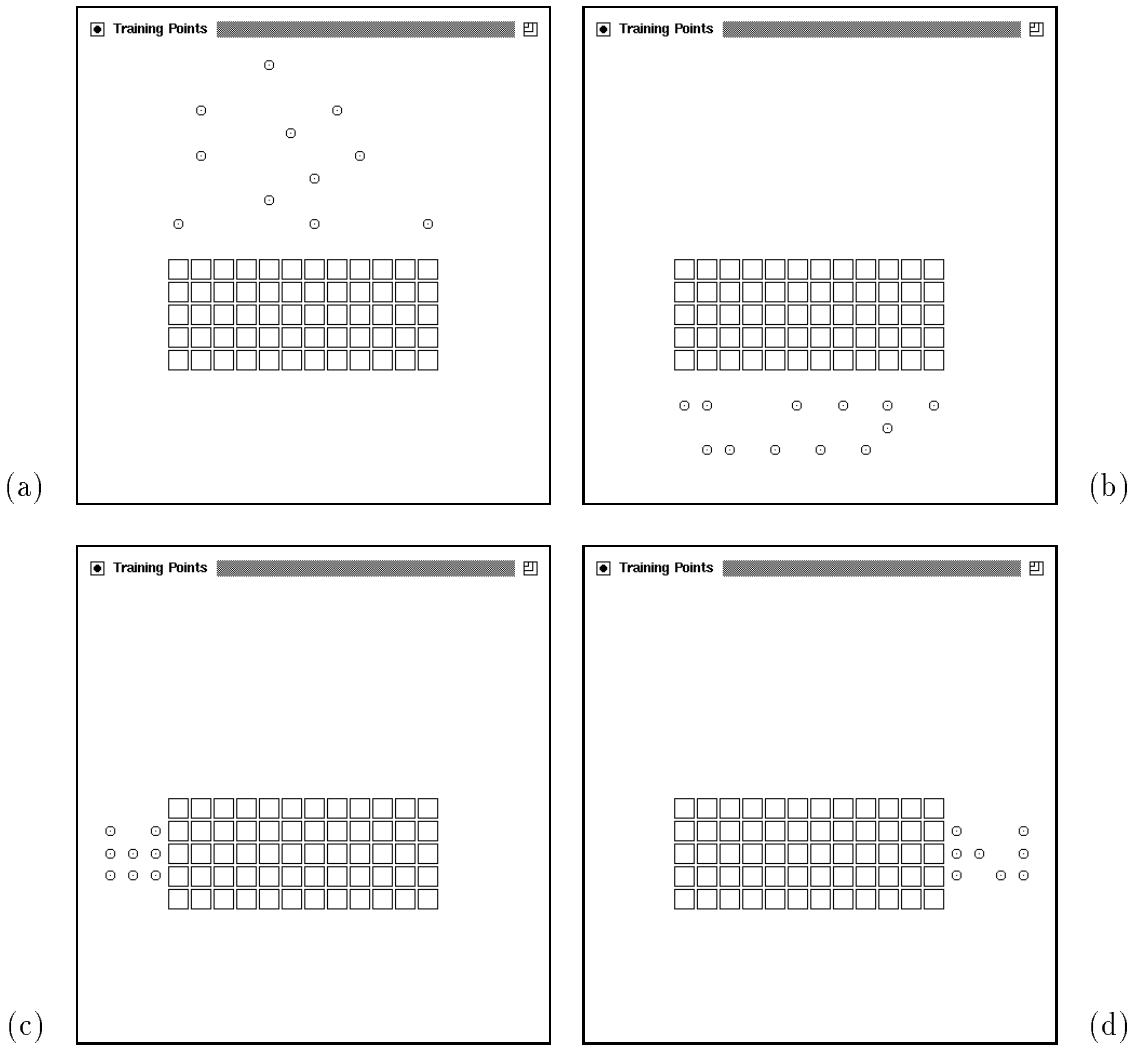


Figure 4.7: Positive instances for *above, below, to the left of,* and *to the right of* relative to a horizontally extended landmark

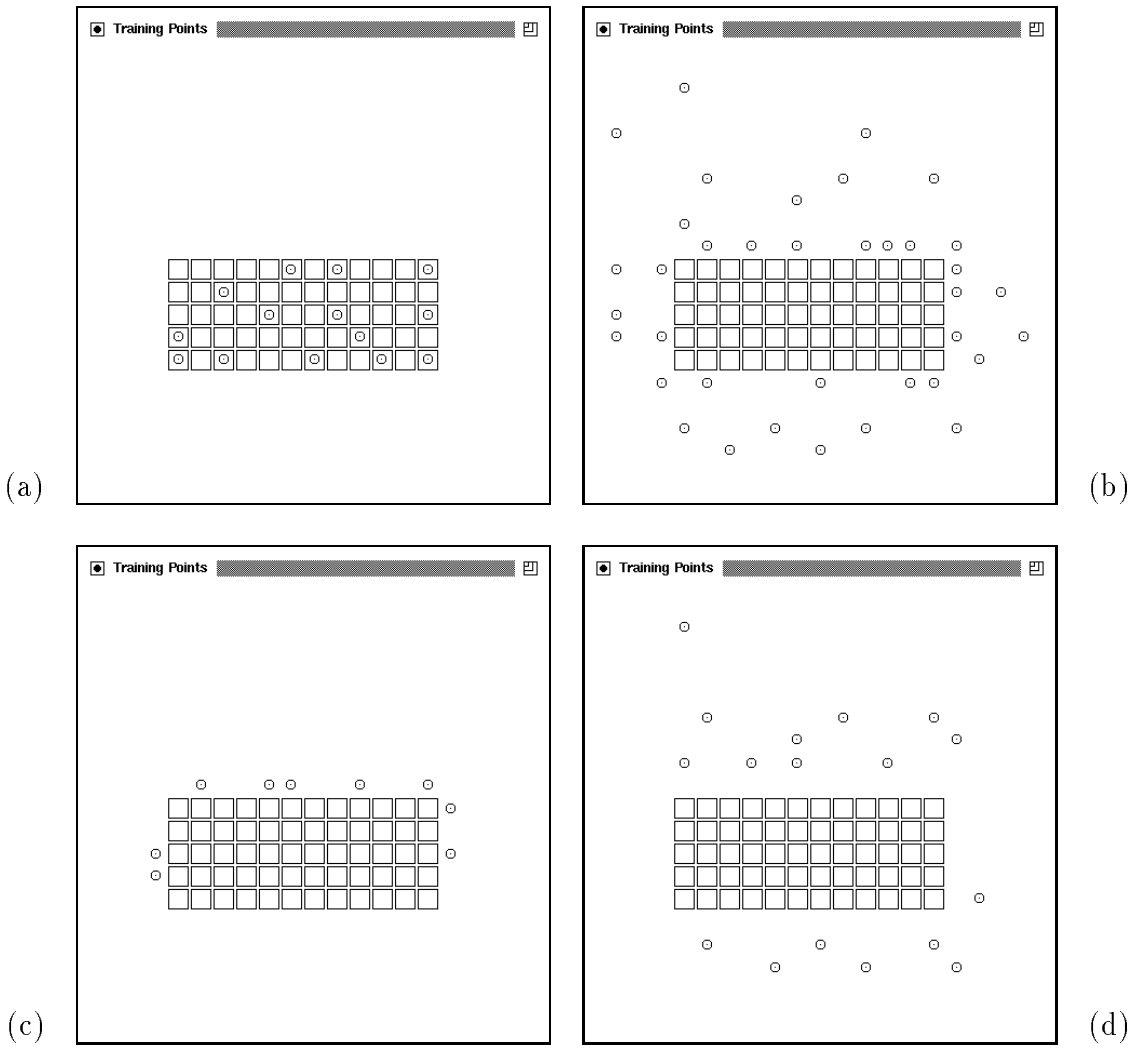
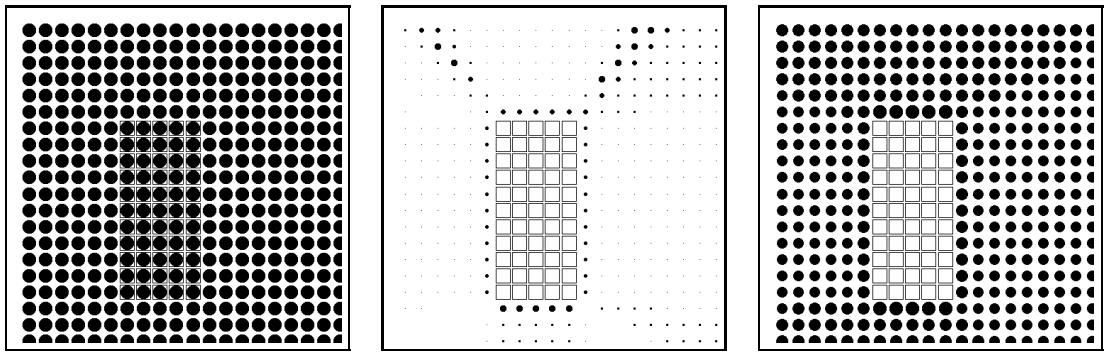


Figure 4.8: Positive instances for *inside, outside, on,* and *off* relative to a horizontally extended landmark



(a) (b) (c)

Figure 4.9: *Outside* learned without negatives, and with strong and weak implicit negatives

---

Figure 4.10 and Figure 4.11 show the results of learning for all eight concepts in this experiment.

Experiments were run varying the value of B, with a learning rate of 1.0 and momentum of 0.0. For each value of B, the learning was allowed to proceed until the error on an accompanying test set, which included both positive and negative instances of each concept, fell to 0.99.[7] The fact that the total summed squared error on this test set, which contained 593 positive and negative instances in total, reached as low as 0.99 (and indeed as low as 0.86 for $B = 0.03$) indicates that the concepts were learned accurately. Note that there was no attenuation of error from negative instances in the test set, only in the training set. Thus, this is evidence that the system neither undergeneralized nor overgeneralized. This can of course also be seen from Figure 4.10 and Figure 4.11.

Unfortunately, the success of this scheme is sensitive to the particular value chosen for B. Figure 4.12 plots the number of epochs required for the test set error to reach 0.99 against the value chosen for B.[8] For all values of B greater than 0.035, the learning never converged. This fact and the diagram, taken together, indicate that only a small range of values for B will yield convergence in fairly short order.

Thus, while it would be possible to empirically try to find an acceptable value for B for each training set one wanted to learn, a far more reasonable approach is to use this technique in conjunction with others, which may alleviate this problem of

[7] This technique of using the error on an accompanying test set as a criterion for stopping training is known as *cross-validation* [Morgan and Bourlard, 1989; Weigend *et al.*, 1990]. The idea is that error on a test set will tend to give a better indication of the capability of the trained network to generalize than will error on the training set, since the network has never seen the actual data in the test set.

[8] Note that the x-axis scale for B is logarithmic, and runs from 0.005 to 0.5.

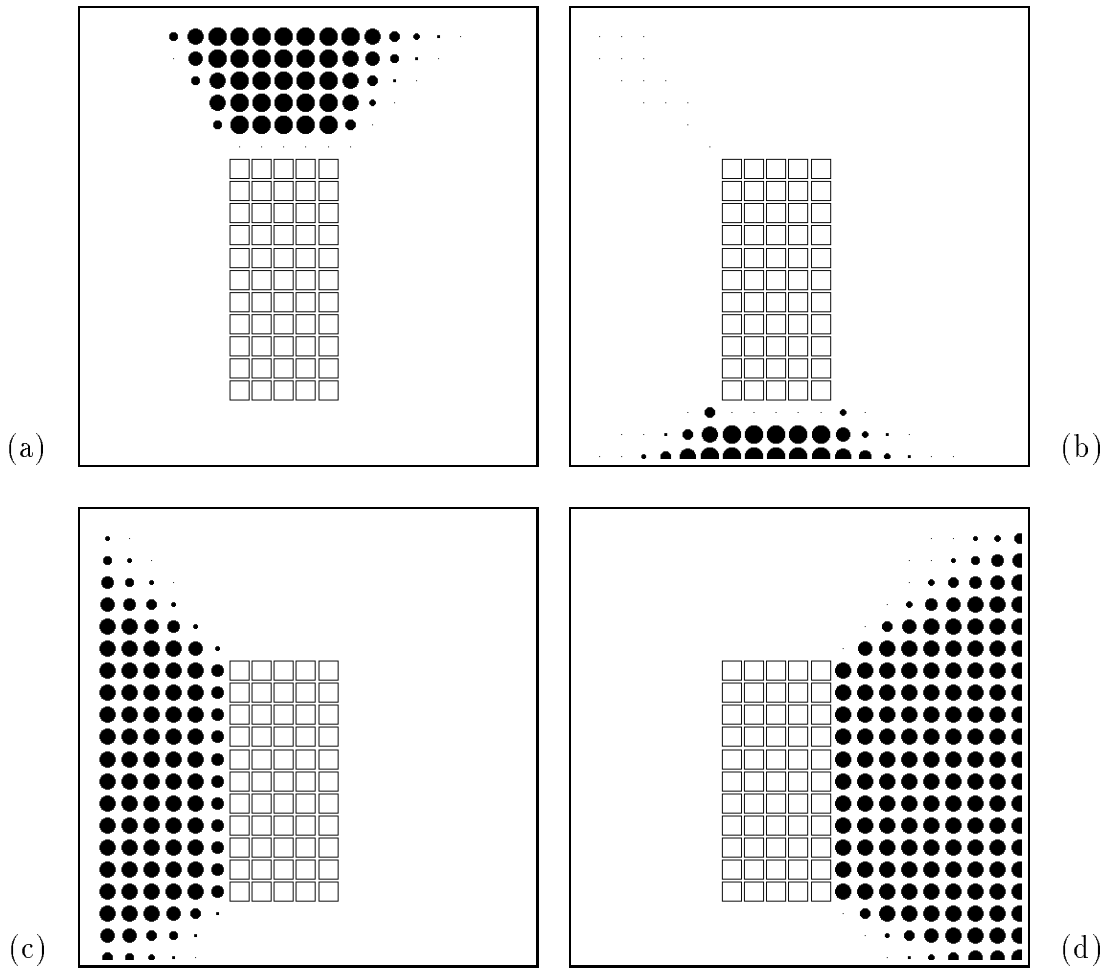


Figure 4.10: *above, below, to the left of,* and *to the right of* learned with weakened implicit negatives

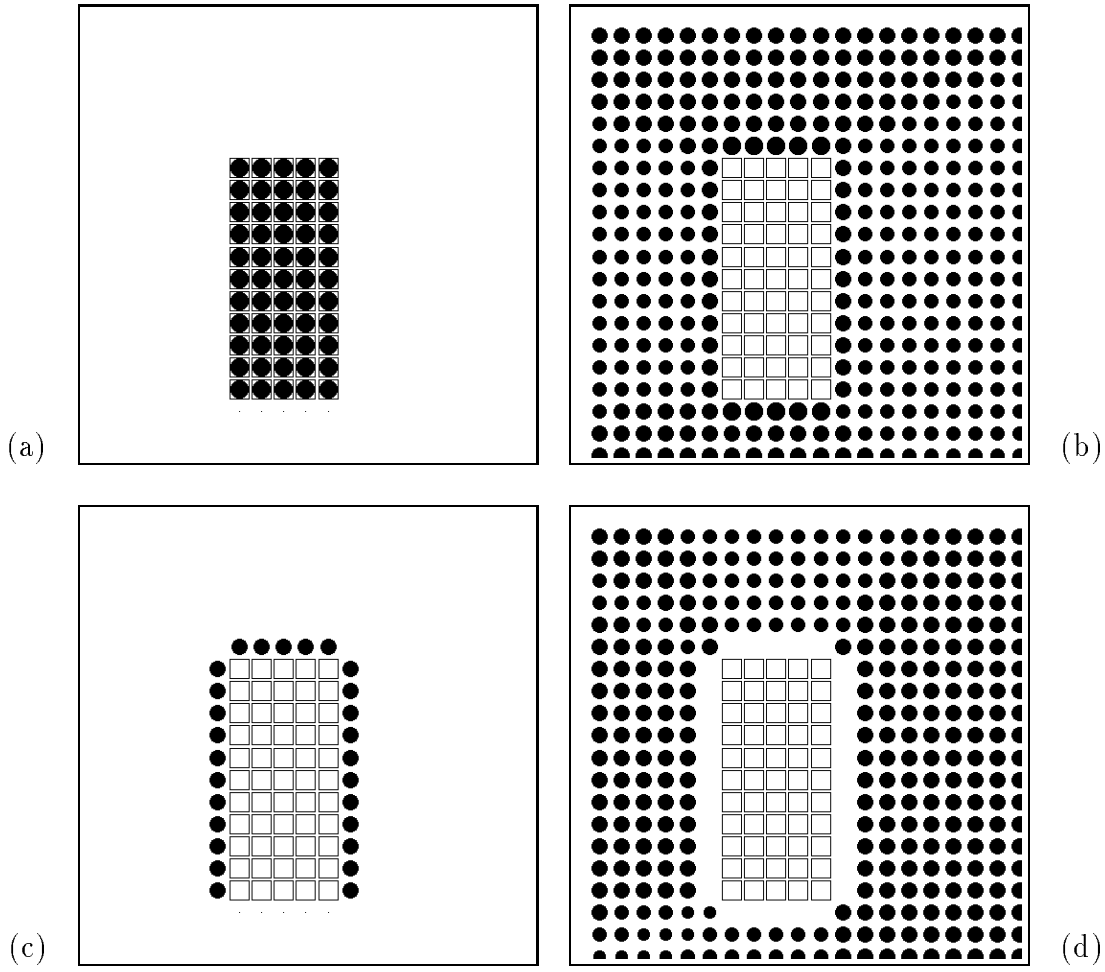


Figure 4.11: *inside, outside, on,* and *off* learned with weakened implicit negatives

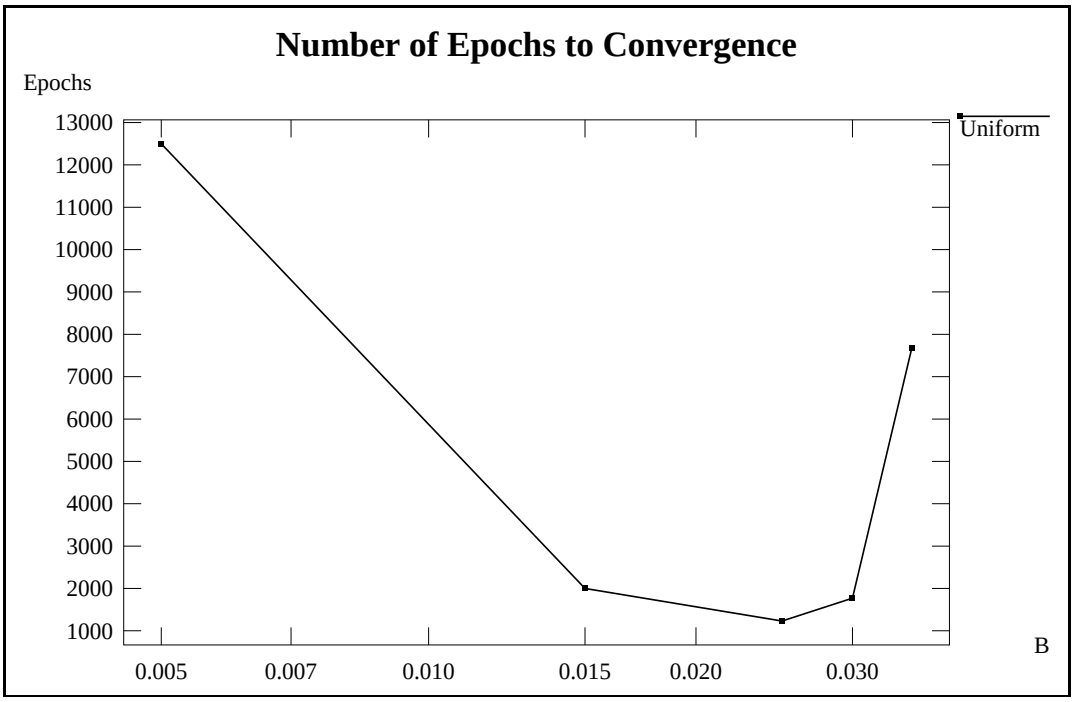


Figure 4.12: Epochs to convergence as a function of B

sensitivity to the specific B value chosen. Two possibilities are the incorporation into the $\beta_{i,p}$ values of prior knowledge regarding the domain, and some means of adapting the $\beta_{i,p}$ values during learning, as discussed above. Of these two, only the former has been investigated in any detail.

4.6.2 Incorporation of Prior Knowledge

If the learning system had access, prior to learning, to knowledge that certain concepts were disjoint, this knowledge could be brought to bear on the learning process. For example, if we knew that *above* and *below* were disjoint, then we could set $\beta_{above,p} = 1.0$ for all patterns p which were positive instances of *below*, and $\beta_{below,p} = 1.0$ for all patterns p which were positive instances of *above*. All remaining $\beta_{i,p}$ values would be set to the uniform attenuation value, B. Intuitively, this means taking implicit negative evidence seriously, provided it comes from a source we know to be reliable, and otherwise, taking it less seriously.

One possible source of knowledge regarding which concepts might be disjoint comes from *antonyms*. It is at least a possibility that the pairing of antonyms is primarily lexical in nature, that is, that it is the terms themselves that are paired,

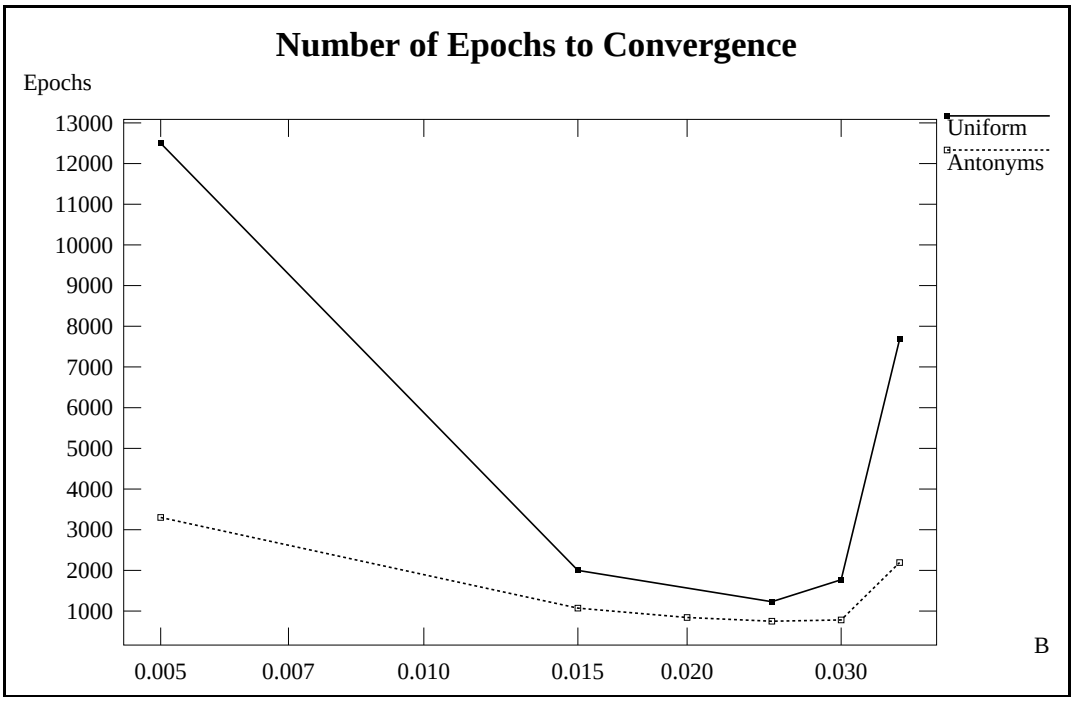


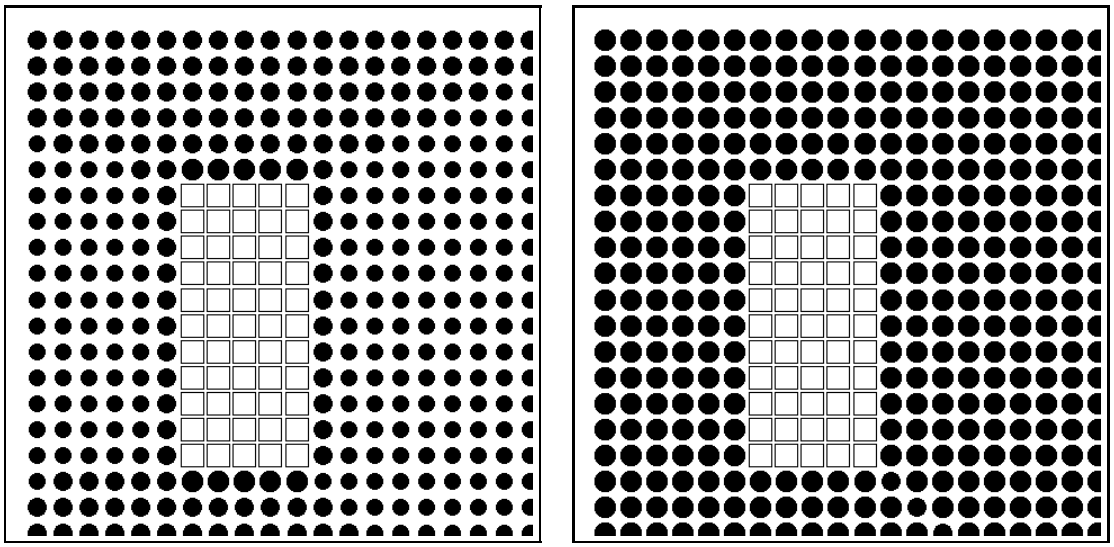
Figure 4.13: Epochs to convergence as a function of B

rather than their meanings.[9] For instance, if we knew that the terms *above* and *below* were opposites, without knowing at all what they actually meant, this knowledge could be used as outlined above, in taking implicit negative evidence from *above* seriously when learning *below*, and vice versa.

A set of experiments was run in which this technique was used. Specifically, *above* and *below* were treated as antonyms, as were the pairs *left / right*, *in / out*, and *on / off*. The results are displayed in Figure 4.13. Here we see the dependence of convergence time on B in two cases. The first is the case of uniform attenuation, as described above. Thus, the solid line in Figure 4.13 is identical to the one in Figure 4.12. The second is the case of no attenuation in the case of antonym pairs, and uniform attenuation otherwise; this is the dotted line in the figure.

It is clear that the incorporation into the system of knowledge of which terms are antonyms assists the learning, helping the system reach a test set error of 0.99 in a

[9]For suggestive evidence supporting this notion, note that the opposite of *above* is *below* and not *under*, even though *under* is roughly synonymous with *below* (at least if we concentrate on the central senses of the terms). In addition, [Tomasello, 1987] presents psycholinguistic data in which English prepositions which are members of antonym pairs (e.g. *in*, *out*, *over*, *under*) are learned earlier than other prepositions which are not (e.g. *by*, *at*). Since knowledge of which concepts are antonyms assists learning, as we shall see, this psycholinguistic result is consistent with the hypothesis that it is the terms, rather than their (learned) meanings, that are paired.



(a) (b)

Figure 4.14: *Outside* learned without, then with, knowledge of antonyms

shorter amount of time than is required without this built-in knowledge. In addition, the system is less sensitive to the exact value of B chosen. This can be seen from the gentler slope in the case of antonym knowledge.

The learning is somewhat more accurate, as well. Figure 4.14 presents the concept *outside*, learned first without knowledge of antonym pairings (Figure 4.14(a) is thus identical to Figure 4.9(c)), then with this knowledge. As can be seen from the figures, the concept is more accurately learned in the latter case (the black circles are larger in the area outside the landmark in (b) than in (a), indicating a stronger response to a point trajector at that location). This fact is also reflected in the minimum total summed squared error over the test set obtained in the two cases: using knowledge of antonym pairings, the minimum test error obtained was 0.37 over 593 test instances, as compared with 0.86 without this knowledge.

Thus, the incorporation into the learning system of prior knowledge regarding the distribution of concepts relative to one another can serve to lessen the problem of sensitivity to the particular attenuation value B chosen, and can yield more accurate learning of the concepts overall.

More generally, we have seen in this chapter that the technique of mutual exclusivity using weak implicit negatives provides a means for learning in the absence of explicit negative evidence. While this chapter has focused solely on the case of static point trajectors, we will in the next two chapters broaden our scope to encompass both nonpunctate trajectors and motion. In §6.5, we shall be using the techniques de-

scribed in this chapter to learn visually-grounded lexical semantics involving moving, nonpunctate trajectors.

In Chapter 1, we noted that one of the central themes of the thesis as a whole is the fact that a computational modeling effort directed at a particular line of linguistic inquiry can, in the modeling process, produce techniques of general applicability, as could modeling efforts in other domains. The method presented in this chapter, the use of weak implicit negative instances arrived at through the principle of mutual exclusivity, provides an example of such a general technique springing from a linguistic modeling effort.

Chapter 5

Static Scenes

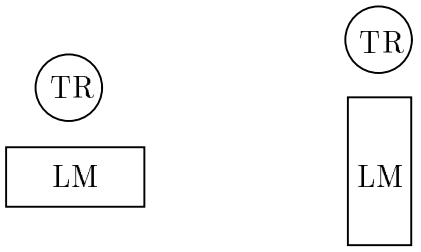
5.1 The Problem 66
5.2 Outline of a Solution 69
5.2.1 Directional Features 69
5.2.2 Non-directional Features 70
5.3 Implementing the Solution 74
5.3.1 Preliminaries 76
5.3.2 Directional Features 78
5.3.3 Non-directional Features 81
5.3.4 The Architecture in Review 88
5.3.5 Motivations from Neuroscience 89
5.4 Results 91
5.4.1 Prototype Effects 94
5.4.2 Cross-Linguistic Variation 95

This chapter describes a system which accepts static scenes as input, and learns to associate scenes with spatial terms from some natural language. This is an intermediate system, one which was later incorporated into the overall architecture described in Chapter 6.

The chapter begins by reiterating some of the challenges involved in constructing a system capable of learning the semantics for spatial terms from any natural language, proceeds to outline a solution to the problem in very general terms, and then presents the technical details of the solution. Finally, the results of running the resulting system on training sets for terms from a number of languages are presented.

5.1 The Problem

Recall from Chapter 2 that languages differ, sometimes dramatically, in their structurings of space. To return to a simple but effective example of this, consider Figure 5.1. This figure contains two scenes, both of which are properly described by English *above*, but which are described by different words in Mixtec. The challenge addressed in this chapter is that of designing and constructing a system that will be able to



English *above* English *above*
Mixtec *siki* Mixtec *šini*

Figure 5.1: Cross-linguistic variation in spatial systems

---

learn a significant part of the spatial system of an arbitrary natural language. Since different languages are sensitive to different perceptual features in scenes, the system must be able to adapt itself to pick up on whatever features are relevant for terms in the language currently being learned. For example, Mixtec appears to be sensitive to the orientation of the major axis of the landmark – the system will have to be able to learn to do the same.

The system is trained on both positive and negative examples of a given spatial term. We are thus for the time being leaving aside the issue of learning without explicit instances. The solution to that problem which was presented in Chapter 4 should in principle be applicable here; in fact, the system used for demonstration purposes in Chapter 4 was a somewhat simpler version of the system presented in this chapter.[1]

Positive and negative examples of particular spatial terms are constructed as illustrated in Figure 5.2. Using the scene editor shown here, one may construct scenes such as this positive example of English *above*. Any of the objects shown in the middle section of the screen may be chosen as either trajector or landmark.[2] These are then arranged relative to one another in the scene. Figure 5.3 presents a set of positive and negative examples of *above*; taken together, these constitute a (small) training set for that term. In each scene, the marks "LM" and "TR" indicate which object is the landmark and which the trajector. The scenes shown were constructed using the scene editor from Figure 5.2.

[1]The exact nature of the difference between the two systems will be discussed once the system design has been described, in §5.3.4.

[2]Each scene contains one trajector and one landmark, and no other objects.

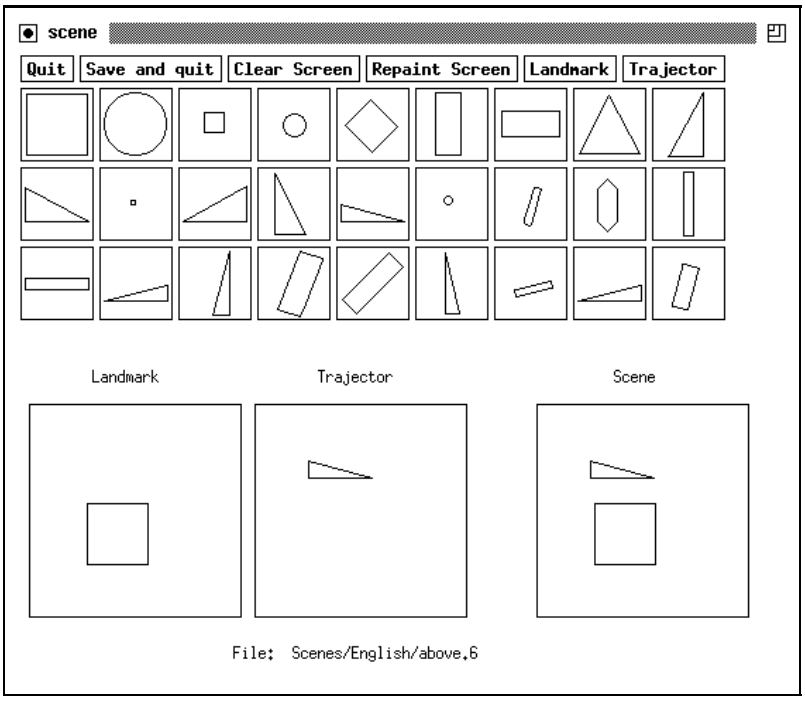
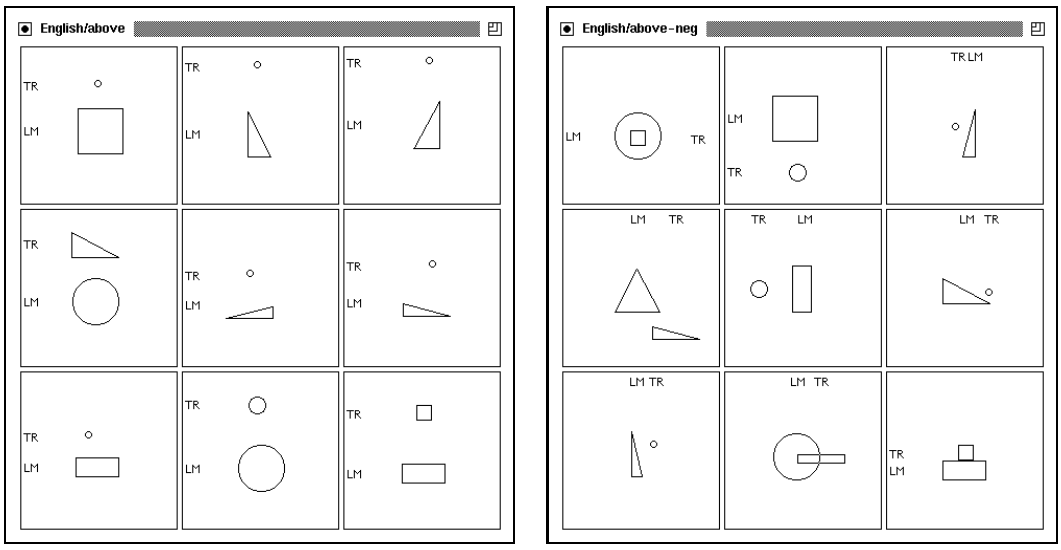


Figure 5.2: Constructing a positive example of English *above*



(a) (b)

Figure 5.3: Positive and negative examples of English *above*

5.2 Outline of a Solution

The general philosophy behind the approach taken here was touched on in Chapter 2: we assume that even very simple spatial concepts, such as English *above*, in fact involve a combination of evidence from various sources.[3] This section is devoted to outlining those sources, describing the perceptual features that the system has access to. These are divided into two classes: *directional* and *non-directional* features. Directional features are those features which are primarily orientational in nature, such as the direction of the imaginary directed line segment connecting the center of mass of the landmark to that of the trajector (the center-of-mass direction). Non-directional features include contact, tightness-of-fit, and inclusion between trajector and landmark; these are non-orientational in character.

5.2.1 Directional Features

A number of directional features are extracted from the scene, learned or built into the architecture. These features are of two general types:

- Relational orientations: Orientations which describe the location of one object relative to another. The two which are used in this version of the system both specify the location of the trajector with respect to the landmark, but in different ways:
  - Proximal orientation: the orientation of the (imaginary) directed line segment connecting the landmark to the trajector where the two objects are closest.
  - Center-of-mass orientation (CoM orientation): the orientation of the (imaginary) directed line segment connecting the center of mass of the landmark to that of the trajector.

  These two relational orientations are illustrated in Figure 5.4. Motivation for this particular choice of features may be found in §2.3.

- Reference orientations: These are orientations with which relational orientations may align. Examples are the major and minor axes of the landmark, and the four cardinal directions: up, down, right, and left.

All of the directional features mentioned above are extracted from the scene, with the exception of upright vertical, which may be either built into the architecture, or learned.

[3] See §2.3 for a discussion of this assumption, including motivation for the particular primitive features discussed here.

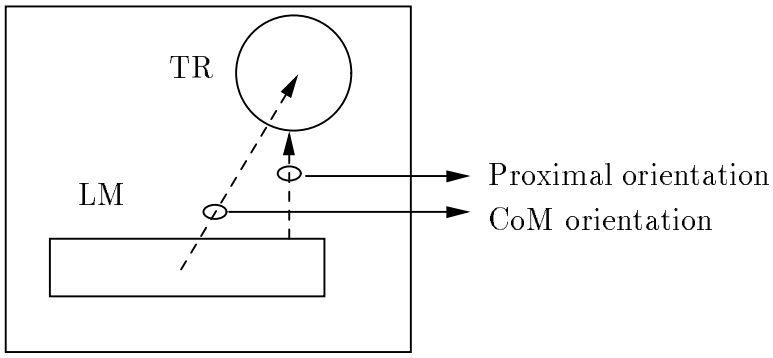


Figure 5.4: Proximal and center-of-mass orientations

There is one very simple principle underlying the manner in which these directional features are used here. This principle is *orientational alignment*: the degree to which a given relational orientation aligns with a given reference orientation.

For example, we might want to know how well the center-of-mass and proximal orientations align with upright vertical. Consider Figure 5.5. This figure shows the proximal and center-of-mass orientations between the landmark and the trajector (in dashed lines), and the degree to which each of these two relational orientations aligns with upright vertical, a reference orientation. Upright vertical is critical as a reference orientation in English, e.g. for *above*, but in other languages we find that other reference orientations are significant. For example, as mentioned earlier, in Mixtec the orientation of the major axis of the landmark is significant. Thus, the system needs to be able to detect this particular directional feature and use it as a reference orientation, so as to measure the alignment of relational orientations with it.

The more general idea being proposed here is that this notion of focusing on the degree of alignment between relational and reference orientations is a useful way to approach the problem, or at least that part of the problem that concerns directional features. While several further specific relational and reference orientations will be introduced in later chapters (see for example Chapter 6), this central idea of measuring alignment is used throughout, unchanged.

5.2.2 Non-directional Features

Non-directional features such as inclusion and contact may be detected by observing the trajector boundary, and seeing if it or portions of it lie within, or immediately adjacent to, the landmark. For example, if each point on the trajector boundary is

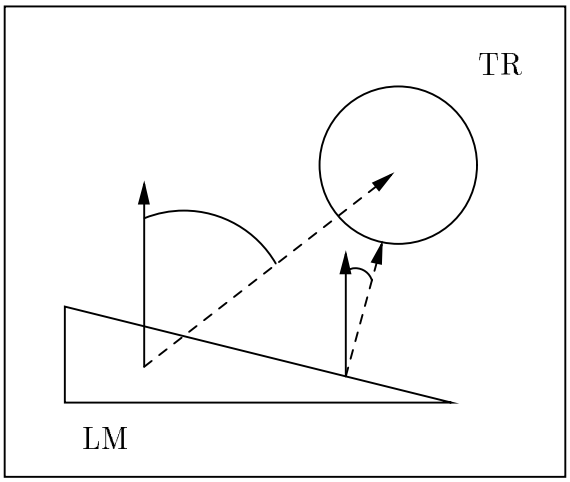


Figure 5.5: Orientation comparison

located within the landmark, this tells us that the trajector as a whole lies within the landmark, as in Figure 5.6 (a). And if at least one point of the trajector boundary is immediately adjacent to the landmark, but none of the trajector boundary is actually within the landmark, this tells us that the trajector as a whole is in contact with the landmark, but has not penetrated it. This situation is shown in Figure 5.6(b).

It is important to distinguish two different cases here. There are situations in which the fraction of the trajector boundary along which some perceptual feature occurs is not relevant – rather, all that is relevant is that the feature occurred at some point. Consider English *on*, for example, as illustrated in Figure 5.7. It makes no difference how much of the trajector boundary is in contact with the landmark; (a) and (b) are equally good examples of *on*. What is significant is that there is contact at some point in both cases.[4] The system should be constructed so as to be able to learn to respond equally to scenes (a) and (b).

On the other hand, there are cases in which the fraction of the trajector boundary along which a perceptual feature occurs *is* significant. As an example of this, consider English *in*, illustrated in Figure 5.8. Here, (a) is an excellent example of *in*, (b) is a fair one, and (c) is a poor one. This graded response varies directly with the proportion of the trajector boundary which lies within the landmark. The system should of course also be able to exhibit behavior of this sort.[5]

[4]Of course, the underlying reason for this is that it is critical to *on* that the landmark *support* the trajector (see Chapter 7 for further discussion of support). And visually, support may be indicated by contact at a single point, or along a larger part of the trajector boundary.

[5]Note that not all languages treat inclusion in this manner. In particular, the Bengali term

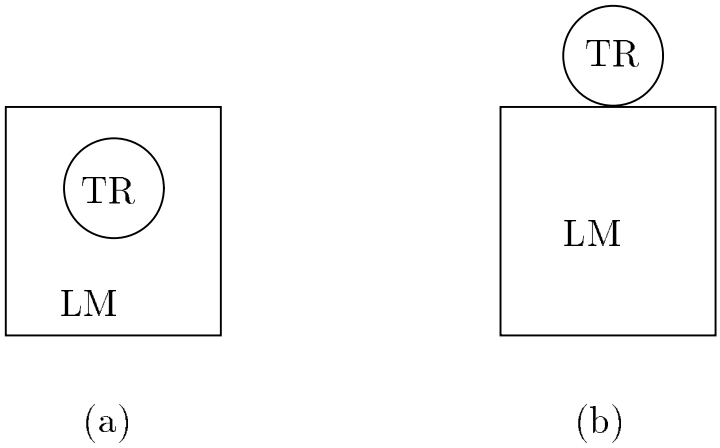


Figure 5.6: Inclusion and contact

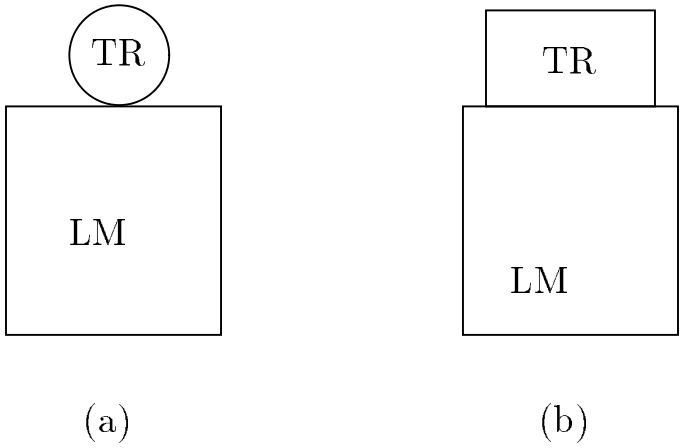


Figure 5.7: English *on*: contact at at least one point along the trajector boundary

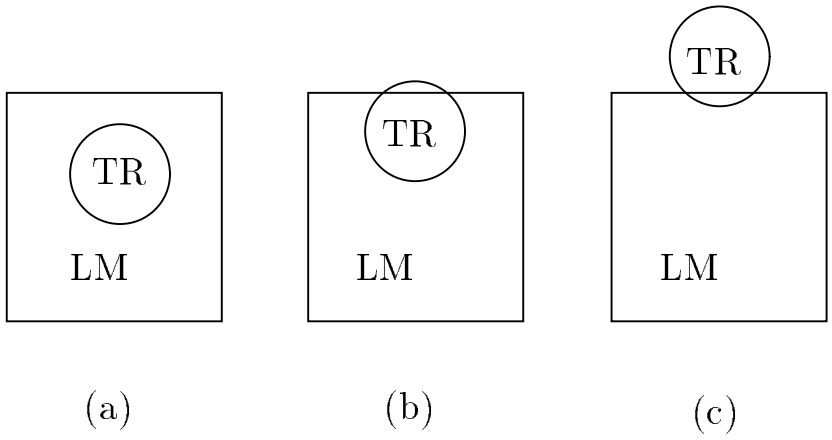


Figure 5.8: English *in*: inclusion along the length of the trajector boundary

---

The system as constructed covers both of these cases, as we shall see. But there is one type of non-directional feature which has not been incorporated into this version of the system: there are currently no features related to *distance* between objects. Extending the system to handle distance, however, should be straightforward. We outline here two distance-related features which appear to be significant.

There is evidence suggesting that, just as proximal and center-of-mass orientations are both necessary features, so are both the *proximal distance*, and the *center-of-mass distance*, where these are the lengths of the imaginary line segments whose orientations give us the proximal and center-of-mass orientations, respectively. Recall that these were illustrated in Figure 5.4. Thus the definitions for these two features are as follows:

- Proximal distance: the length of the (imaginary) directed line segment connecting the landmark to the trajector, where the two objects are closest.
- Center-of-mass distance: the length of the (imaginary) directed line segment connecting the center of mass of the landmark to that of the trajector.

Consider Figure 5.9 for example. The scene in (a) is a good example of English

bhethoray, which denotes complete inclusion of the trajector within the landmark, does not exhibit a graded response of this sort. If even a small portion of the trajector boundary lies outside the landmark, the scene as a whole will be judged a very poor example of *bhethoray* [Ahmad, 1992].

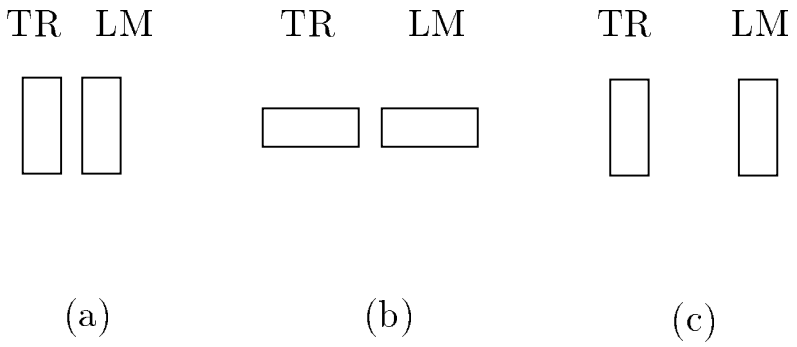


Figure 5.9: The role of distance features in English *near* (see text)

---

near, (b) is not quite as good an example,[6] and (c) is a good deal worse. What is interesting here is that in (a) and (b), the proximal distance is the same, so the fact that (a) seems a better example of *near* than (b) indicates that there is more to the concept than just the proximal distance. Analogously, in (b) and (c) the center-of-mass distance is the same, so the judgment difference for these two scenes cannot be due to that feature. The assumption is that these two distance features both play a role in judgments of this sort, much the way proximal and center-of-mass orientations both play a role in other judgments (recall §2.3). Further discussion of the role of distance features may be found in Chapter 7.

5.3 Implementing the Solution

Now that the general ideas behind the approach taken here have been presented, we may move on to the architecture of the system itself. This is shown in Figure 5.10. The system as a whole is trained under a variant of back-propagation.

While there is a good deal of detail shown in this figure, 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*

[6] Opinions vary on this, but most people who were asked for their judgments, in informal questioning, found (a) to be a somewhat better example of *near* than (b). Nobody judged (b) to be better.

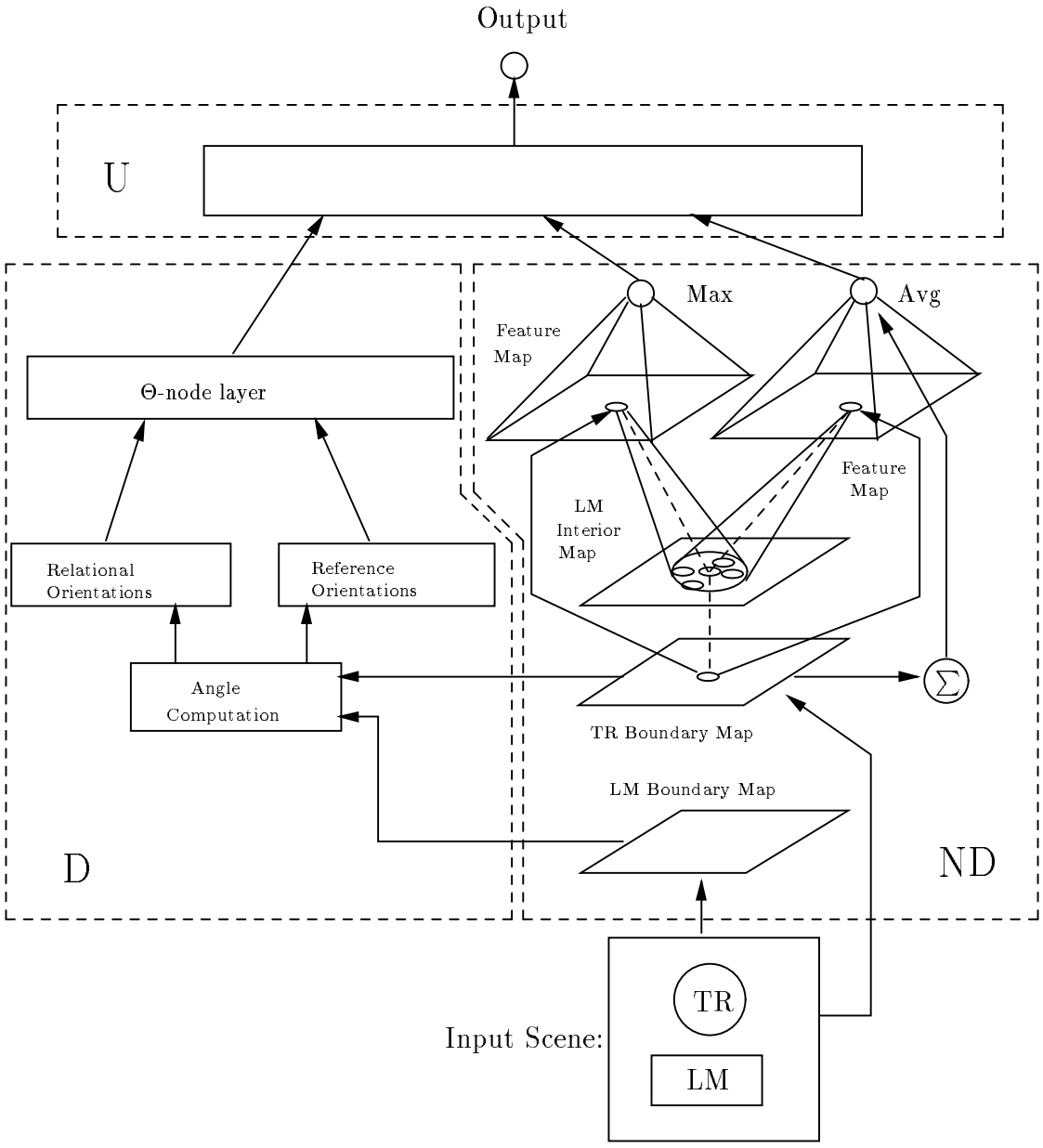


Figure 5.10: Architecture overview

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 **D**, **ND**, and **U**. **D** is responsible for handling directional features, while **ND** handles non-directional features. Both of these lower two modules are highly structured, and were designed with the spatial concept learning task very much in mind. Module **U**, on the other hand, is an unstructured upper layer, leading to the output node. There is full connectivity between the top nodes in modules **D** and **ND**, and the layer of module **U**. Learning occurs in each of these modules.

  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 and *enhanced generalization* 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.

We begin by describing the computation, from the image, of perceptual features which the system takes as primitives. Then, the structural devices in modules **D** and **ND** are described in turn.

5.3.1 Preliminaries

Each scene that the system receives as input undergoes pre-processing, for the purpose of extracting from the image a number of features which will be used by the system. This pre-processing is briefly described below.

Most of this pre-processing is not currently done in a connectionist fashion, as we shall see. There is nothing inherently non-connectionist about the tasks, however, and connectionist implementations could be easily constructed.

Topographic Maps

As mentioned above, the system accepts input by copying the outlines of the trajector and the landmark into the TR and LM boundary maps, respectively.

Once this has been done, the system determines the interior of both the landmark and the trajector. This operation is illustrated for a single object in Figure 5.11.

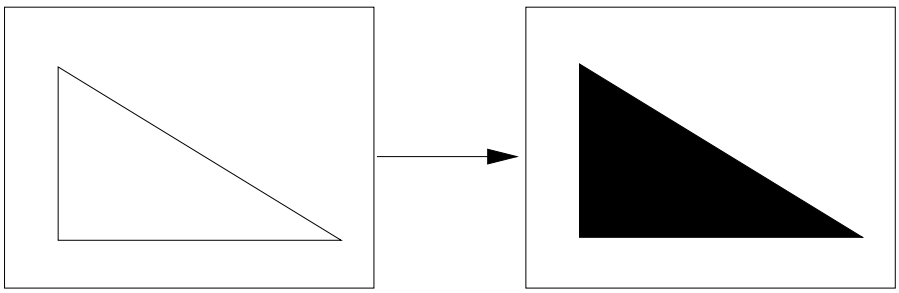


Figure 5.11: Computing the interior of an object

These "filled-in" copies of the two relevant objects are then used in further pre-processing, as discussed below. In addition, the "filled-in" landmark is stored in the *LM Interior Map*, which, as can be seen in Figure 5.10, provides input to higher levels of the architecture.

This "filling-in" operation is accomplished using a simple spreading mechanism, similar to the basic operation of spreading activation proposed in Shimon Ullman's work on *visual routines* [Ullman, 1984].[7]

There are of course other means to accomplish this task; for example, [Sejnowski and Hinton, 1987] present a model based on the Boltzmann machine [Ackley *et al.*, 1987] which performs a similar operation. This might be used in place of the mechanism used here.

This computation is the only pre-processing computation which is performed by a connectionist network.

Center of Mass

Once the trajector and landmark outlines have been filled in, we can easily compute the centers of mass for these objects. This will be necessary in order to find the center-of-mass orientation for the scene.

For a given object, let n be the area of the object (easily determined by counting the activated pixels in a filled-in copy of the object), Obj be the set of points making up the object, and $\vec{p} = (x, y)$. Then the center of mass of the object is given by $(\bar{x}, \bar{y})$,

[7]The actual method used begins by activating every pixel in the scene, and then picking some point on the perimeter of the scene, which is assumed to lie outside the object, and deactivating the pixel at that point. Each neighbor of this deactivated pixel is then deactivated, and the deactivation spreads out from pixel to pixel, until the object boundary is reached. The boundary is never deactivated, and therefore the interior of the object is never deactivated either. The resulting map contains a filled-in copy of the original outline.

where

$$\bar{x} = (\frac{1}{n}) \sum_{\vec{p} \in Obj} x; \qquad \bar{y} = (\frac{1}{n}) \sum_{\vec{p} \in Obj} y \tag{5.1}$$

Major Axis Orientation

As mentioned earlier, the major axis orientation of the landmark appears to be useful as a reference orientation. There exists a standard computer vision method for determining the major axis orientation of an object. This method, which is the one used here, amounts to finding the orientation of the line for which the integral of the square of the distance to points in the object is a minimum, i.e. the orientation of minimum dispersion. The integral used is

$$E = \int \int_I r^2 b(x,y) dx dy \tag{5.2}$$

where r is the perpendicular distance from the point (x, y) to the line sought after, I is the image, and $b(x, y)$ is the *characteristic function* of the object, i.e. 1 if (x, y) is part of the object and 0 otherwise. The orientation of the line which minimizes this integral is the major axis orientation of the object. There exists a straightforward formula which yields this orientation of minimum dispersion; details may be found in [Ballard and Brown, 1982; Horn, 1986].

Proximal and Center-of-Mass Orientations

Determining the proximal orientation is currently done in the simplest possible manner. For each pair of points $[\vec{l}, \vec{t}]$ where $\vec{l}$ is a point on the boundary of the landmark and $\vec{t}$ is a point on the boundary of the trajector, we find the distance between the two points, and pick that pair of points $[\vec{l_m}, \vec{t_m}]$ for which the distance is minimum. The proximal orientation is then the orientation of the directed line segment connecting $\vec{l_m}$ to $\vec{t_m}$.

The center-of-mass orientation is also trivial to compute, once the centers of mass of the two objects have been determined, as described above.

5.3.2 Directional Features

Now that the pre-processing operations have been presented, we can move on to describe the system proper. We begin with that part of the architecture which handles directional features. This is module **D** in Figure 5.10. The pre-processing related to directional features, which we have just described, is denoted by the box labeled "Angle Computation" within this module. From here, the computed reference and

relational orientations are copied to their respective buffers one layer up in the architecture. Thus, the only section of this module that we have not yet discussed at all is the box labeled "Θ-node layer". We now turn our attention to it.

Recall from §5.2.1 that the central question being addressed here, concerning directional features, is *orientational alignment*:

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

For example, as was shown in Figure 5.5, we might want to know how well the center-of-mass and proximal orientations align with upright vertical.

All orientations in the system are represented in radians. Given this, we can easily measure the degree of alignment of two orientations by using a simple Gaussian of the sines and cosines of the two angles:

$$f(r) = \exp\left[-\frac{(sin(\theta) - sin(r))^2 + (cos(\theta) - cos(r))^2}{\sigma^2}\right] \tag{5.3}$$

where r is the relational orientation, and θ is the reference orientation to which it is being compared. σ controls the "width" of the Gaussian. This will return the value 1.0 when the two orientations are perfectly aligned, and will drop off as the relational orientation deviates from the reference orientation. The use of sine and cosine in the function has the advantage that under this scheme, orientation pairs such as 0.1 radians and $1.99pi$ radians will be considered to be "near" each other, i.e. they will produce a response close to 1.0. This is as it should be, since the two orientations are nearly in alignment.[8]

Units which perform this comparison are termed Θ-nodes, illustrated in Figure 5.12. Note that the relational orientation r is supplied as input to the node, while the reference orientation θ is stored in a variable internal to the node. This internal variable may be clamped to a value supplied externally, as shown by the dotted line leading to the internal variable. The σ for the Gaussian is also kept as an internal variable in the node.

The Θ-node layer in the architecture is simply a set of such Θ-nodes, each comparing a relational orientation to a reference orientation.

The system as a whole is trained under a variant of back-propagation, and Θ-node internal variables may be trained together with the weights of the network in which the Θ-nodes are embedded. In order to train the internal variables of a Θ-node, we need to determine the partial derivative of the error with respect to these variables, i.e. $\frac{\partial E}{\partial \theta}$ and $\frac{\partial E}{\partial \sigma}$. These are easily obtained once we find the partial derivative of f (recall Equation 5.3) with respect to each of the internal variables:

$$\frac{\partial f}{\partial \theta} = \frac{-2(cos(\theta) - cos(r))sin(\theta) + 2cos(\theta)(sin(\theta) - sin(r))}{\exp\left[\frac{((sin(\theta)-sin(r))^2+(cos(\theta)-cos(r))^2)}{\sigma^2}\right]\sigma^2} \tag{5.4}$$

[8] See [Zemel *et al.*, 1992] for another formulation of directional units in connectionist networks.

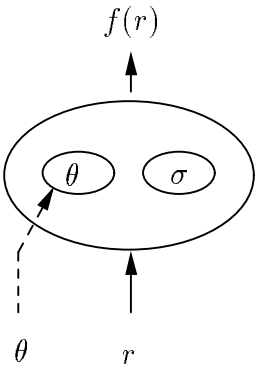


Figure 5.12: Internal structure of a single Θ-node

$$\frac{\partial f}{\partial \sigma} = \frac{2((sin(\theta) - sin(r))^2 + (cos(\theta) - cos(r))^2)}{\exp\left[\frac{((sin(\theta)-sin(r))^2+(cos(\theta)-cos(r))^2)}{\sigma^2}\right]\sigma^3} \tag{5.5}$$

Every Θ-node will learn its σ, and several, though not all, will learn their reference orientations θ as well. The ones that do not learn their values for θ accept external input which tunes the node to a particular reference orientation on the fly, via the dotted link in Figure 5.12. We shall see examples of both means of handling reference orientations in Θ-nodes.

The Θ-node layer shown in the architecture contains 10 Θ-nodes:

- There are two Θ-nodes which take the major axis orientation of the current landmark as their reference orientation. One of these nodes takes the proximal orientation as its relational orientation input, and the other takes the center-of-mass orientation.

- There are 8 Θ-nodes which do not accept an externally-imposed reference orientation. Rather, each of these learns an appropriate value for its θ internal variable, i.e. its reference orientation. Four of these nodes take the proximal orientation as relational orientation, and four take the center-of-mass orientation.

Learning in connectionist networks in general is often facilitated by initializing weights to values that the network user knows to be near the desired solution. Similarly, learning may be facilitated in the system described here by initializing the θ variables of the eight Θ-nodes which learn their reference orientations, such that the initial values for the reference orientations are at or near values which the network

user knows to be of significance. For example, in learning English spatial terms, the reference orientations 0, $\pi/2$, π, and $3\pi/2$, corresponding to the four cardinal directions "right", "up", "left", and "down", will be useful for such terms as *above, below, beside* and the like. Therefore, the learning process can be simplified by initializing the reference orientations of Θ-nodes that learn their θ values to these directions.[9]

5.3.3 Non-directional Features

Feature Maps

Returning momentarily once again to Figure 5.10, we now consider module **ND**, which handles non-directional features such as inclusion and contact, among others.

To begin with, notice that the TR and LM boundary maps, and the LM interior map, discussed above, all lie within this module.

Directly above the LM interior map, and receiving input from both it and the TR boundary map, are two *feature maps*. These are the central structural devices of this module.

The design of a single feature map is shown in Figure 5.13. 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 all different positions are constrained to be identical (see [LeCun, 1989] for details on this technique of weight-sharing). This implies that there are actually only five *weights* to be adjusted for a single feature map, despite all the links: one for the center links, and four for the surround links.
- Each node in the feature map is gated by the node in the corresponding position in the TR boundary map, such that the feature map node will respond only if the node in the TR boundary map is activated.

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

$$f_i = [\sigma(\sum_{j \in N(i)} l_j w_{ij}) \times t_i], \tag{5.6}$$

[9] As it turns out, initializing weights is particularly useful here, since it is difficult to learn the mean and variance of a Gaussian simultaneously when both have been randomly initialized. The approach taken here then has been to initialize the mean – i.e. θ – to one of the cardinal directions, and to initialize σ, the variance, to some intermediate value.

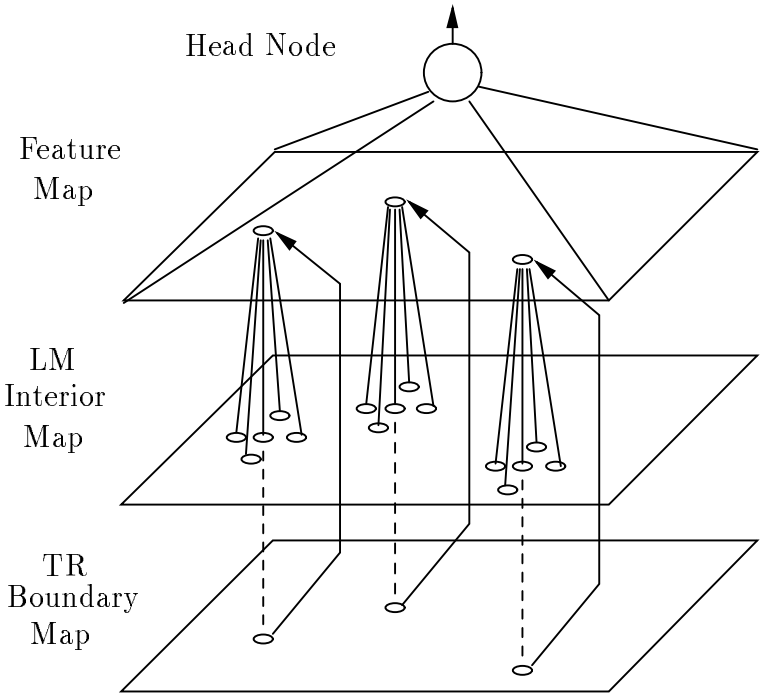


Figure 5.13: Feature map architecture

where $\sigma()$ is the usual sigmoidal squashing function (see Chapter 3), 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 activations (either 0 or 1) of the LM interior map unit at position i, and of 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.

For example, imagine that the receptive field for feature map units has been trained so as to respond strongly to the presence of the landmark interior at the central point being examined. Then we might expect to see situations like the one portrayed in Figure 5.14. Here the input scene shows a small square trajector partially inside a triangular landmark. The receptive field within the LM interior map for a single unit in the feature map is shown,[10] as is the gating input from the TR boundary map to that feature map unit. The feature map as a whole will contain a strip of activation, one unit wide, corresponding to that portion of the trajector boundary which is inside the landmark. This is because the receptive field has been trained to cause feature map units to respond strongly to the presence of LM interior at the central point of the receptive field, but since the feature map units are gated by the TR boundary map, only those feature map units which are both within the LM interior and a part of the trajector boundary will be strongly activated: this gives us the "C"-shaped strip of activation we see. In essence, the feature map is detecting *inclusion* of parts of the trajector boundary in the landmark interior.

To consider another example, imagine now that the feature map receptive field has been trained so as to respond strongly to situations in which there is no landmark interior at the central point of the field, but there is at one of the four peripheral receptive field points. This would result in configurations like the one shown in Figure 5.15. Here, the feature map contains a single straight strip of activation, corresponding to those points on the trajector boundary which are immediately adjacent to, but not within, the landmark. Thus, the feature map is now essentially detecting *contact* between trajector boundary points and the landmark.

Recall from §5.2.2 above that we can distinguish at least two ways in which features of this sort may be dealt with in language. On the one hand, for terms like English *on*, it does not matter how much of the trajector boundary is in contact with the landmark; all that matters is that contact exists at some point along the trajector boundary. On the other hand, for such terms as English *in*, the amount of the trajector boundary which is contained within the landmark makes a difference in the

[10]Only three of the five links are shown, to avoid an excessively cluttered exposition.

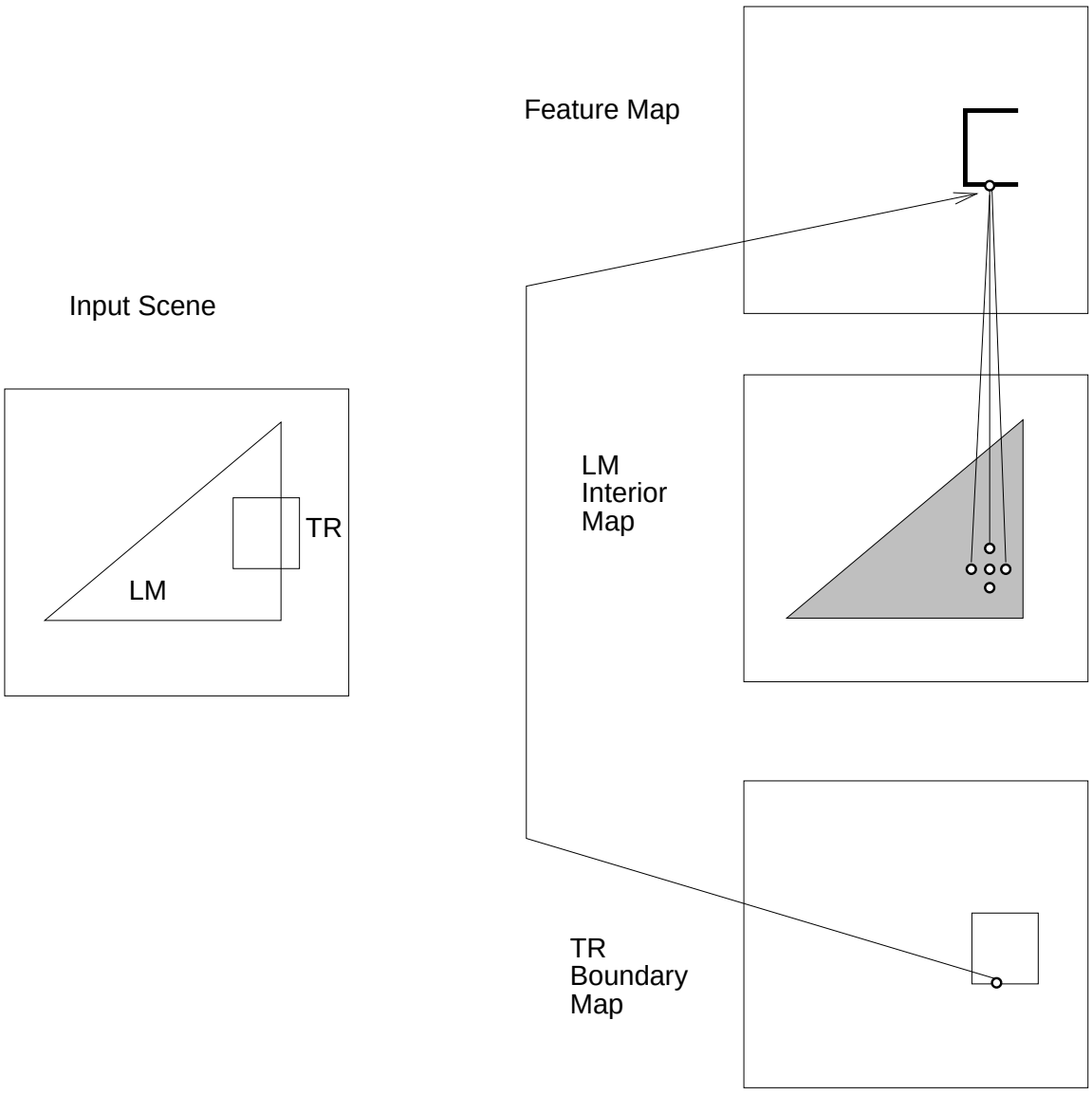


Figure 5.14: Detecting inclusion (see text)

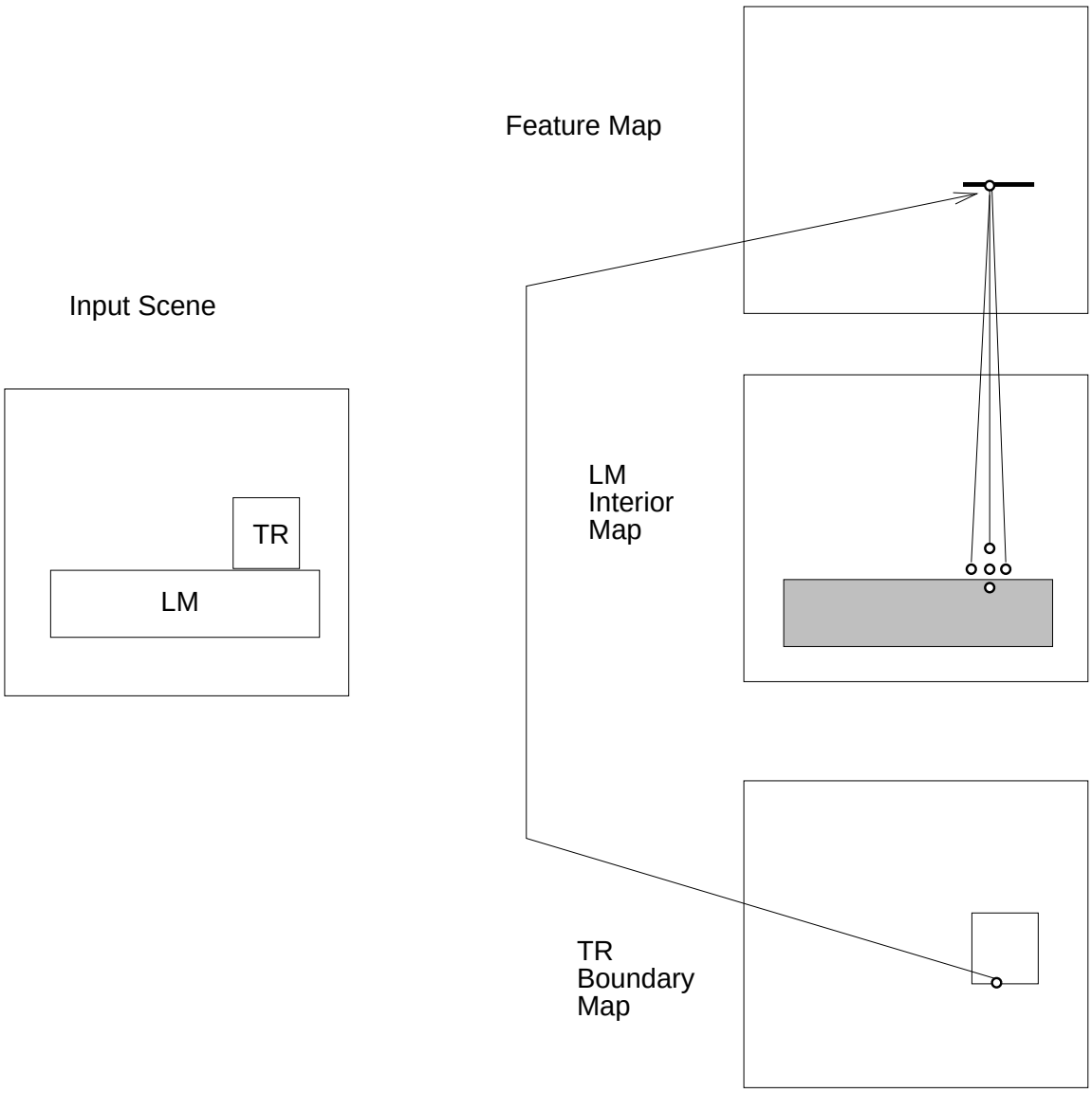


Figure 5.15: Detecting contact (see text)

judged applicability of the term to a scene. Thus, we have one situation in which all we care about is the simple presence or absence of a feature, and another in which we care about the amount of the trajector boundary along which a feature is present.[11]

This distinction provides the motivation for the two sorts of head nodes (recall Figure 5.13) that exist in the system, taking a function of all the units in the feature map below:

- **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.

$$F = [\sum_i f_i]/n$$

  where n is the number of points in the TR boundary.[12] Since f_i is zero when there is no TR boundary point at i (recall Equation 5.6), we are actually averaging over those points in the visual field which are part of the TR boundary, not over the entire map.

The idea here is that a feature map head node which takes the maximum response over the feature map units in the map below it will allow the system to respond equally to, for example, contact at a single point and contact over a large extent of the trajector boundary. This is so since the maximum is insensitive to the number of nodes in the feature map which actually reach that maximum. On the other hand, a head node that takes the average of the feature map units will allow the system to deliver a graded response depending on the fraction of the trajector boundary along which a given feature occurs.

The overall architecture shown in Figure 5.10 contains two feature maps, one headed by a node returning the maximum over the map, the other headed by an averaging node.

Learning in Feature Maps

As mentioned above, the system is trained under a variant of back-propagation. In order to train the weights leading into feature map units, several minor changes to

[11] Note that the feature of inclusion is not always of this latter variety. In particular, the Bengali term *moddhay* is applicable even if only a small portion of the trajector boundary is within the landmark.

[12] The number of units which are activated in the TR boundary map are summed, and this sum is supplied as input to the averaging head node. This can be seen in Figure 5.10.

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.

- One cannot back-propagate error through a *Max* node, since the maximum function is non-differentiable.[13] Therefore, the approach taken here is to view the *Max* node essentially as a "virtual" node of the type found in the map below. More precisely, the *Max* head node of a feature map is treated as if it were itself that node i in the map below which had the maximum response. Clearly, since its output is the maximum of all outputs in the feature map below, it will behave like node i during the forward pass. We arrange to have it behave like node i during the backward pass as well. Essentially, the feature map structure as a whole, head node a included, will behave as if it were node i.

  If m is the *Max* head node of a feature map, then during the backward pass of back-propagation, the value $\delta_m = -\frac{\partial E}{\partial net_m}$ is computed for head node m as if m were that unit i in the map below which had the maximum response during the forward pass, i.e as if m were a unit whose activation function was of the form shown in Equation 5.6, located at position i in the map. This value is then propagated down to feature map node i itself, such that $\delta_i = \delta_m$, enabling us to determine the weight updates for the incoming weights of node i. Node i then updates the weights on its incoming links, and constrains the incoming weight vectors for all other feature map units to be identical to that for node i. It is in this sense that m, and indeed the feature map structure as a whole, is treated as if it were that feature map node i with the maximum response.

- Things are a good deal more straightforward in the case of *Avg* head nodes, since the average is a differentiable function. Let a be an averaging head node. In order to back-propagate error through node a, we need to find the quantity $\delta_a = -\frac{\partial E}{\partial net_a}$, where net_a is the net input to a, i.e. $\sum_i f_i$. (Note that this assumes that each link connecting a feature map unit f_i to the head node a is weighted with value 1.0.) This quantity is easily found:

$$\delta_a = -\frac{\partial E}{\partial net_a} = -\frac{\partial E}{\partial o_a}\frac{\partial o_a}{\partial net_a} = -\frac{\partial E}{\partial o_a}\frac{1}{n} \tag{5.7}$$

  where o_a is the output of node a, and n is the number of points in the trajector boundary.

  We can now use the usual back-propagation formula to find the updates for the incoming weight vectors for feature map units. These are then all updated such

[13] There do exist differentiable functions which are similar in functionality to the maximum, such as the *softmax* function discussed in [Bridle, 1990]. The maximum was used here because it is simpler than softmax, and because the required deviation from straightforward back-propagation is both minor and effective.

that corresponding links at different feature map locations are constrained to have the same weight.

Using these minor 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.

5.3.4 The Architecture in Review

Now that the structured modules in the design have been discussed in detail, we may turn our attention to two remaining matters: firstly, the difference between the structural devices used here and the ones used in the simplified system in Chapter 4, and secondly, the unstructured portion of the system, which is identical in the two versions of the system.

The simplified version of the system used in Chapter 4 accepted only trajectors which were a single point. As a result, there was no need for elaborate feature map structures of the sort outlined above for the detection of non-directional features. Instead, there were three single nodes of the sort found within feature maps in this system, each with a highly localized receptive field viewing the LM interior map. It was arranged that each of them viewed the LM interior map at the point at which the point trajector was currently located. Thus, each of them could learn a non-directional feature such as inclusion or contact of that trajector point relative to the landmark interior. The output of these three nodes fed into the unstructured upper portion of the network, just as the outputs of the feature map head nodes do in the current architecture.

Regarding the current architecture, we have already covered in detail the structure of modules **D** and **ND** (recall Figure 5.10), the two structured modules of the system. The design of module **U** 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 **U** 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 **D**) indicating the degree to which the proximal orientation aligns with upright vertical, and a feature map head node (in module **ND**) indicating the presence or absence of contact.

These learned features can then be combined through module **U**, in more or less "classic" connectionist fashion, to yield an output value. Thus, the nodes in module

U represent combinations of the features in the two modules below. 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.

5.3.5 Motivations from Neuroscience

As we noted in Chapter 1, no attempt is made in this thesis to provide structure-to-structure correspondences between the architectural structures used here and actual neural circuitry in the brain. Nonetheless, the model as a whole is neurally-inspired in that it makes use of essentially neuron-like computing units. In addition, there are a number of neurobiological results, some classic and some more recent, which can serve to motivate the task as a whole and some of the particular structures used in the model.

In the brain, there are two distinct major pathways that project forward from the visual areas of the occipital lobe to cortical association areas. One of these projects to the temporal lobe and the other to the parietal lobe. Recently, workers in neuroscience have come to hypothesize that these two paths are specialized for two distinct functions; specifically, that the occipito-temporal pathway is specialized for identifying *what* an object is, while the occipito-parietal pathway is specialized for identifying *where* the object is [Mishkin, 1972; Ungerleider and Mishkin, 1982].

This functional split provides some motivation for the spatial relations learning task being addressed here, as it indicates that there seems to be a neuroanatomical basis for considering the problem of where an object is – which is after all the focus of the work presented here – separately from the issue of object recognition or classification. The observation that "posterior parietal cortex seems to be concerned with the perception of spatial relations among objects" [Mishkin, 1972] is helpful in that it reassures us that we are not making an artificial division when we restrict our attention to spatial relations among objects, as opposed to any of their many other visual attributes.

This reassurance regarding the task as a whole then is one sort of motivation which work in neuroscience provides. In addition to this, there are a number of architectural structures present in the model here which are similar in overall flavor to structures which are known to exist at various levels of the visual system, although almost certainly not in the configuration used here.[14] These structures are briefly

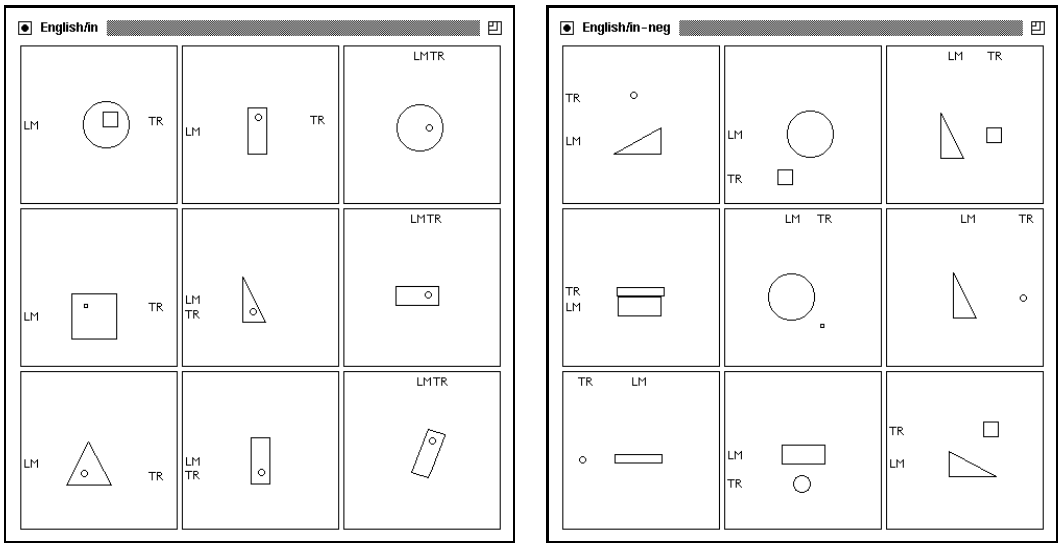
[14]Note that any pretense at a structure-to-structure mapping would fall apart at this point, since the neuroanatomical evidence for these structures comes from widely separated parts of the visual system.

outlined below.

- Retinotopic maps: Maps of units which roughly preserve retinal ordering, such that nearby positions in the retinal image are represented by nearby units in the map, are found at several levels within the visual system [DeValois and DeValois, 1990]. This provides some motivation for the trajector and landmark boundary and interior maps used in the system here. It is worth noting, however, that there are some fundamental differences between the maps found in the model and those in the brain. Most particularly, the mapping to visual cortex is in fact logarithmic [Tootell *et al.*, 1982], such that logarithmically spaced concentric rings in a visual stimulus map to equally-spaced concentric rings in the cortical representation. This logarithmic mapping is not present in the model, although it should not be incompatible with the basic architectural ideas described here.

- Center-surround receptive fields: In a classic study, [Kuffler, 1953] reported the existence, in the ganglion cells of the cat retina, of cells with circularly symmetric center-surround receptive fields. These receptive fields consist of opposing centers and surrounds, such that an excitatory center is paired with an inhibitory surround, and vice versa. Within the model presented here, the circular receptive fields of units in the feature maps are similar in overall flavor to Kuffler's receptive fields, although much simplified. In particular, once trained to detect inclusion (recall for example Figure 5.14 and the accompanying discussion), the receptive fields in the model tend to have an excitatory center paired with a surround which is, if not always inhibitory, at least less excitatory than the center.

- Orientation sensitive cells: While there are no known neuroanatomical structures which correspond directly to Θ-nodes, it is well-known that the majority of cells in visual cortex are in fact sensitive to the orientation of visual stimuli [Hubel and Wiesel, 1959; Hubel and Wiesel, 1962]. Thus, the basic principle of orientation sensitivity of visual cortical cells is well-established, although not for such derived orientations as the ones used in this system, which do not correspond to any physical luminance or color discontinuity in the image itself. However, in this regard, it is worth noting that [von der Heydt *et al.*, 1984] have shown that cells in area 18 of monkey visual cortex are sensitive to the orientations of *illusory contours*. These are contours which are perceived by human observers of particular visual stimuli, but which do not correspond to physical luminance or color discontinuities in the image (see [Kanizsa, 1979] for a review of this phenomenon). So the results of von der Heydt *et al* are striking because they point out that there are cells in visual cortex which respond to orientations which are not physically there in the image. Thus, the principle of sensitivity to non-physical orientations has a precedent, even if not for the



(a) (b)

Figure 5.16: Positive and negative training examples for English *in*

particular non-physical orientations we are concerned with here. These results can be viewed as a very general, loose form of motivation for Θ-nodes.

So there is a form of motivation from neuroscience for some aspects of the model at hand, although this falls far short of providing any basis for claims of structural isomorphisms between the model and the brain. Having now covered the architecture itself in detail, and neuroscientific motivations for it briefly, we can move on to consider the system's performance.

5.4 Results

The system was trained under quickprop [Fahlman, 1988], a variant of back-propagation which exhibits fast convergence. The spatial terms were learned one at a time, using the architecture shown in Figure 5.10. As mentioned earlier, the training runs took place using explicit negative evidence for each spatial term.[15] Training was in all cases very fast, requiring under 100 quickprop epochs to attain error rates under 0.0001.

Figure 5.16 presents a typical training set, that for the English preposition *in*. Here, nine positive instances of *in* are presented in (a), and nine negative instances in (b).

[15]See Chapter 4 for a means of avoiding the use of explicit negative evidence.

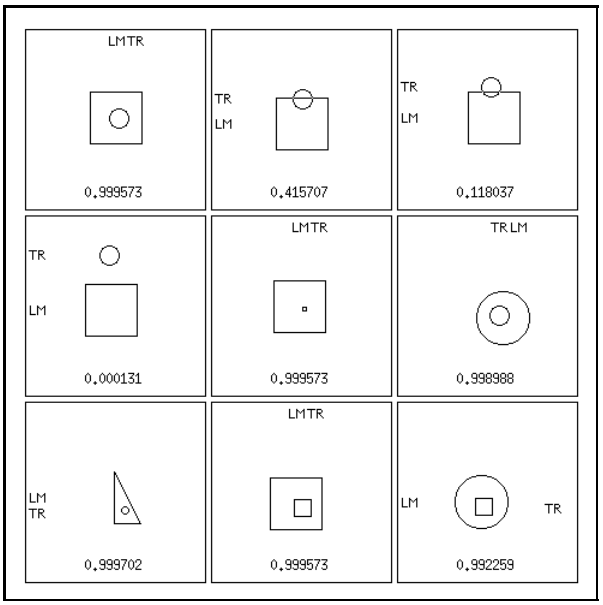
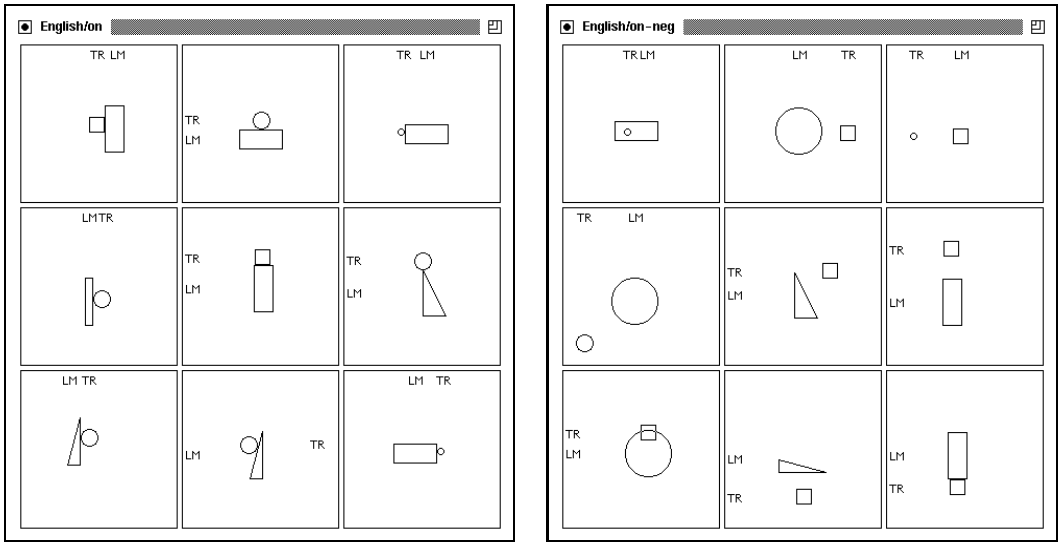


Figure 5.17: English *in*: a test set

Figure 5.17 illustrates the performance of the system on a test set, after it has been trained on the above training set for English *in*. The number between 0 and 1 at the bottom of each scene is the strength of the system's response to the scene. Note that these responses are graded, varying with the amount of the trajector boundary which lies within the landmark.

There is nothing in the training set itself that dictates that the response should be graded in this manner. In fact, other training runs on *in* have yielded responses which were far more discrete than this; both types of results are valid, given this training set. It is worth noting that the system's ability to interpolate in this way can be attributed to the *Avg* node heading a feature map, since runs which were made on an architecture with two feature maps headed by *Max* nodes, rather than one headed by a *Max* node and one by an *Avg* node, did not yield the same smoothly varying graded results. Recall that graded responses of this sort, varying with the amount of the trajector boundary inside the landmark, were the motivation for incorporating *Avg* nodes in the system in the first place.

Figure 5.18 presents positive and negative training examples for another English preposition, *on*, and Figure 5.19 presents the performance of the system on a test set, after having learned *on*. The number between 0.0 and 1.0 at the bottom of each scene indicates how good an example of *on* the system has judged the scene to be. Note that in this case, the amount of the trajector boundary over which the non-directional feature of contact exists does not seem to influence the output of the system to any significant degree; all responses shown are 0.997 or higher, despite the fact that the



(a) (b)

Figure 5.18: Positive and negative training examples for English *on*

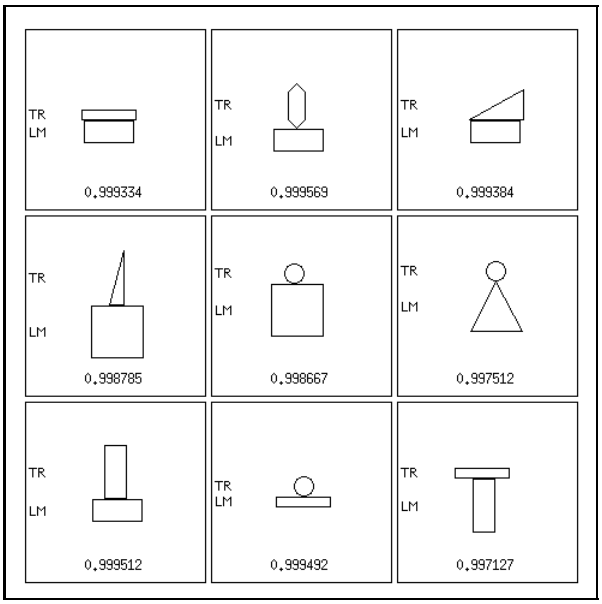


Figure 5.19: English *on*: a test set

amount of trajector boundary in contact with the landmark varies substantially.

Just as the earlier presentation of *in* reflected the importance to the system of *Avg* nodes heading feature maps, this presentation reflects the importance of *Max* feature map head nodes. The inclusion in the system of a feature map headed by a *Max* node greatly facilitates the learning of spatial terms like *on*, which are insensitive to the amount of the trajector boundary along which a non-directional feature occurs. An architecture almost identical to the one used here, but with two feature maps headed by *Avg* nodes instead of one headed by an *Avg* node and one by a *Max* node, took an average of 686 quickprop epochs to converge to less than 0.01 error learning English *on*, while the original architecture required an average of 72 quickprop epochs.[16]

5.4.1 Prototype Effects

Recall from the discussion of prototype effects in Chapter 2 that one possible explanation for prototype effects, both in general and specifically here in the domain of spatial categorization, is that prototypical instances of categories are instances in which there is strong evidence for the category from each of a number of different factors, each of which provides partial support for category membership. More concretely, one could imagine a prototypical *above* as being one in which both the proximal and center-of-mass orientations are perfectly aligned with upright vertical.

We have already seen a case of graded category membership, namely English *in*. Figure 5.20 presents another, in which the overall response is more clearly dependent on a combination of factors. This is English *above*, two of the contributing factors of course being the proximal and center-of-mass orientations. This figure shows the results of running a system which had been trained on English *above*, using the training set shown in Figure 5.3, on a test set.

The first row of scenes in Figure 5.20 shows a small circular trajector located relative to a triangle. In the three scenes, the proximal orientation is kept constant, but the trajector "slides" up the hypotenuse of the triangle, so that the center-of-mass orientation gets closer and closer to upright vertical. At the same time, the scenes become increasingly prototypical cases of *above*, as reflected in the responses given by the system to these three scenes.[17] This then shows the role played by the center-of-mass orientation in classifications of English *above*.

The second row of scenes shows that the proximal orientation plays a role in these classifications as well.[18] The center-of-mass orientation is roughly the same in the three scenes of this row, but the proximal orientation varies, getting closer and closer to upright vertical. Correspondingly, the system's response differs from scene

[16]These numbers were obtained by averaging over 5 runs under each architecture.

[17]Note that these scenes correspond to the situations which were portrayed in Figure 2.6(a) and Figure 2.8, and which were discussed in connection with those figures.

[18]These correspond to variations on the situations shown in Figure 2.6.

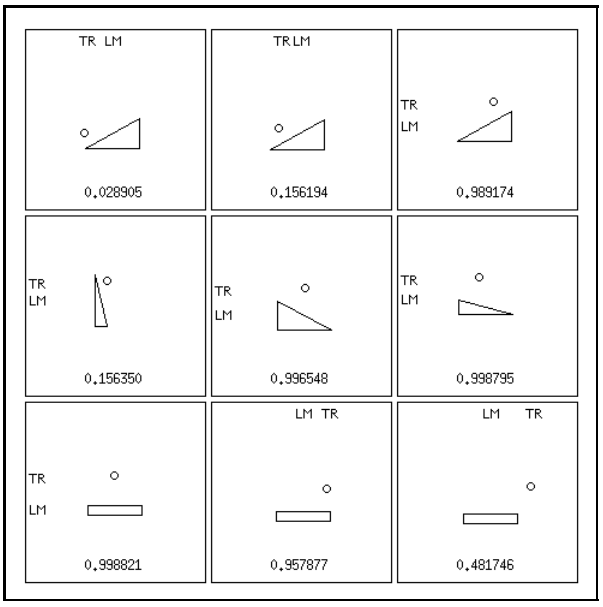


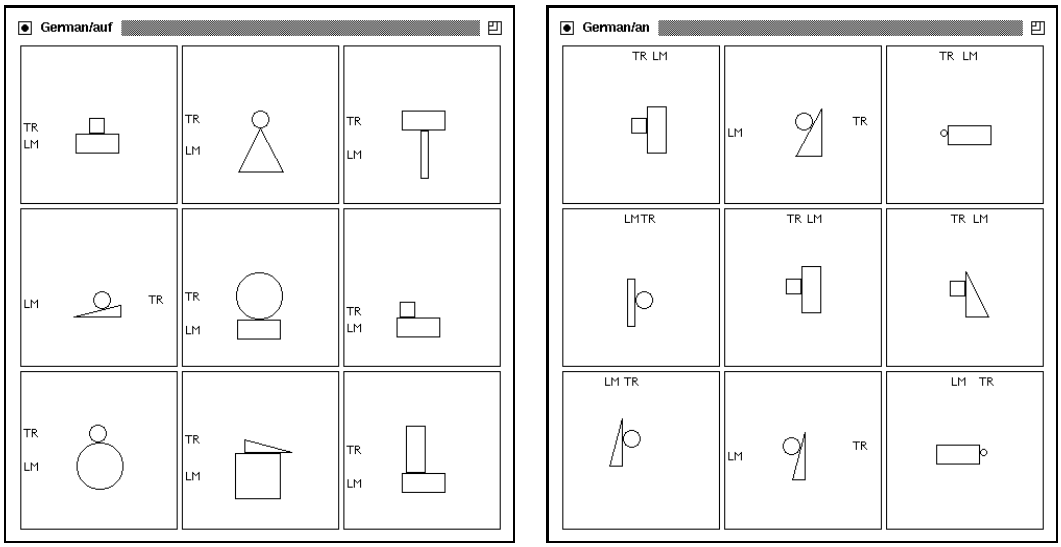
Figure 5.20: English *above*: prototype effects

to scene, more or less in accord with our intuition as to how good an example of *above* the particular scene is.

Finally, the scenes of the third row parallel the situations depicted in Figure 2.10, and illustrate the notion of prototypicality as the co-occurrence of strong evidence from several sources. In the first scene, both proximal and center-of-mass orientations are aligned with upright vertical, and the system's response is correspondingly strong. In the second scene, the proximal orientation is still perfectly aligned with upright vertical, but the center-of-mass orientation is not; the response for this somewhat less prototypical instance of *above* is slightly weaker. And finally, in the third scene neither of the two orientations is perfectly aligned with upright vertical. Here the response is significantly weaker, corresponding to our intuition that while this scene is an instance of *above*, it is a rather weak one.

5.4.2 Cross-Linguistic Variation

To illustrate the system's ability to learn spatial terms from different languages, we present its performance on terms from the three languages, other than English, which were focused on most closely in Chapter 2. These are German, Bengali, and Mixtec.



(a) (b)

Figure 5.21: Positive examples of German *auf* and *an*

German

Although German is closely related to English, there are differences in the ways in which the two languages structure space. One particular difference which we shall focus on concerns the German words *auf* and *an*. While clearly distinct in meaning, both are used in situations which would be described, in English, using *on*.

This is shown in Figure 5.21, which contains positive examples of *auf* and *an*. As one can see from the figure, *auf* denotes spatial relations in which the trajector is located on the top of the landmark, while *an* denotes relations in which the trajector is located on the side of the landmark.[19]

Figure 5.22 presents the response of the system, after it was trained for German *auf*, to positive examples of English *on*. The number at the bottom of each scene indicates how good an example of *auf* the system has judged the scene to be. As can be seen from the scenes shown here, only those instances of English *on* in which the trajector is on the top of the landmark are considered to be good examples of German *auf*. There is thus an orientational component to German *auf* which is missing from English *on*: only scenes in which the proximal and center-of-mass orientations are nearly aligned with upright vertical are taken as good examples of *auf*, whereas this

[19]This is true for the sense of *an* shown here, but *an* is highly polysemous, and can also be used in a variety of other senses. For example, if someone were seated *at* their desk, one would use *an* in German to describe the relation between the person (trajector) and the desk (landmark). For the time being, however, we will be focusing only on single senses of words.

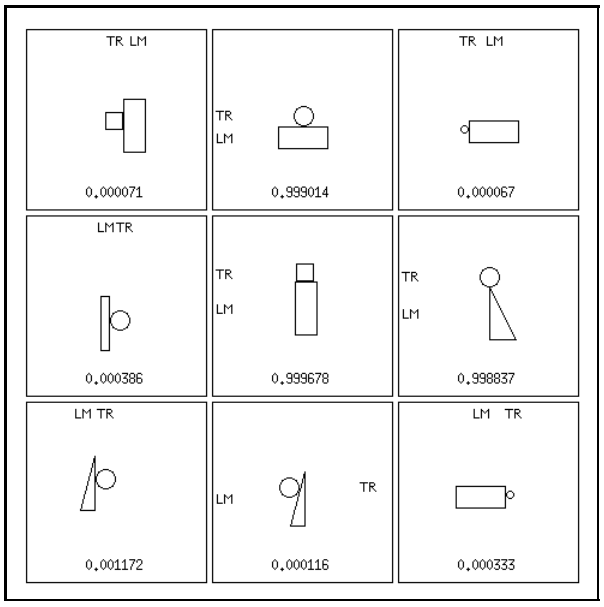


Figure 5.22: German *auf* tested on positive examples of English *on*

---

is not so for English *on*.[20]

Similarly, Figure 5.23 shows the response of the system, after being trained on German *an*, to positive examples of English *on*. The difference between German and English is again highlighted here, as it is only those scenes in which the trajector is on the side of the landmark that are considered by the system to be good examples of *an*.

Bengali

Bengali, which is, like English and German, an Indo-European language, exhibits a spatial structuring which differs from that of both English and German in at least one significant way. Figure 5.24 presents positive examples of the Bengali terms *bhethoray* and *moddhay*. These are both similar to English *in*, but there is a difference: *bhethoray* denotes complete inclusion of the trajector in the landmark, while *moddhay* denotes inclusion which is at least partial. While English *in* denotes inclusion in a graded fashion such that a trajector which is partially included in a landmark is considered to be an example of *in* to at least some degree (recall Figure 5.17), this sort of graded

[20]The non-visual feature of *support* appears to play a crucial role in both English *on* and German *auf*. This feature is not present in the system as it stands, so we must make do with the visual correlates of support, namely contact and alignment of orientational features with upright vertical. See Chapter 7 for attempts at extending the current system to handle force-related features such as support.

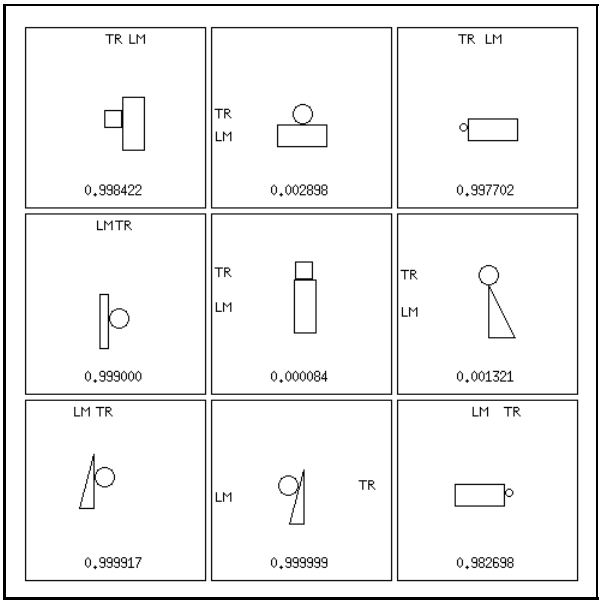
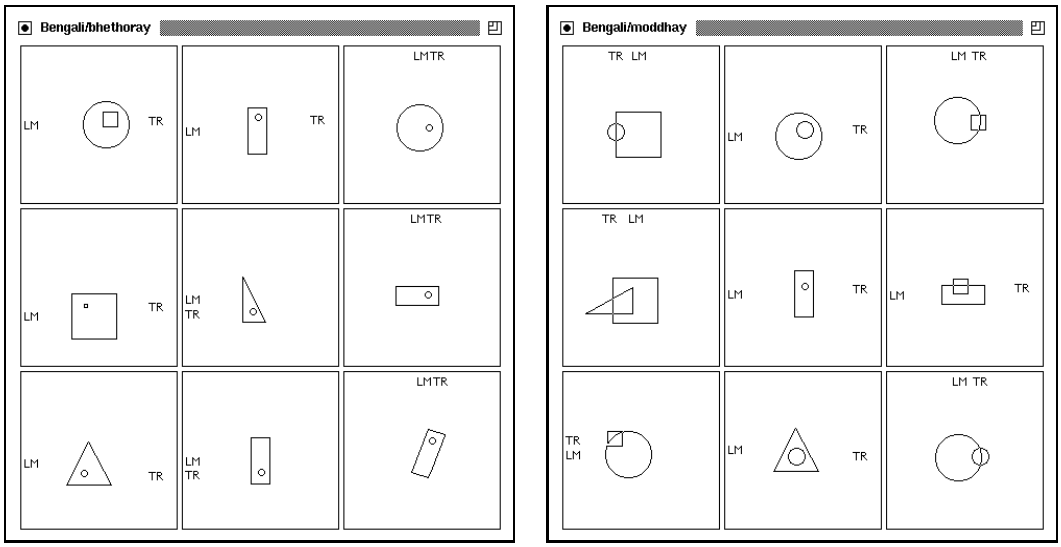


Figure 5.23: German *an* tested on positive examples of English *on*



(a) (b)

Figure 5.24: Positive examples of Bengali *bhethoray* and *moddhay*

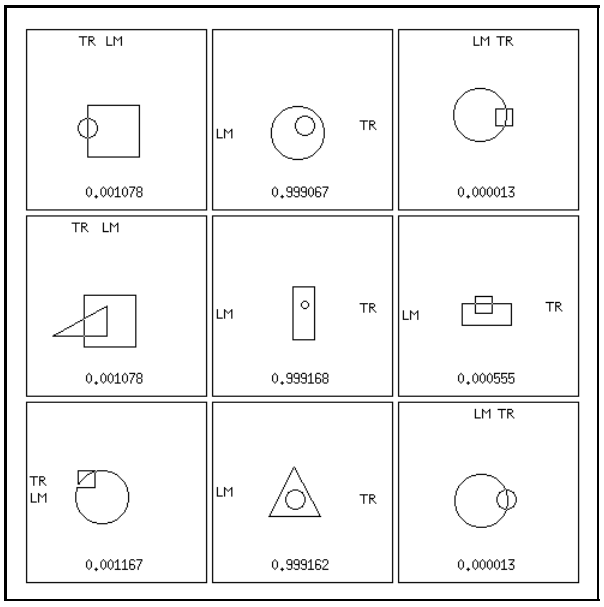


Figure 5.25: Bengali *bhethoray* tested on positive examples of Bengali *moddhay*

response is not found for either *bhethoray* or *moddhay* [Ahmad, 1992].

Figure 5.25 illustrates this phenomenon by showing the response of the system, after training on *bhethoray*, to positive examples of *moddhay*. The classification is not graded as it is for English *in*; rather, only scenes which show the trajector entirely within the landmark elicit a strong response. Scenes featuring only partial inclusion are taken to be extremely poor examples of *bhethoray*. This is of course a function of the training set; while not shown here, the negative examples for *bhethoray* included a number of scenes exhibiting partial inclusion.

Figure 5.26 presents the response of the system, after being trained for *moddhay*, on the test set that was earlier used for English *in*, shown in Figure 5.17. The system has learned to respond strongly to any scene in which the trajector is even slightly contained within the landmark.

Thus, English *in* and Bengali *bhethoray* and *moddhay* are examples of three different ways in which the system can learn to handle the feature of inclusion:

- In a graded fashion, dependent on the amount of the trajector boundary located within the landmark (English *in*).
- Discretely, such that all of the trajector must lie within the landmark (Bengali *bhethoray*). Any amount of trajectory boundary lying outside the landmark immediately disqualifies the scene.
- Discretely, such that some portion of the trajector must lie within the landmark

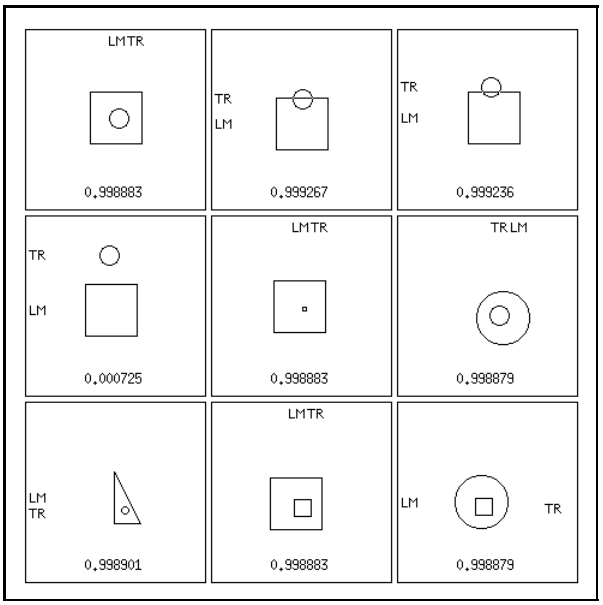
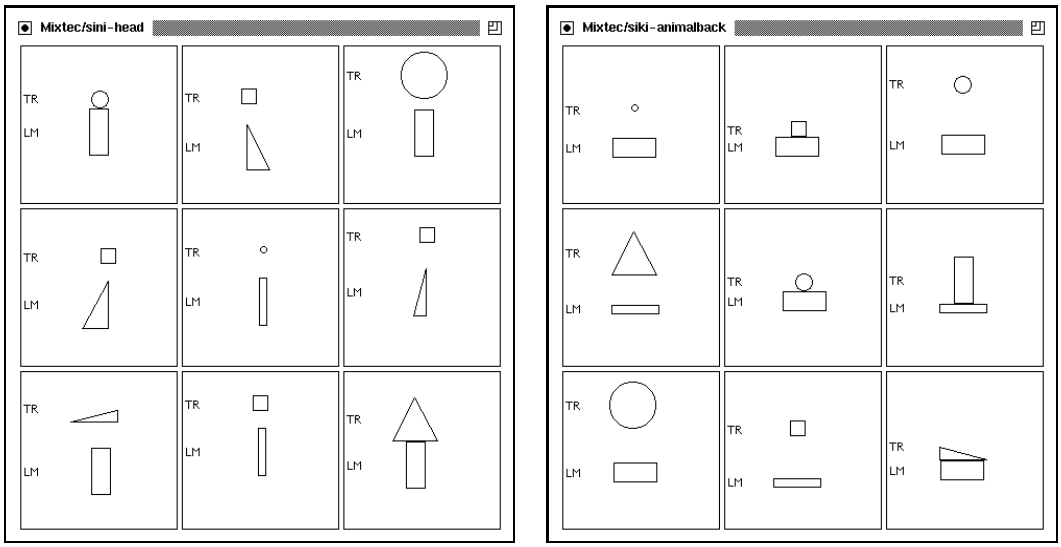


Figure 5.26: Bengali *moddhay* tested on positive examples of English *in*

(Bengali *moddhay*). Any fraction of the trajector boundary lying within the landmark suffices to cause a scene to be judged a good example of *moddhay*.

Mixtec

Mixtec, a Mexican Indian language, possesses a spatial system which differs radically from that of English. Figure 5.27 presents positive examples of the two Mixtec terms *šini* and *siki*, which translate literally to "head" and "animal back", respectively, but which are used to denote spatial relations as well as body parts. Both are used only in situations in which the trajector is either above or on top of the landmark, but the choice of which Mixtec term to use is also dependent on the orientation of the major axis of the landmark, as can be seen from the scenes in the figure above. If the landmark is vertically extended, the term *šini* ("head") is used. The motivation behind this is that if one were to view the landmark as an erect biped, for instance a human being, the trajector would be located at or near the landmark's head. And it is of course easiest to view the landmark as such when it is vertically rather than horizontally extended. On the other hand, if the landmark is horizontally extended, the term *siki* ("animal back") is used. The motivation in this case arises from the fact that if one were to view the landmark as a quadruped on all fours, the trajector would be located at or near the animal's back. It is critical to note while the motivation for the terms springs from the anatomy of bipeds and quadrupeds, knowledge of this anatomy is not required in learning the perceptually-grounded semantics for the



(a) (b)

Figure 5.27: Positive examples of Mixtec *šini* and *siki*

terms; indeed the system being presented is able to learn these Mixtec terms without any specialized knowledge of anatomy.

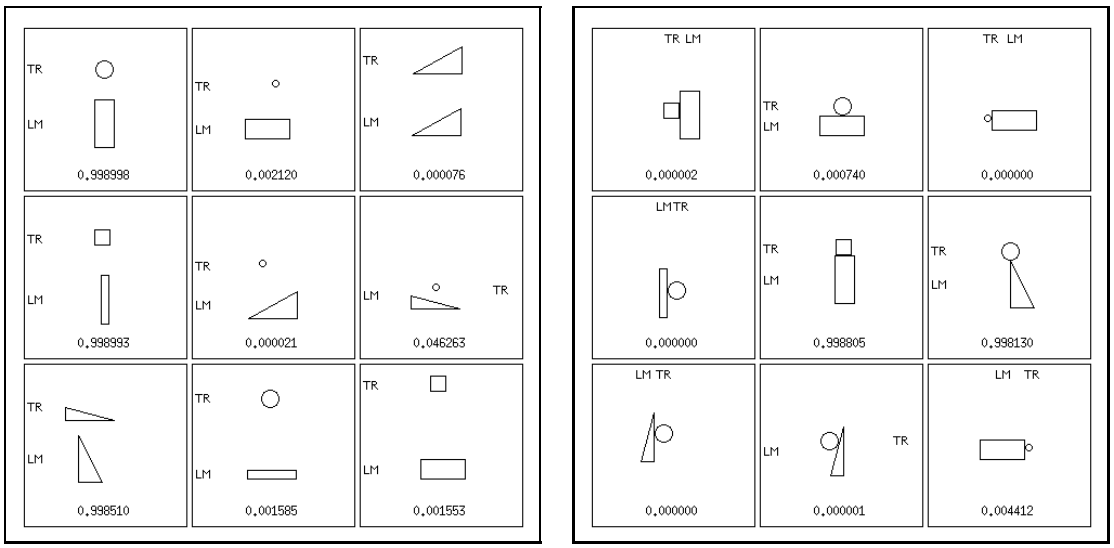
Figure 5.28 shows the results of running the system, after it has been trained on *šini*, on test sets consisting of positive examples of English *above* and *on*. Notice that contact, which is semantically significant in English, is not in Mixtec, and that the system is able to adapt itself to this: there are instances of strong responses to both scenes which exhibit contact and scenes which do not. The major axis orientation of the landmark, however, is significant.

Figure 5.29 similarly presents the results of running the system, after it was trained on *siki*, on the same two test sets. Note that contact is not significant in this case either, but that the major axis orientation of the landmark is.

Discussion

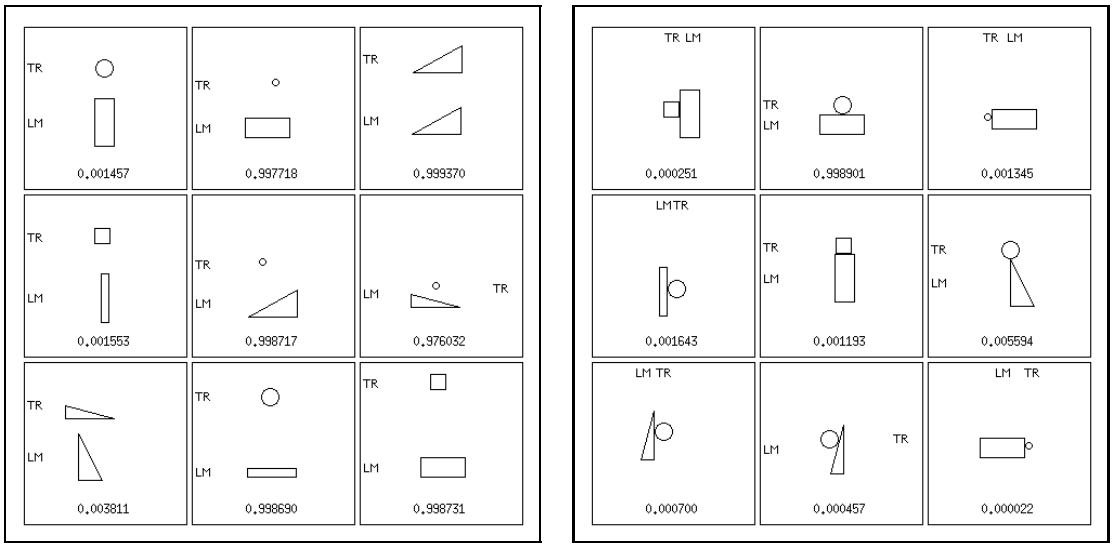
In addition to the languages discussed here, the system has been successfully trained and tested on Japanese *ue ni*, which is similar to English *above* except that it is insensitive to contact between the trajector and landmark. For example, the location of a cup *on* a table would be described in Japanese as *ue ni*, as would a lamp hanging *above* a table [Hasegawa, 1991].

So the system has been tested on a variety of languages with differing spatial systems, some – like Mixtec – radically different from that of English, and others



(a) (b)

Figure 5.28: Mixtec *šini* tested on positive examples of English *above* and *on*



(a) (b)

Figure 5.29: Mixtec *siki* tested on positive examples of English *above* and *on*

– like German – more subtly so. None of the languages tested so far has had a spatial system that the system could not learn, although the testing has been so preliminary in nature that no strong assertions regarding the system's ability to learn any language's spatial system are warranted. What we do have, however, is at least a start down the road to a system which will be able to learn the spatial system of any of a large range of natural languages.

This chapter has concerned itself exclusively with static scenes. Since much of the lexicon concerning spatial relations often concerns moving objects, we move on now to consider the case of motion.

Chapter 6

Motion

6.1 The Problem 105
6.1.1 Training Data 105
6.1.2 Assumptions 106
6.2 Outline of a Solution 108
6.3 Implementing the Solution 111
6.3.1 Three Potential Methods 112
6.3.2 Source and Path Buffers 117
6.4 Results 127
6.4.1 Back Propagation with State Units 127
6.4.2 Source and Path Buffers 130
6.5 Learning Movies without Explicit Negative Evidence . . 134
6.6 A Fictitious Spatial Concept: *in/out-of* 145
6.7 Discussion 147

Now that we have covered the learning of perceptually-grounded semantics for spatial terms in static scenes, we move on to handle motion. This chapter begins by describing the problem of learning motion-based semantics for spatial terms, and then presents an architecture which is able to perform this learning task. As we shall see, this architecture is an extension of the one just presented in Chapter 5.

Input to this modified system takes the form of simple *movies* of objects moving relative to one another, rather than static scenes of objects. Since the problem of acquiring lexical semantics for spatial terms was originally formulated, in Chapter 1, in terms of associating simple movies of moving objects with lexemes, this means we are finally addressing the task that forms the focus of this thesis, rather than some intermediate task along the way.

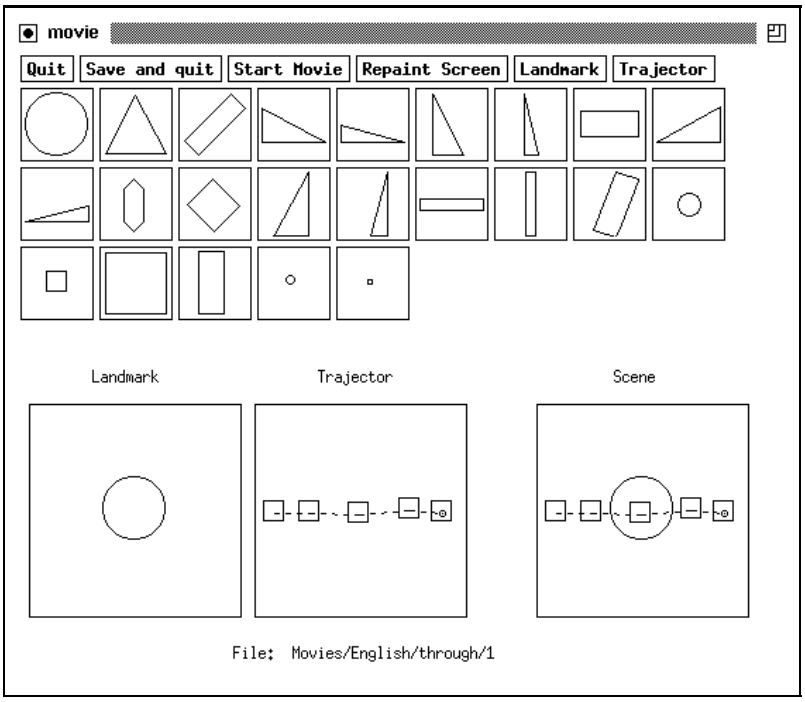


Figure 6.1: Constructing a positive example of English *through*

6.1 The Problem

6.1.1 Training Data

Positive and negative examples of particular spatial terms are constructed as illustrated in Figure 6.1, very much the same way static scenes were constructed in Chapter 5, except that the user must now indicate where the trajector is on a number of successive frames, rather than just once for a scene. Note that the landmark remains stationary throughout the movie. In the five-frame movie shown here, we have a circular landmark, and a small square trajector moving through it. The dashed lines connect successive positions of the trajector as it moves relative to the landmark, and the final frame of the movie is indicated by a tiny circle located inside the trajector. In this movie, the trajector is to the right of the landmark in the final frame.

As in Chapter 5, we at first make use here of explicit negative evidence for each spatial term learned, temporarily leaving aside the issue of learning in the absence of such explicit negative evidence. However, §6.5 describes an experiment in which motion-based semantics for spatial terms are learned without explicit negative evidence, using the methods of Chapter 4.

Figure 6.2 presents a set of positive examples of English *in*, constructed using the movie editor of Figure 6.1. This is a subset of the complete set of positives used in

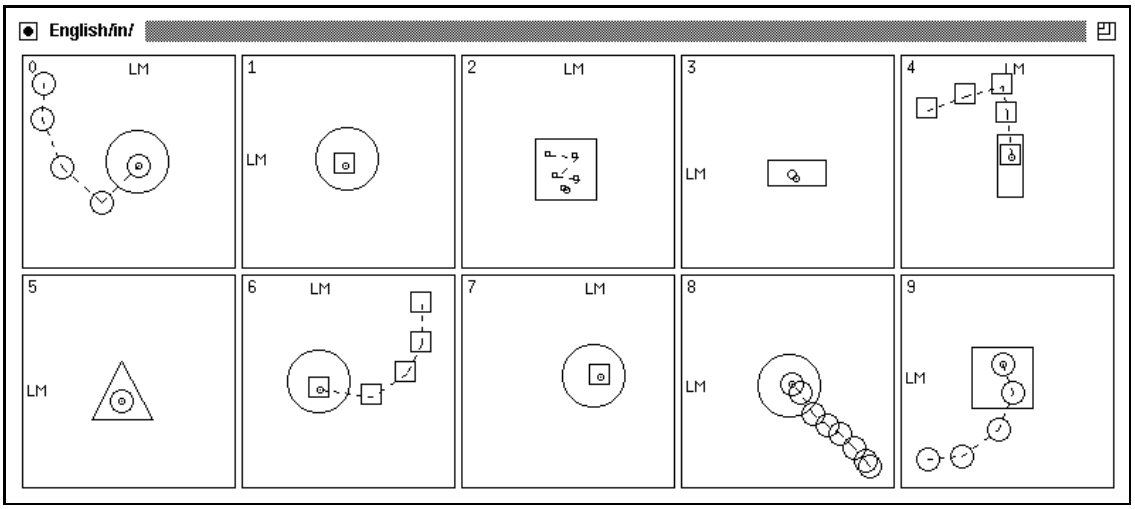


Figure 6.2: Positive examples of English *in*

training the system for *in*. Note that two different senses of the word are represented here:

- Location, either static or moving, entirely within the landmark interior (movies 1, 2, 3, 5, and 7).
- Motion into the landmark interior (movies 0, 4, 6, 8, and 9).

Static scenes are represented here by movies in which the trajector remains in the same position for all frames of the movie, i.e. by movies in which "nothing happens".

Figure 6.3 presents negative examples of English *in*. This is a subset of the negatives used in training the system.

The task the system must perform is learning to associate movies of this sort with appropriate spatial terms from some language.

6.1.2 Assumptions

A number of assumptions have been made regarding the nature of the training sets and the training process. These are spelled out explicitly here.

Arbitrary Length

We assume that movies viewed by the system, from either training sets or test sets, may be of arbitrary length. Notice for example that while most of the movies in Figure 6.2 and Figure 6.3 are five frames in length, there are two which are eight

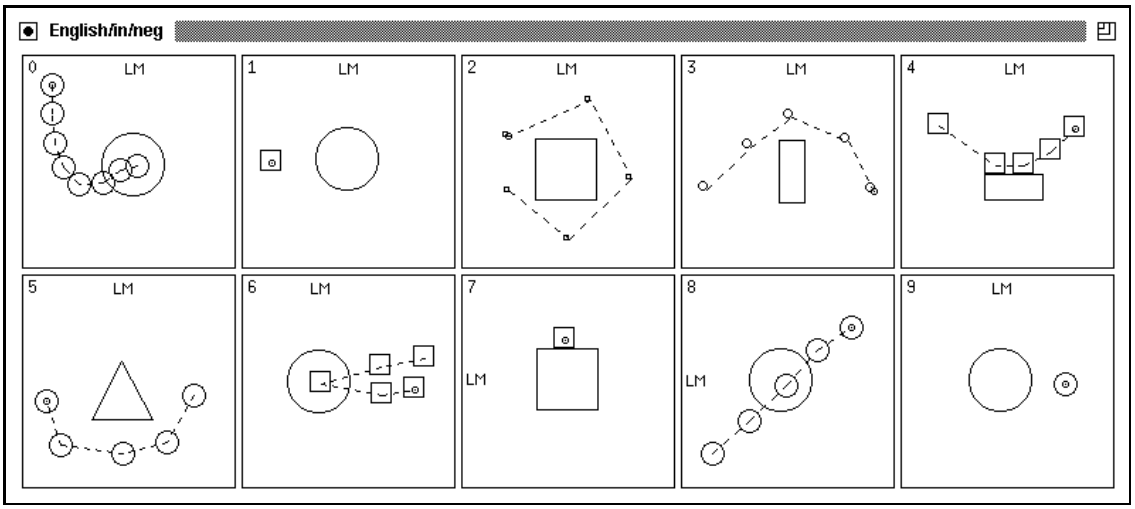


Figure 6.3: Negative examples of English *in*

frames each (number 8 in Figure 6.2 and number 0 in Figure 6.3).[1] Our own intuitive judgment of whether or not the trajector went *in* the landmark is unaffected by the number of frames it takes to represent the event; the system should similarly be insensitive to movie length.

Response at the Final Time Step

We take the system's response at the final time step of the movie, i.e. after viewing the final frame, to be its response to the movie as a whole. Note that this assumption and the one above, taken together, imply that the system must respond at every time step, indicating how good an example of the spatial term being learned it judges that portion of the movie viewed so far to be. This is so since, following the first assumption, movies may be of arbitrary length, therefore each frame could be the final one, i.e. the one after which a judgment regarding the movie as a whole will be required. So the system must give a "running" judgment of the movie so far, at each time step.

Training at the Final Time Step

We assume that the system receives a training signal, indicating whether or not the movie just viewed was an example of the spatial term being learned, only at the end of the movie. Note that the training data only specifies what the movie as a whole is labeled as (e.g. a positive example of *through*), and does not say anything about

[1] The "static" movies in these figures contain five identical frames each.

the individual frames which make up the movie. Since it is the system's response at the last time step that we take to be its response to the movie overall, it is this response which is compared to the desired response to the whole movie. These two are compared using the standard error computation (recall Equation 3.2), and the error is back-propagated.

Now that the assumptions underlying the general approach taken here have been detailed, we move on to an outline of a solution to the problem. The details of the solution will be filled in in later sections.

6.2 Outline of a Solution

Figure 6.4 presents, in very broad outline, an overview of the architecture used here for motion. Successive frames of a movie are provided as input to a *frame analyzer (FA)*, which produces, for each frame, a representation of that frame in terms of a set of derived static features. This representation of the current frame is placed in the buffer labeled *Current*, and from there, it is passed on to the *motion module*. The motion module views the representations of successive frames in the *Current* buffer of the frame analyzer, and, if the system has been trained for a specific spatial term, outputs a judgment of how good an example of that term the movie is, up to and including the current frame.

The frame analyzer is illustrated in Figure 6.5. As should be apparent from the figure, the frame analyzer is of essentially the same form as the architecture presented in Chapter 5, for classifying static scenes. There are only two differences: one is that in this version, the frame analyzer does not have an output node at the top. Rather, the output from the *Current* buffer feeds into the motion module.

The other difference, which is not evident in the figure but which plays a critical role in motion processing, is that a new directional primitive has been added: the direction of motion of the trajector at each point in time. For a given time step, this is defined as the orientation of the directed line segment connecting the center of mass of the trajector in the preceding frame with its center of mass in the current frame, as illustrated in Figure 6.6. Notice that since there is no frame before the first frame, the direction of motion for the first time step is defined to be the same as that at the second time step. The direction of motion is currently used both as a relational orientation and as a reference orientation.

It is critical to note that the frame analyzer is trained along with the motion module. Thus, the system must determine which static features to extract from each frame, and at the same time, must determine how to combine individual movie frame representations over time. Thus, there is learning in both the frame analyzer and the motion module, simultaneously.

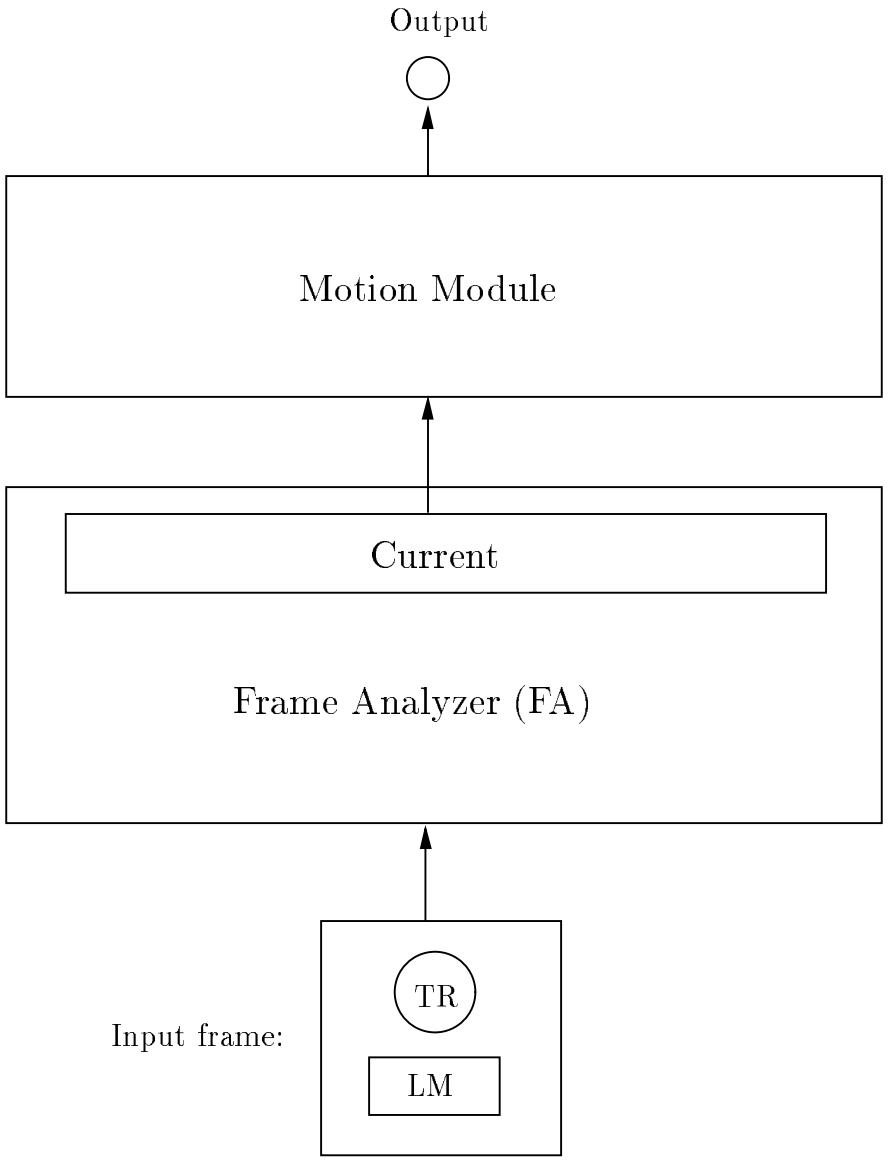


Figure 6.4: Architecture outline

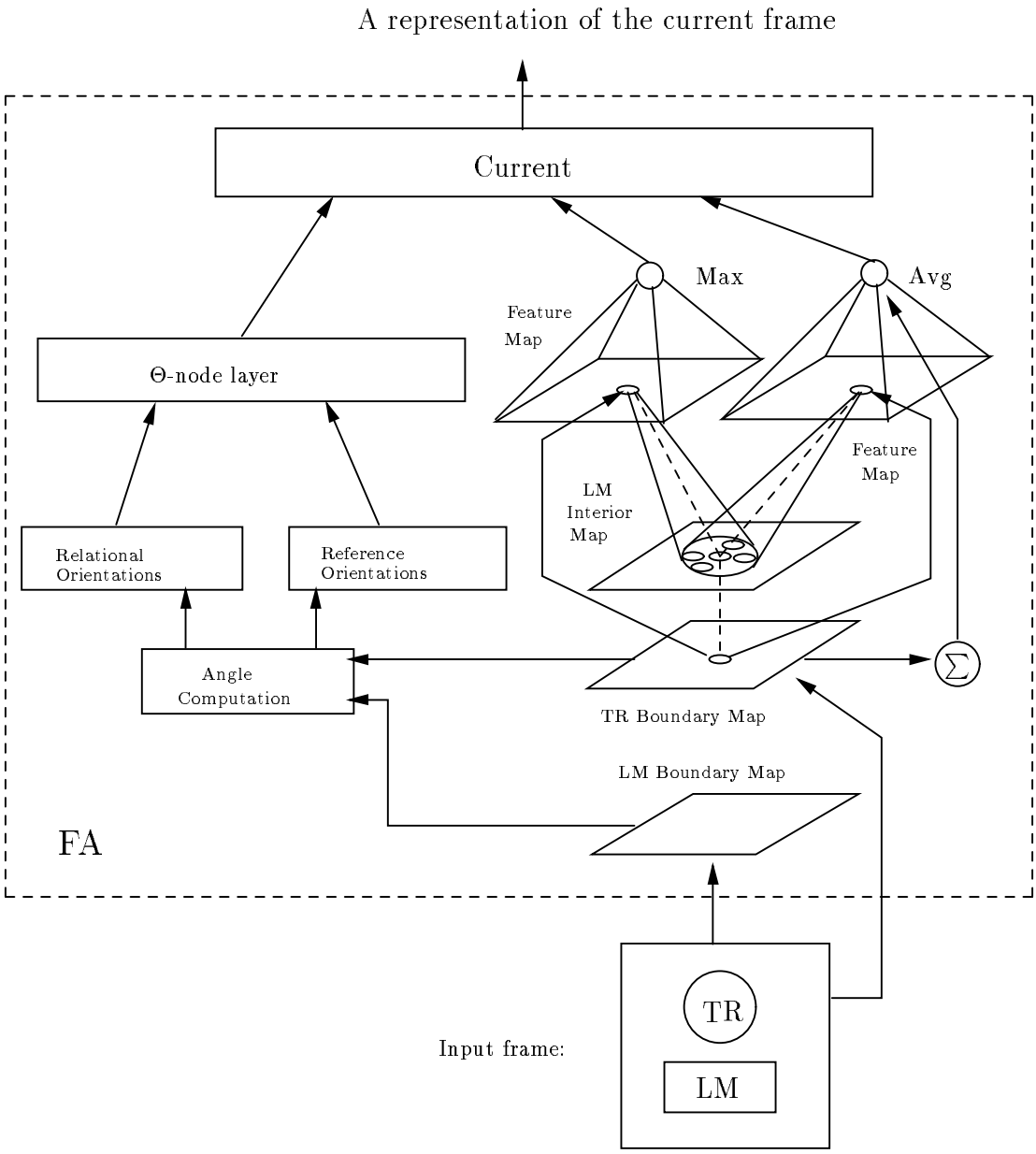


Figure 6.5: Frame analyzer

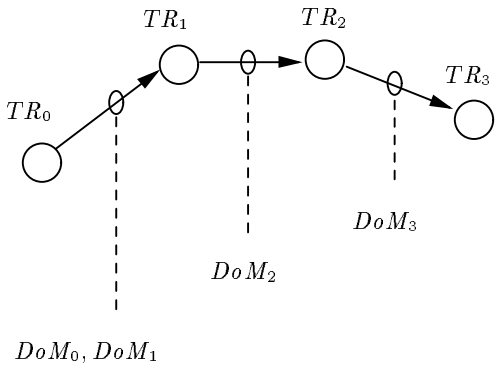


Figure 6.6: Direction of motion (DoM) in a 4-frame movie

6.3 Implementing the Solution

Half of the architecture has already been specified, of course: the details of the frame analyzer were covered in Chapter 5, and the addition of a simple motion-based directional primitive was noted above. In this section, we describe the implementation of the motion module, and the manner in which the two modules of the system work together.

As we have seen, the motion module receives representations of successive frames of the movie, and delivers a classification of the movie as a whole at the end of the movie. Since the module receives as input a sequence of static frame representations, this is essentially a sequence recognition task.

In Chapter 3, we reviewed three standard connectionist methods for learning sequences: *back propagation through time, time-delay neural networks,* and *back propagation with state units.* Each of these standard methods provides at least a potential design framework for the motion module.

In actuality, none of these three is the method eventually adopted here, although a variation on the general theme of back propagation with state units was examined. Before proceeding to cover the actual design in detail, we begin by outlining the reasons why none of the three traditional candidates was chosen.

6.3.1 Three Potential Methods

Back Propagation Through Time

Recall from Chapter 3 that back propagation through time amounts to "unrolling" a recurrent network over time, so that there is one layer in the resulting network for each time step in the sequence being learned. Figure 6.7 illustrates how this would work for the movie-learning task we are faced with here. Figure 6.7(a) shows an overview of a simple architecture, along the lines outlined above, in which the motion module is simply a layer of hidden units with completely interconnected recurrent connections w. In this diagram, the frame analyzer is denoted by the box labeled *(FA)*, which contains the *Current* buffer. Successive frames are provided as input to the FA, and from there, representations of the frames are fed as input to the single hidden layer of the motion module. Since this hidden layer has connections back to itself, one (rather naive) hope might be that these connections, under training, will adapt themselves so that the resulting representations in the motion module capture precisely those aspects of the movie which are relevant to classifying the movie correctly.

Even if this were the case, however, there would remain several solid reasons to avoid this approach. Figure 6.7(b) presents the network after it has been "unrolled", preparatory to training on a movie of length t. Note that the frame analyzer (FA) has to be replicated, once for each frame in the movie, so that features can be extracted from each frame, providing input to the corresponding motion module layer. Corresponding weights in these FAs will have to be tied, just as the weights w connecting unrolled layers are tied, as is usually done under back propagation through time. Thus, given this replication of frame analyzers, on top of the replication of recurrent layers and weights which is usual under back propagation through time, this approach is quite expensive in terms of memory consumption.

Another drawback, as we mentioned earlier in Chapter 3, is the fact that tall networks with tied weights between layers, such as would result when trying to learn long sequences, tend to be difficult to train [Mozer, 1988; Pollack, 1990a]. This is a point against the method of back propagation through time in general. Since we assume here that movies may be of arbitrary length, this is potentially a serious problem with this method.

Time-Delay Neural Networks

Time-delay neural networks, as presented in Chapter 3, handle sequences by accepting as input, at a given point in time, the current input together with the inputs from the last k time steps, for some fixed k. Figure 6.8 presents a time-delay neural network functioning as the motion module in the overall design; here, $k = 4$, so the network views the contents of the frame analyzer *Current* buffer at the current time step, and the last four time steps as well. The resulting network is quite shallow and relatively

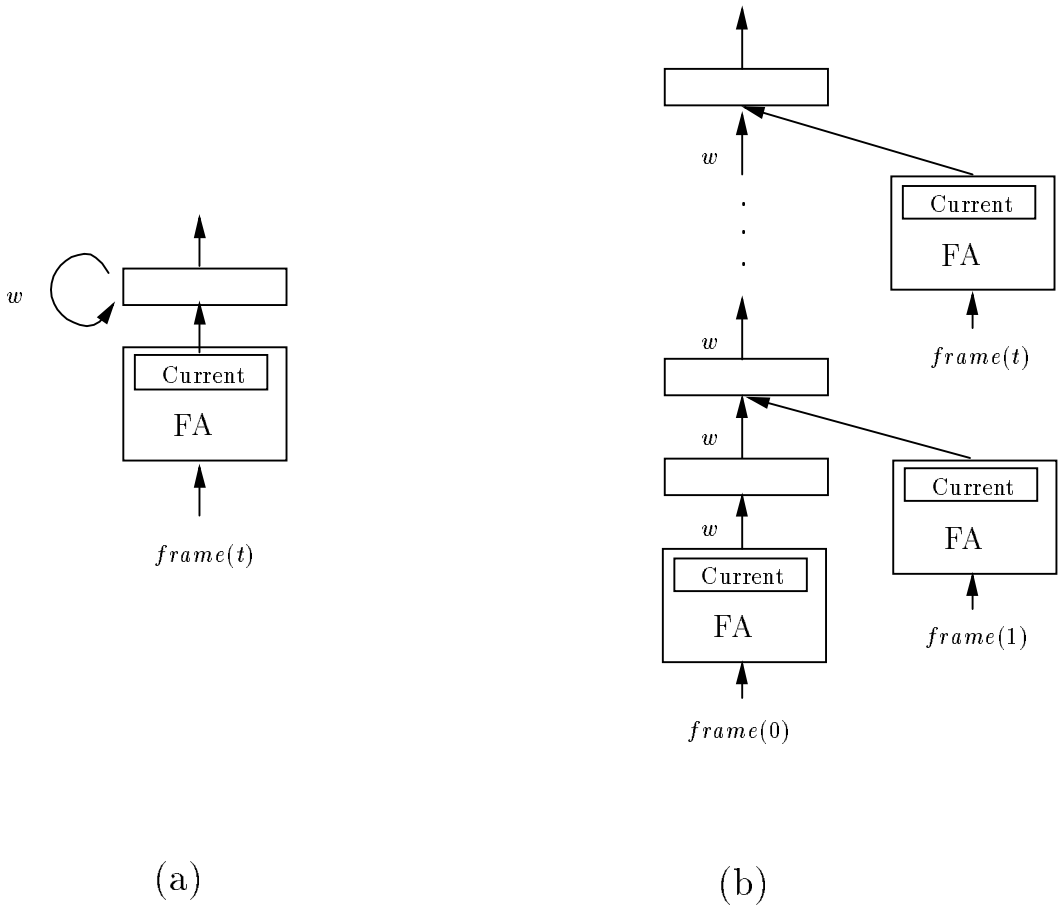


Figure 6.7: Motion using back propagation through time

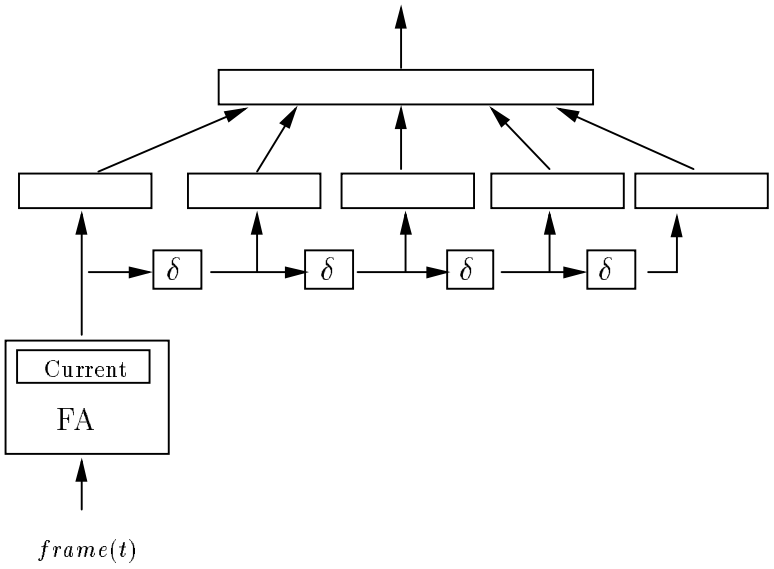


Figure 6.8: Motion using a time-delay neural network

---

small, so the problems associated with back propagation through time are avoided. However, one obvious drawback to solutions of this sort is the fact that only $k+1$ (here, five) frames of the movie are viewed at a given time. This means that at the end of viewing a movie, only the last $k+1$ frames will provide input to the network; any significant earlier frames will be overlooked. If we knew ahead of time that all movies would be of a particular length, this might be a reasonable approach, but as we allow movies of arbitrary length, this is a serious problem.

Back Propagation with State Units

One standard approach to sequence learning which was tested was the application of back propagation with state units, specifically, the Elman architecture [Elman, 1988], and a variant thereof. Figure 6.9 illustrates two architectures of this sort. Figure 6.9(a) presents a straightforward application of the Elman architecture to the task at hand. Here, the contents of the hidden layer at each time step are copied down into the state buffer to provide part of the input for the next time step. Thus, the hidden layer at time step t accepts as input $s(t-1)$ and the output of the frame analyzer at time step t, and produces $s(t)$.

Since the system as a whole is trained only on the last frame of the movie, it is exceedingly unlikely that this architecture will learn correctly. This is so since

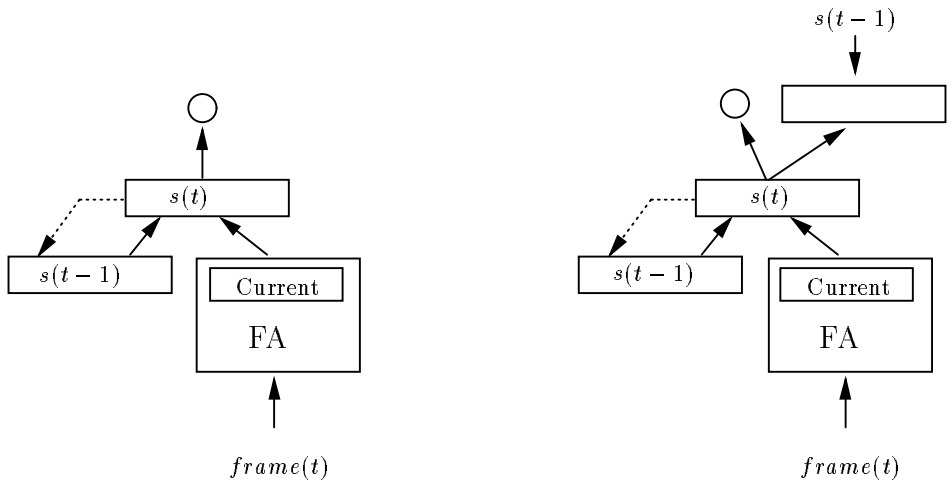


Figure 6.9: Motion using back propagation with state units

there is no error feedback during the course of the movie to force the representations formed in the hidden layer to encode anything about the movie that will be of use in its eventual classification at the last time step. Networks of this sort have proven successful under training at each time step.

The architecture shown in Figure 6.9(b) is a variation on the standard Elman architecture which addresses this problem, by introducing a means of training on each time step when there is only a single training signal at the end of the movie. This is identical to the architecture shown in (a), except that the system is trained, at every time step *except the first and the last*, to output the contents of the hidden layer at the previous time step. That is, at time t, when the contents of the hidden layer are $s(t)$, the system must produce $s(t-1)$ on its auxiliary outputs, shown as a box in the output layer (the output node for the concept being learned is indicated by a small circle). The idea behind this approach is that if the system successfully performs this learning task, then $s(t)$ will contain a representation of $s(t-1)$, $s(t-1)$ will contain a representation of $s(t-2)$, and so forth, in the sense that the network will always be able to produce $s(t-1)$ from $s(t)$, and $s(t-2)$ from $s(t-1)$. This is potentially a worthwhile state of affairs because at time t the hidden layer will contain a representation of the entire movie up to and including frame t, since the hidden layer representation for each frame has encoded in it all the information which was encoded in the previous one. Thus, since the movie as a whole is encoded in the hidden layer, the system should be able to use the hidden layer representation to arrive at a judgment regarding the entire movie, at the last time step. This method, similar in flavor to Jordan Pollack's *recursive auto-associative memory* [Pollack, 1988; Pollack, 1990b], has been used with some success for sequence learning in the domain of natural language sentence comprehension [Miikkaulainen, 1991], and in inducing finite state machines [Maskara and Noetzel, 1992].

Regardless of the performance of this mechanism once it has learned (which we shall be covering in §6.4 below), it has one drawback, which it shares with all architectures trained using the Elman approach: as discussed in Chapter 3, the Elman approach does not perform true gradient descent. This means that it will not tend to learn effectively when trained using second-order methods which take large steps in weight space, and which therefore usually yield quick convergence. Rather, it must be trained using "ordinary" back propagation or some similarly cautious variant, which will make training a relatively time-consuming process. This situation is of course avoided if we compute the true gradient; not surprisingly, informal studies indicate that second-order methods work well on recurrent networks when following the true gradient [Beutter, 1992]. However, the computation of the true gradient is in itself an expensive operation.

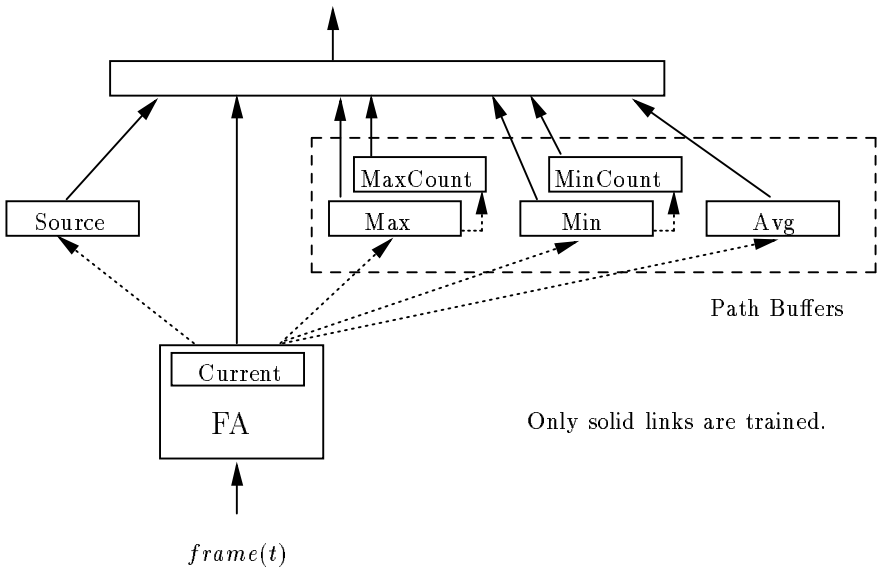


Figure 6.10: Motion using source and path buffers

6.3.2 Source and Path Buffers

Architecture

Figure 6.10 illustrates the approach finally adopted, in handling sequences of movie frames over time. While this is structurally more complex than the Elman-style architecture just covered, it is more suited to the task at hand, in that the built-in structures were designed specifically to capture linguistically relevant aspects of the movie as a whole. Like the Elman-style architectures we have been discussing, this mechanism does not follow the true gradient (see the section on training, below); nonetheless, empirical testing reveals that often this architecture responds well to second-order methods.

We now focus on the motivation for the built-in structures, just what it is about the movie that they capture, and how they do it.

The central motivation for the design is borrowed from cognitive linguistics. [Lakoff, 1987] notes that linguistically, events are often structured in terms of three components: a *source*, or point of origin for motion, a *path*, or trajectory traced through space by the motion, and a *destination*, the endpoint of motion. Lakoff's point is that lexemes which refer to motion-based events are often easily characterizable in terms of these three elements. Given this, it makes sense to build into the system structures

which capture this intuitively natural-seeming decomposition of the event.

Before moving on to present the details of this architecture, however, it is worthwhile noting briefly that there is also a non-linguistic source of motivation for the conceptual decomposition of motion events into source, path, and destination: this is the visual phenomenon of *apparent motion* [Kolers, 1972]. Human subjects, when presented with an object displayed briefly at one position in the visual field, and then with another copy of the object displayed at another position, often perceive the object moving smoothly from the first point to the second. In the terms we are using here, given the source and then the destination, the subjects perceptually infer the path. Thus, source and destination serve as anchor-points of sorts, from which the path is computed. Connectionist research in modeling this phenomenon [Olson, 1989] has focused on the computation of explicit trajectories given starting and ending points.

Returning now to the architecture at hand, it is easy to see how one might represent the elements of source and destination. Recall that at each time step, the frame analyzer produces, in the *Current* buffer, a representation of the static features present in the current frame. So to represent the role of source, all we need is a copy of what the *Current* buffer contained at the first time step of the movie. The buffer labeled *Source* in Figure 6.10 in fact contains exactly that, and remains unchanged throughout the course of the movie. The link connecting the *Current* buffer to the *Source* buffer is shown as a dotted line, indicating that this link is not trained; rather, it performs a simple copy operation, and only on the first time step of the movie.

Representing the linguistic role of destination is even simpler: at the last time step of the movie, the contents of the *Current* buffer itself will be a representation of the destination.[2]

The remaining problem, then, is representing the path, or trajectory, which the trajector traces out as it moves. This is done by the part of the network which is shown in dashed outline in Figure 6.10, labeled *Path Buffers*. (Note that the links connecting the *Current* buffer to these buffers are shown as dotted lines, indicating that they are not trained.) We will shortly turn to describing the function of the buffers that make up this part of the network.[3]

The basic assumption behind the design of this part of the network is that it will suffice, for our purposes here, to track each of the derived features detected by the frame analyzer, and to note, for each such feature, the minimum, maximum, and average activations attained by the unit representing that feature over the course of the movie, and in the case of the minimum and maximum, how often these minimum

[2]The way in which source and destination are handled is thus somewhat reminiscent of time-delay neural networks, in that input to the network (or here, the motion module) at different times is fed into different buffers within the network.

[3]This is of course not the only conceivable means of representing paths – see for example the discussion of reified paths in §7.1.4 for an alternative method.

(a) (b)

Figure 6.11: Motivating the path buffers (see text)

---

and maximum values were reached. We try now to briefly motivate this assumption.

Consider Figure 6.11. The diagram in (a) presents a good example of English *through*. Note that it is critical to the movie's classification as *through* that the trajector be inside the landmark at some point during the middle of the movie, and at neither the beginning nor the end. However, a number of factors are irrelevant to its classification as *through*; these include the fraction of the trajectory which was covered before entering the landmark, the fraction of the trajectory spent inside the landmark, and so on. We would like to be able to pick up on the simple fact that the trajector was inside the landmark *at some point* during the trajectory. Detecting the maximum and minimum values of features over the course of the movie corresponds to this notion of caring only about the occurrence versus non-occurrence of a feature during the movie taken as a whole. This is so since if there is a unit which detects, say, inclusion of the trajector in the landmark in a single frame, we can tell at the end of the movie if the trajector ever entered the landmark during the movie by examining the maximum value ever attained by that inclusion-detecting unit, so the system will be able to discriminate between the two cases.

It is not quite true, however, that only the occurrence or non-occurrence of a feature over the course of the movie is significant for our example spatial term, English *through*. Consider Figure 6.11(b). This presents what is at best a weak example of *through*. Nonetheless, it is a motion event in which the trajector begins outside the landmark, ends up outside the landmark, and is inside the landmark at some point along the way. Thus, under the scenario outlined above, the minimum and maximum values for a postulated inclusion-detecting unit would be the same for this movie as for the one shown in (a). As we would like our representations of the movies to allow us to differentiate them, since one is an excellent example of *through* and the other not, we also keep track of the number of times the activation of a feature-detecting

unit (e.g. an inclusion-detecting unit) reaches its maximum and minimum values. In (a), the trajector is outside the landmark for two separate stretches of time, and inside it once, whereas in (b) it is outside it for three separate stretches, and inside it for two. This will be reflected in the number of times the minimum and maximum values are reached for the inclusion-detecting unit.

In addition to these quantities, we also keep track of the average activation of each feature-detecting unit over time.

This overview has provided just a rough sketch of the approach. We shall now move on to the implementation of these ideas, and in §6.4, we shall see the results of taking this approach to handling motion.

The path traversed by the trajector is represented using a set of buffers each of which contains the same number of units as the *Current* buffer of the frame analyzer, and each of which computes some function (either the minimum, maximum, or average) of the values seen in the units of *Current* buffer over the course of the movie. The units in the *Current* buffer will learn to detect static features (such as inclusion, from our example above); thus, these are the the feature-detecting units referred to above. There is a one-to-one correspondence between units in the *Current* buffer and units in each of these path buffers, such that a given unit in one of the path buffers will compute some function of the values seen in the corresponding *Current* buffer unit over the course of a movie. The functions computed by units in these buffers are as follows:

- Units in the path buffer labeled *Max* record the maximum value attained by the corresponding *Current* buffer unit over the course of the movie
- Units in the path buffer labeled *MaxCount* record the number of times that maximum value was attained
- Units in the path buffer labeled *Min* record the minimum value attained by the corresponding *Current* buffer unit over the course of the movie
- Units in the path buffer labeled *MinCount* record the number of times that minimum value was attained
- Units in the path buffer labeled *Avg* record the average value of the corresponding *Current* buffer unit over the course of the movie

To make this somewhat more concrete, consider Figure 6.12. This figure illustrates as an example the path buffer each unit of which computes the maximum value attained by the corresponding *Current* buffer unit during the movie. So the *Max* buffer records the maximum value ever attained by each of the derived features learned by the frame

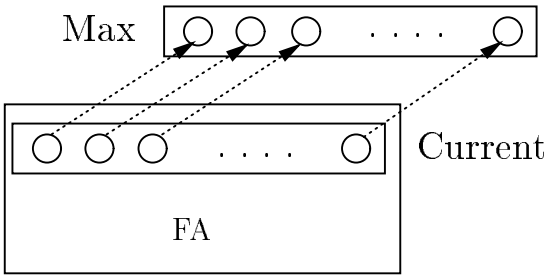


Figure 6.12: The *Max* buffer

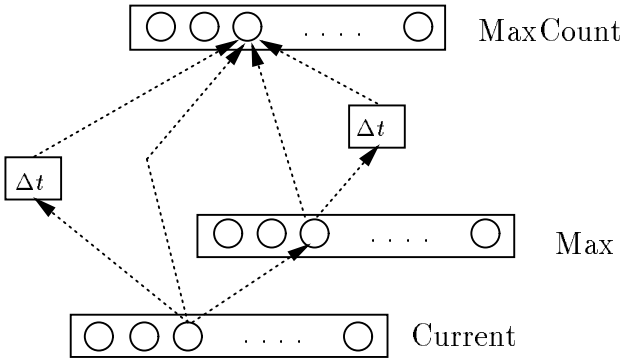


Figure 6.13: The *Max* and *MaxCount* buffers

analyzer. More specifically, the function for node i in the *Max* buffer at time step t is

$$Max_i(t) = \begin{cases} Current_i(t) & \text{if } t = 0 \\ Current_i(t) & \text{if } Current_i(t) > Max_i(t-1) \\ Max_i(t-1) & \text{otherwise} \end{cases} \tag{6.1}$$

The *Min* buffer similarly records the minimum value attained by *Current* buffer nodes over the course of a movie.

Figure 6.13 presents the *MaxCount* buffer and its relation to the *Current* and *Max* buffers. Each unit in the *MaxCount* buffer keeps a representation of the number of times the corresponding unit in the *Current* buffer has hit what is currently its maximum value so far. A value of 0 for a counter unit indicates that the corresponding feature has reached its maximum once, a value of 0.1 indicates it has reached it twice, and so on. To this end, unit i in the *MaxCount* buffer is initialized to 0 at the

beginning of each movie. After that, the function for the node at time step t is

$$MaxCount_i(t) = \begin{cases} 0 & \text{if } Current_i(t) > Max_i(t) \\ MaxCount_i(t-1) + 0.1 & \text{if } \left[\begin{array}{l} Current_i(t) \approx Max_i(t) \wedge \\ Current_i(t-1) < Max_i(t) \wedge \\ Max_i(t) \approx Max_i(t-1) \end{array} \right] \\ MaxCount_i(t-1) & \text{otherwise} \end{cases} \tag{6.2}$$

where "$\approx$" indicates equality within some externally specified tolerance ϵ. The idea here is that each time a new maximum is reached for a feature unit, the corresponding counter unit is reset to 0, and each time the old maximum is reached again, the counter is incremented, by 0.1. The *MinCount* counter buffer operates in an analogous fashion.

These five buffers taken together constitute a representation of the path taken by the trajector. Taken together with the *Source* and *Current* buffers, they provide a representation of the trajectory as a whole in the tripartite source-path-destination scheme suggested by work in cognitive linguistics and by the phenomenon of apparent motion. While the structures presented here are somewhat complex, they are well suited to the task at hand, as we shall see when covering the results of learning in §6.4.

To conclude our overview of the architecture, we note that the top hidden layer shown in Figure 6.10 accepts input from the *Src* buffer, the *Current* buffer, and the five path buffers, and yields an overall response.

Training

Referring back to Figure 6.10, note that the links connecting the *Current* buffer of the frame analyzer to the *Source* and path buffers are indicated by dotted lines; this means that they are not trained, and that error is not back-propagated along them. All the solid links shown in the diagram, and the weights inside the frame analyzer (recall Figure 6.5), will be trained, however.

Since error back-propagation occurs only along solid links, the frame analyzer will receive error signals only via the solid link which connects it to the topmost hidden layer in the architecture. And since there is a training signal only on the last time step, error will be back-propagated along this link into the frame analyzer only on the last time step. This has a profound effect on the kinds of events the network can learn to recognize. The frame analyzer will learn to detect only static features which are present in the final frame of some movie, since it is only on the final frame that error feedback is received. In addition, it will detect only those static features in the final frame whose presence there is directly relevant to the correct classification of the movie as a whole. As we shall see, this is not a serious problem for our purposes here, but it is important to realize what the limitations are of the architecture being proposed.

Note that this form of training constitutes a deviation from true gradient descent, since under true gradient descent, each frame of the movie would contribute to the gradient which is followed in the frame analyzer. As it is, only the final frame does. This approximation provides us with a means to train the network without having to compute the actual gradient, though this does result in the limitations pointed out above. Since calculating the true gradient is an expensive process, particularly for movies of many frames, this approach yields a somewhat limited, but more efficient, alternative. The limitations do not appear to be a hindrance for the task at hand.

Consider the training examples for English *in* shown in Figure 6.2 and Figure 6.3. Learning *in* should be straightforward, since the relevant static feature, namely inclusion, is present in the last frame of all the positive instances, and absent in the last frame of all the negative instances. Once the static feature of inclusion has been learned, the system can learn to track it over time using the source and path buffers, to arrive at a classification for each movie.

Now consider Figure 6.14 and Figure 6.15, which present positive and negative examples of English *through*, respectively. These are the movies used in training the network. Here, the static feature of inclusion is again relevant, but it appears only in mid-event for the positive instances, never at the final frame. In addition, it does not consistently appear in the final frame of the set of negatives shown here, in fact appearing in the final frame for only three movies out of a total of 48.[4] This situation is clearly more problematic than the one we saw regarding *in*: we cannot expect the error feedback to cause the network to detect the feature of inclusion, as it is simply not there to be detected at the last time step most of the time.[5]

The reason this is not a serious problem here is that we can easily get around it by training for several spatial terms simultaneously, as shown in Figure 6.16. Each positive or negative instance for one term is taken as a don't-care for the other.[6] The idea here is that in the process of learning *in*, the network will learn to detect inclusion, and that can then be used in learning *through*. We shall see examples of learning in this manner in §6.4 below. In addition, we shall examine the case of learning more than two terms together, and learning several terms together without the use of explicit negative evidence.

In Chapter 4, learning concepts together as a set was critical for the purpose of learning without explicit negative evidence. We now have another reason for learning them as a set: the static features which are learned for one term can be used in learning others, which might not have been learnable if the network had not been

[4]These are movies 9, 20, and 22 of Figure 6.15.

[5]Note that movie 24 in Figure 6.15 is an example of the sort of "false *through*" which was discussed earlier, and was illustrated in Figure 6.11(b). Examples of this sort provided the motivation for the count buffers which were incorporated into the architecture.

[6]Recall that in Chapter 4 we took a positive instance of one term to be an implicit negative instance for all others. We use don't-cares here because we are temporarily leaving aside the issue of learning without explicit negative evidence.

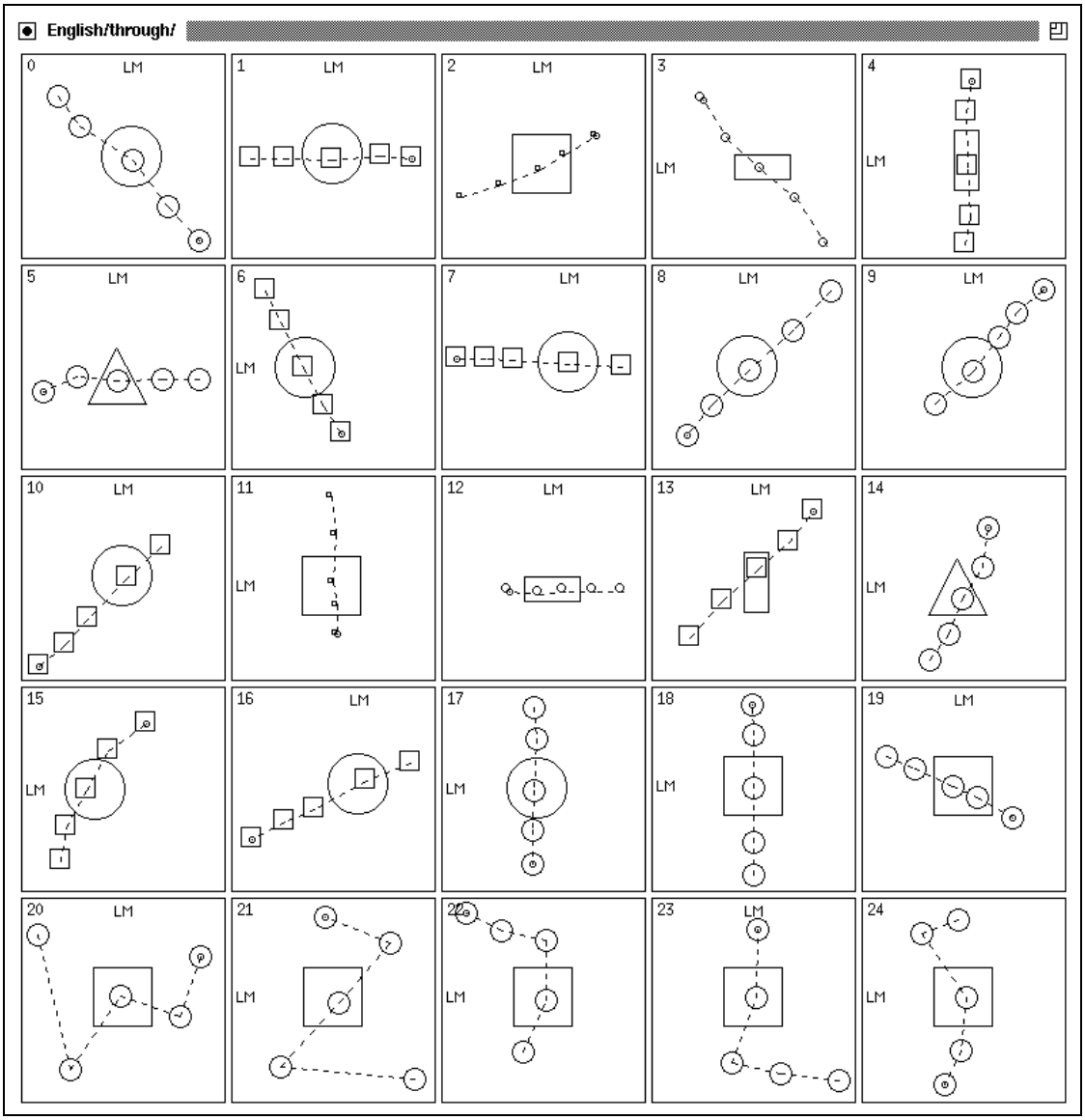


Figure 6.14: Positive examples of English *through*

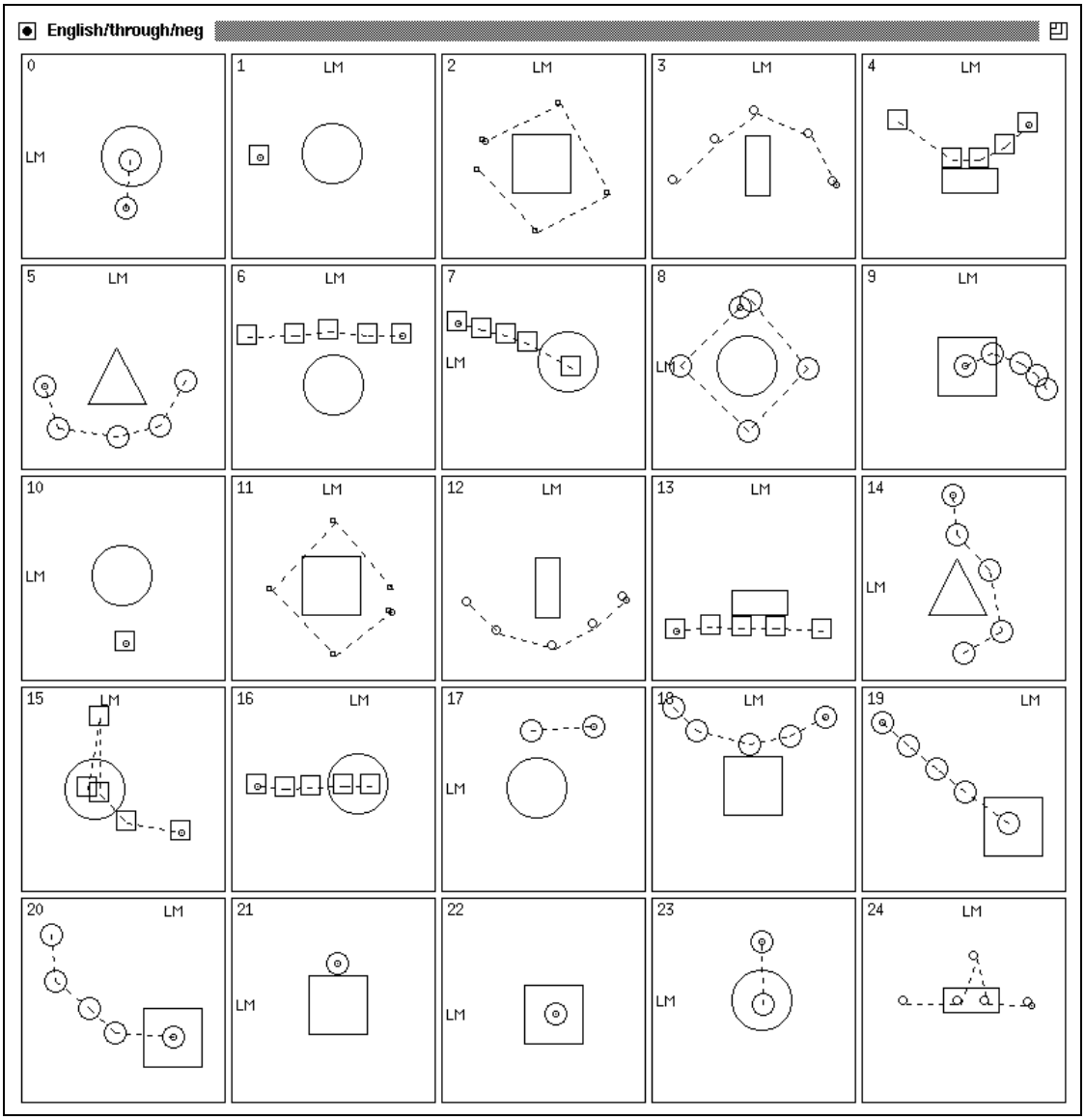


Figure 6.15: Negative examples of English *through*

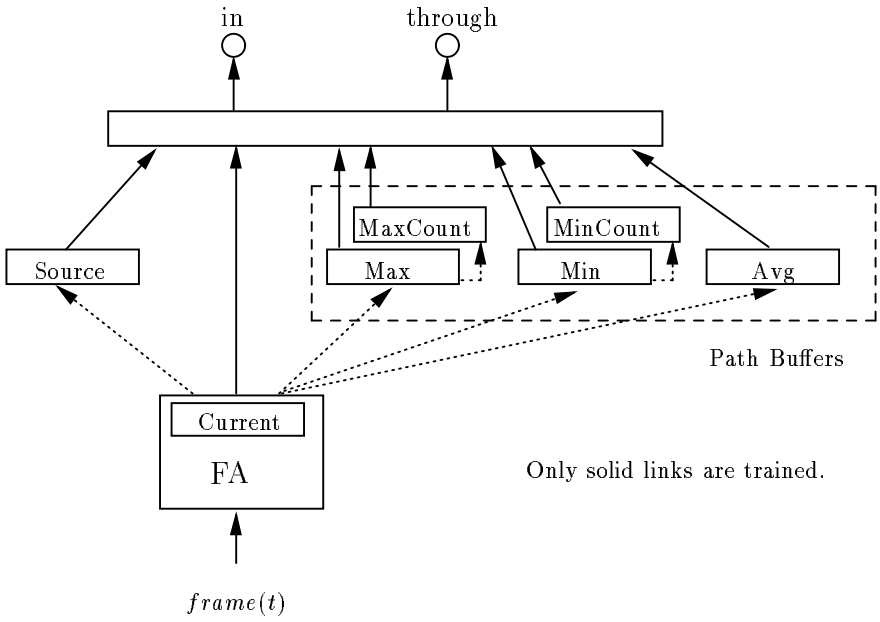


Figure 6.16: Learning *in* and *through* simultaneously

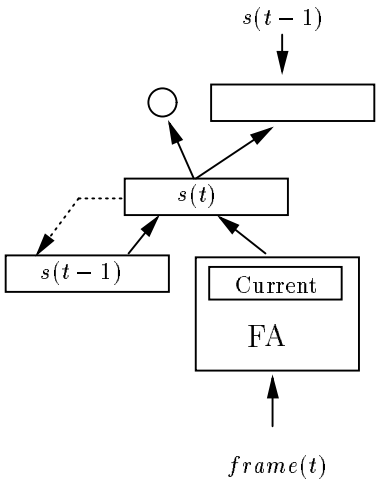


Figure 6.17: Back propagation using state units

---

forced to detect the relevant static features. Of course, children are also exposed to "training data" for a large set of words in parallel, rather than learning one at a time, so it is reasonable for a cognitive model to do the same.

6.4 Results

This section presents the results of training, using two different basic designs for the motion module: an Elman-style network, and the architecture just described above, based on the use of source and path buffers.

6.4.1 Back Propagation with State Units

A pilot study was undertaken to determine the viability of using back propagation with state units for the task of learning motion-based semantics for spatial terms. The architecture used was originally presented in Figure 6.9(b), and is shown once again in Figure 6.17. Recall that here the network must produce, as output, the state of its hidden layer on the previous time step. This forces the network to form a hidden-layer representation of the entire movie up to and including the current frame (refer back to §6.3.1 for further details on this method).

Given that the network uses this method to form a representation of the path, there is a concern one might have at the outset regarding the application of this technique here: one would expect the representations formed to be very sensitive to the exact frame in which a particular static feature appeared. This is so since the representation of the movie up to and including the current frame includes an encoding of the movie up to and including the previous frame. Thus, at the end, we have a representation not only of those events which occurred during the movie, but of the entire movie itself, frame by frame. Since there is no mechanism to detect the abstract notion of a particular static feature occurring *at some unspecified point* during the movie, there is no reason to expect that the representations formed will lend themselves easily to this.

To examine this issue, I first trained the frame analyzer to detect inclusion, then trained the architecture shown in Figure 6.17 on English *through*, which relies on the appearance of inclusion at some unspecified point in the middle of the path. This latter training was done with the weights in the frame analyzer frozen, so that they continued to detect inclusion. The idea behind doing this was to separate the issue of learning static features from the issue of learning to detect the appearance of a feature during the path described by the trajector. The results of learning *through* were then examined.

The network shown here was originally trained to detect inclusion by having it learn to discriminate between movies in which the trajector is inside the landmark in the final frame, and movies in which it is not.[7] This was done solely to make sure that the frame analyzer had learned to detect inclusion. It was verified that at least one node in the *Current* layer of the frame analyzer was in fact responding to inclusion. The output of this node was tracked over time as the network processed the positive examples of English *through* presented in Figure 6.14; the results are shown in Figure 6.18. Here, each box corresponds to a movie, the tick marks correspond to individual frames within the movie, and the height of the traced line corresponds to the strength of activation of the unit being tracked, for a particular frame of a particular movie. This activation varies between 0.0 and 1.0. A movie-by-movie comparison of this figure with Figure 6.14 reveals that the unit being tracked, in the *Current* buffer of the frame analyzer, appears to be detecting the inclusion of the trajector in the landmark. For example, the unit's response goes high on only the third frame of the first movie, and for the third and fourth frames of the third movie, corresponding to the frames during which the trajector was inside the landmark for those movies.

Once the frame analyzer had learned inclusion, the weights inside the frame analyzer were frozen, so that it would continue detecting inclusion in individual frames, and the network was trained on the positive and negative examples of English *through* shown in Figure 6.14 and Figure 6.15. Recall that we freeze the weights in the frame

[7] This is essentially the same as training it to recognize *in* (recall Figure 6.2 and Figure 6.3).

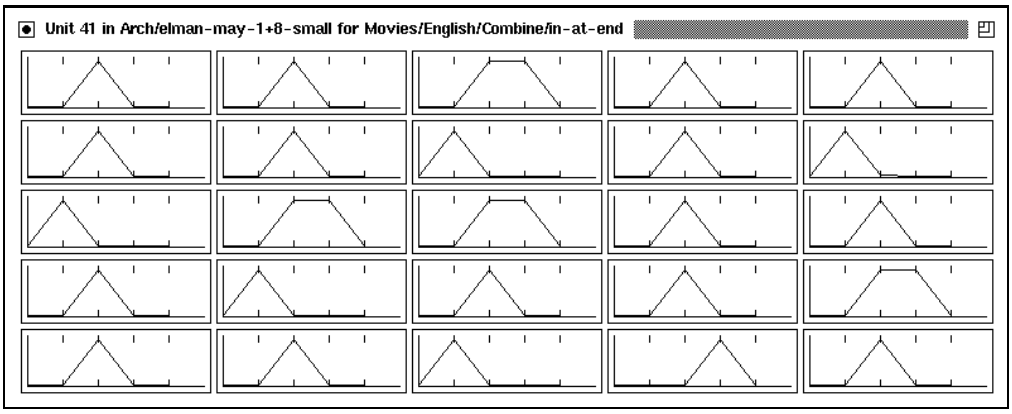


Figure 6.18: Detecting inclusion in the positive examples of English *through*

analyzer in order to temporarily leave aside the issue of static feature learning: we are primarily interested in investigating the system's ability to detect the appearance of a feature (here, inclusion) at some point during the path.

Training proceeded as described above: at each time step except for the last and the first, the network was required to replicate its state from the previous time step on the auxiliary outputs (for the first and last time steps, the teacher signal was don't-care for these outputs). For the remaining output node, the one learning *through*, the teacher signal was don't-care for all time steps except the last. At the last time step, this node received a teacher signal indicating whether the movie just viewed was a positive or negative example. At each time step, back-propagation was performed for that time step only, i.e. no back-propagation through time was performed.

There were a total of 25 positive and 25 negative examples of *through* in the training set. The best results obtained were convergence to 0.46 summed squared error after 2630 back-propagation epochs.[8] The system's performance on a test set can be seen in Figure 6.19. Here, the small number at the bottom of each box indicates how good an example of *through* the system has judged the movie as a whole to be. Clearly, the system appears to be quite sensitive to the actual frame in which inclusion occurred. Notice that those movies in which inclusion occurred only in the fourth frame are not judged as good examples of *through*. This is in spite of the fact that the training set included a movie similar to these (the second movie from the right in the bottom row of Figure 6.14). That movie in the training set was never learned correctly either. In addition, notice that movie number 7 here, a 20-frame

[8]The system was trained under "ordinary" back-propagation, rather than a second-order method such as quickprop [Fahlman, 1988], because training in this fashion, without back-propagation through time or some other means of arriving at the true gradient, yields only an approximation to the gradient. Thus, a relatively "cautious" learning algorithm is called for. Training networks of this sort under quickprop has not generally been successful.

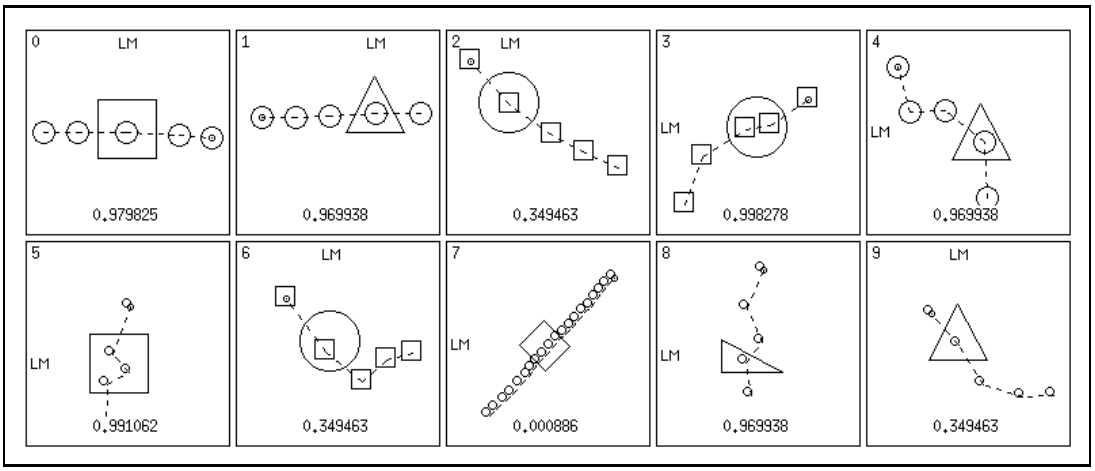


Figure 6.19: English *through* learned with state units: a test set

movie which most speakers would consider a perfectly good *through*, is judged as a very poor example of *through*.

Recall, moreover, that this was an artificial training situation: the weights inside the frame analyzer were kept frozen the entire time, so that the problem of learning to detect inclusion was separated from the problem of sequence learning. The fact that poor results were obtained even in this idealized situation is an indication that the method as a whole is not well suited to the task at hand.

6.4.2 Source and Path Buffers

Given these disappointing results from the simple recurrent network approach, we now turn our attention to the architecture described in detail earlier, involving the use of source and path buffers to keep track of the appearance of static features over time. Three sets of experiments are described: first, learning a pair of spatial prepositions from English, then a pair from Russian, and finally, a set of eight spatial terms from English.

English *in* and *through*

The English terms *in* and *through* were learned using the architecture portrayed in Figure 6.16, with one output node for *in* and one for *through*. Recall that each positive or negative instance for one term is taken, for now, as a don't-care for the other.

The training set consisted of 25 positive and 25 negative examples of *in*, and 25 positive and 25 negative examples of *through*. Part of the training set for *in* is displayed in Figure 6.2 and Figure 6.3, while the entire training set for *through* is

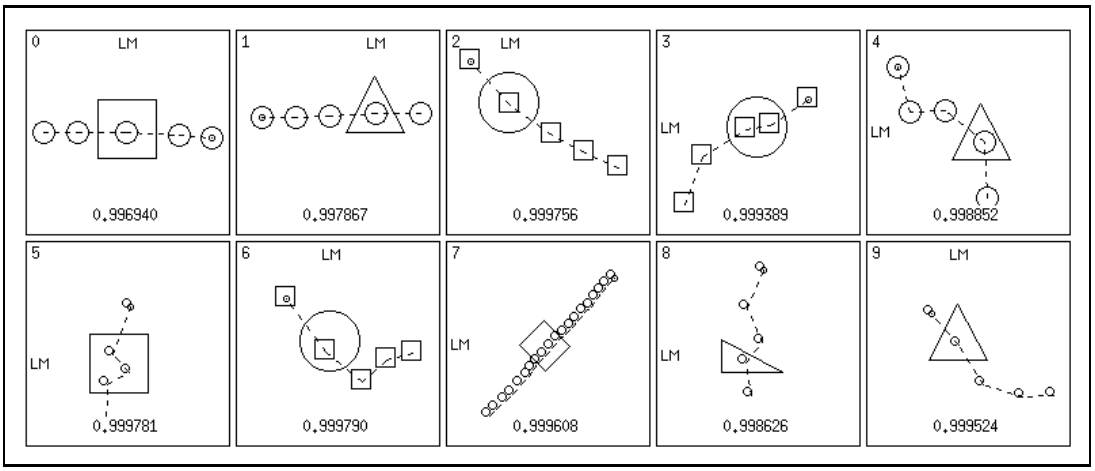


Figure 6.20: English *through* learned with source and path buffers: a test set

displayed in Figure 6.14 and Figure 6.15. Learning was extremely fast, converging to less than 0.01 error in 80 quickprop epochs.[9]

Figure 6.20 illustrates the system's performance, once trained, on the same test set for *through* that was used in testing the architecture with state units (recall Figure 6.19). In addition, Figure 6.21 presents the system's performance on a test set of negative examples of *through*. The performance is clearly much better in this case. The system is no longer sensitive to the particular frame in which an event occurred (unless of course that frame is either the first or the last). The total summed squared error on these test sets of positive and negative examples of *through*, together with a similar pair for *in*, was under 0.01, over a total of 40 test movies, indicating excellent generalization from the training set to the test set.

It is particularly instructive to note that movie 7 in Figure 6.20, which is 20 frames in length, is correctly classified, even though the training set contained only movies of five frames each. This indicates that the system is able to generalize to correctly classify movies of greater length than any in its training set.

More generally, this experiment indicates that this method is potentially very useful in the spatial sequence learning task we are faced with here. However, it is of course critically dependent on learning several terms at once. Preliminary trials, which attempted to learn *through* by itself, failed: the learning never converged to an acceptably low error rate, under either quickprop or the more cautious backpropagation. This failure is attributable to the fact that the system is unable to learn

[9]The use of quickprop, a second-order method, in this context is somewhat risky, as this training regimen does not follow the true gradient for those weights which are in the frame analyzer, and is thus at least potentially vulnerable to the same problems with second-order methods as simple recurrent networks are when trained in the Elman paradigm.

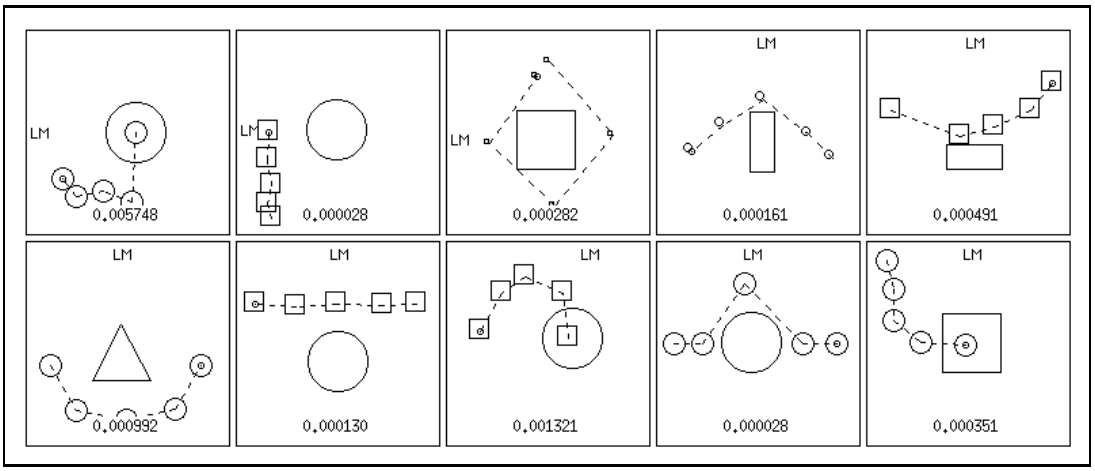


Figure 6.21: Negative examples of English *through*: a test set

to detect the static feature of inclusion, since that does not appear with any regularity in the last frame of movies in the training set.

Russian *pod* and *iz-pod*

To demonstrate the system's ability to learn motion-based semantics for spatial terms from a language other than English, we focus on Russian. Russian is chosen because, while its static terms tend to be fairly similar to those of English, there are some interesting differences in the domain of motion.

Figure 6.22 presents the Russian prepositions *pod* and *iz-pod*. While *pod* translates to "below" or "under" in English, there is no single English preposition corresponding to Russian *iz-pod*, which is best translated as "out from underneath" [Taube *et al.*, 1987].[10]

These two Russian prepositions were learned together, the idea being that in the process of learning *pod*, for which it is critical that the trajector be under the landmark in the last frame, the system will adapt itself so that the frame analyzer learns to detect the static feature of "being underneath". This static feature can then be used for learning *iz-pod* as well as *pod*.

The system was supplied with 16 positive and 16 negative exmaples of *pod*, and 16 positive and 16 negative examples of *iz-pod*. Learning was again extremely fast, requiring only 40 quickprop epochs to converge to summed squared error under 0.01. The system generalized well for this experiment as well: Figure 6.23 and Figure 6.24 present the system's judgments of how good an example of *iz-pod* each of the movies

[10]The single word *iz-pod* is morphologically derived from the words *iz* ("from") and *pod* ("under"), much the way English *into* is derived from *in* and *to*.

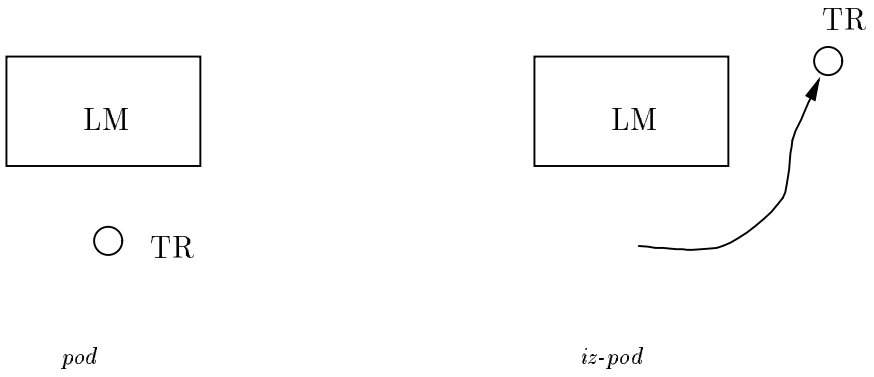


Figure 6.22: Russian *pod* and *iz-pod*

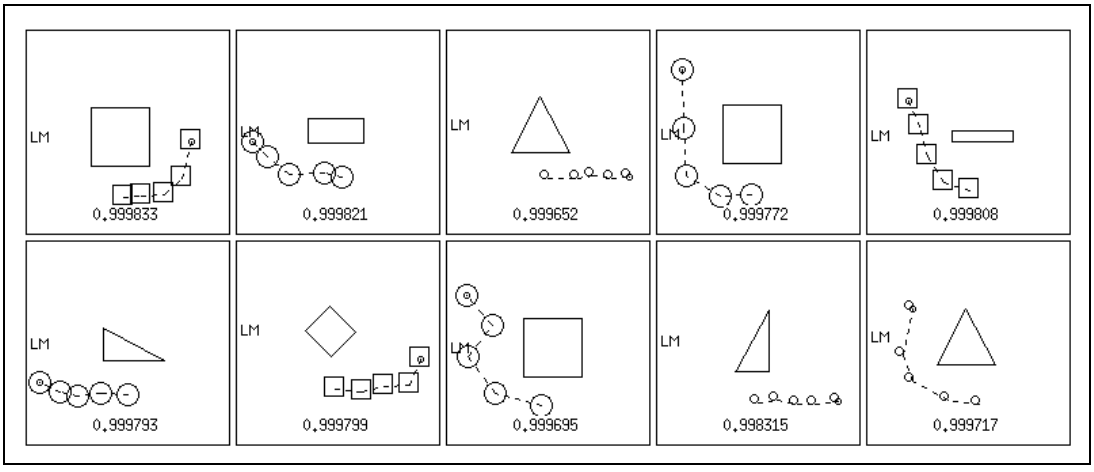


Figure 6.23: Positive examples of Russian *iz-pod*: a test set

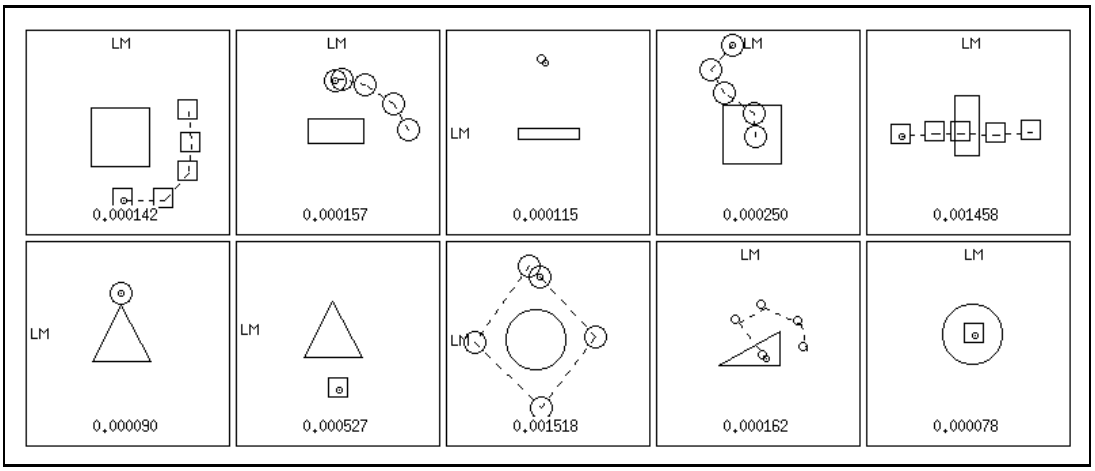


Figure 6.24: Negative examples of Russian *iz-pod*: a test set

shown is. The total summed squared error on these sets together with similar positive and negative test sets for *pod* was 0.02, over a total of 40 test movies.

6.5 Learning Movies without Explicit Negative Evidence

We now present an experiment which integrates the various aspects of the overall learning task: the system is trained to recognize movies corresponding to a set of ten English spatial terms, without the benefit of explicit negative evidence. In focusing on this integrated task, this experiment brings together the concerns of Chapter 4 (learning without explicit negative evidence), Chapter 5 (learning static features in individual frames), and Chapter 6 (movie recognition).

The ten English spatial terms are *above, below, left, right, around, in, on, out of, through,* and *over*. (Note that the English preposition *over* is highly polysemous [Brugman, 1981]. For the time being we are considering only the sense of *over* which denotes motion over and across the landmark. Further study of the other senses is discussed in §7.1.1.)

Figure 6.25 presents the architecture used in the experiments described here. This is the same as the architectures used above, except that there are ten output nodes instead of two.

This system was supplied with 126 training movies, each composed of between five and nine frames.[11] Training required 5000 epochs under back-propagation, eventually converging to less than 0.01 summed squared error. At that point, performance on

[11]There were 25 examples of *in*, 16 examples of each of *above* and *below*, 13 examples of *through*, 10 examples of each of *around, on, out of,* and *over*, and 8 examples of each of *left* and *right*.

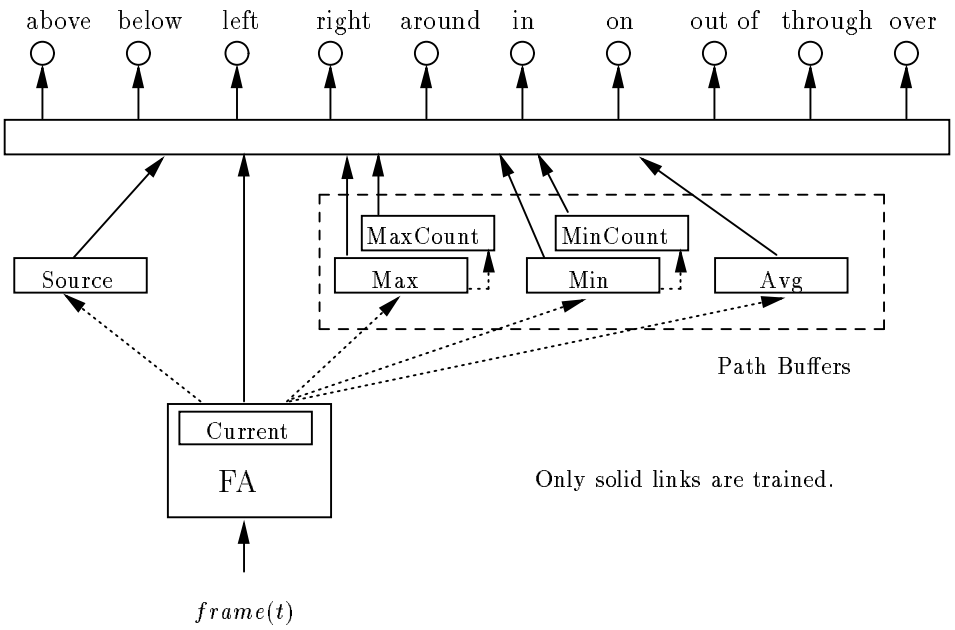


Figure 6.25: Learning ten English spatial terms simultaneously

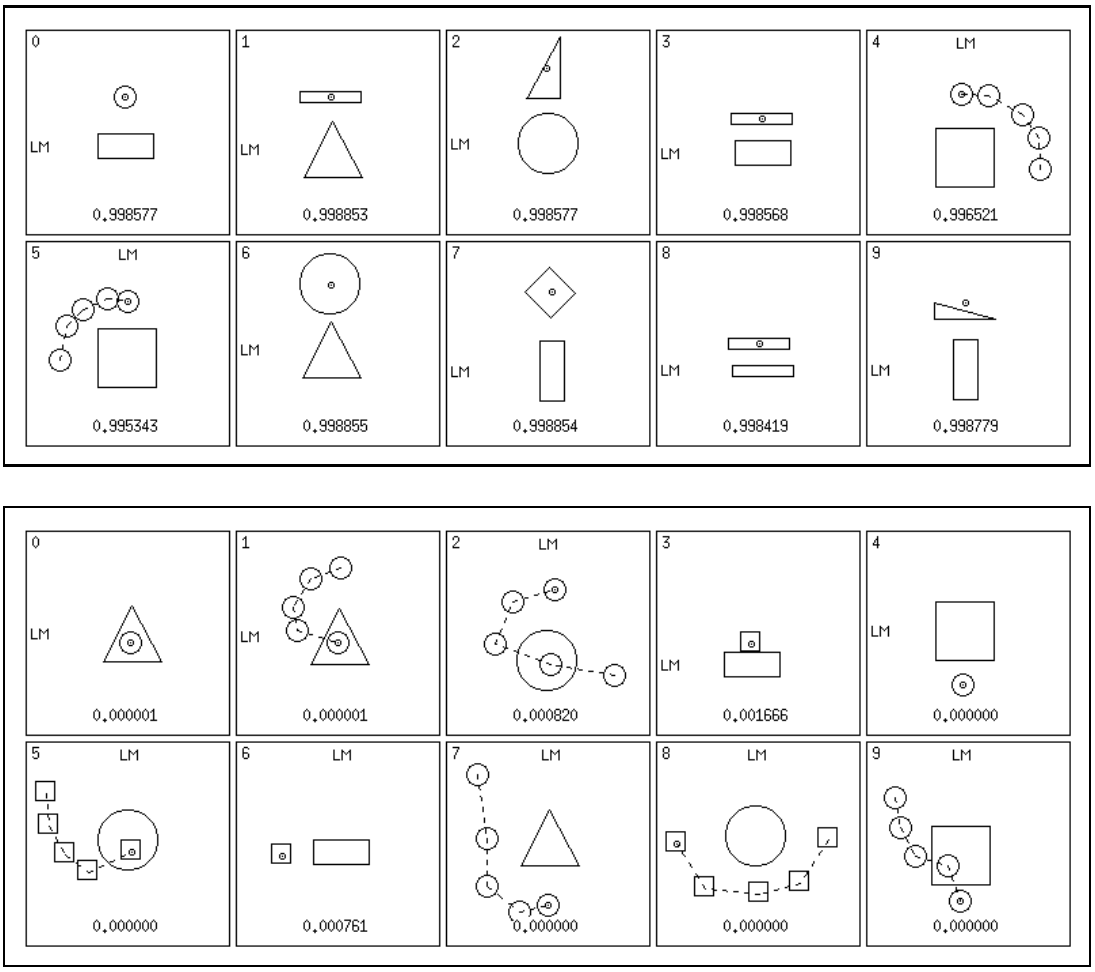


Figure 6.26: Positive and negative examples of English *above*: a test set

---

a test set of positive and negative examples of each of the prepositions showed 0.17 error, over a total of 90 test movies.

We now present examples of the system's performance on these test sets. Figure 6.26 presents the system's performance on the test sets for *above*; Figure 6.27 presents *below*; Figure 6.28 presents *left*; Figure 6.29 presents *right*; Figure 6.30 presents *around*; Figure 6.31 presents *in*; Figure 6.32 presents *on*; and Figure 6.33 presents *out of*. In addition, the system's performance on *through* and *over* will be presented below.

Notice that most responses are within 0.1 of the desired output, and that none are more than 0.37 from the desired output.

These responses were obtained at the last time step of each movie. However, recall from our discussion earlier in this chapter that we can expect the training regimen used here to cause the system to respond appropriately *at each time step* to the

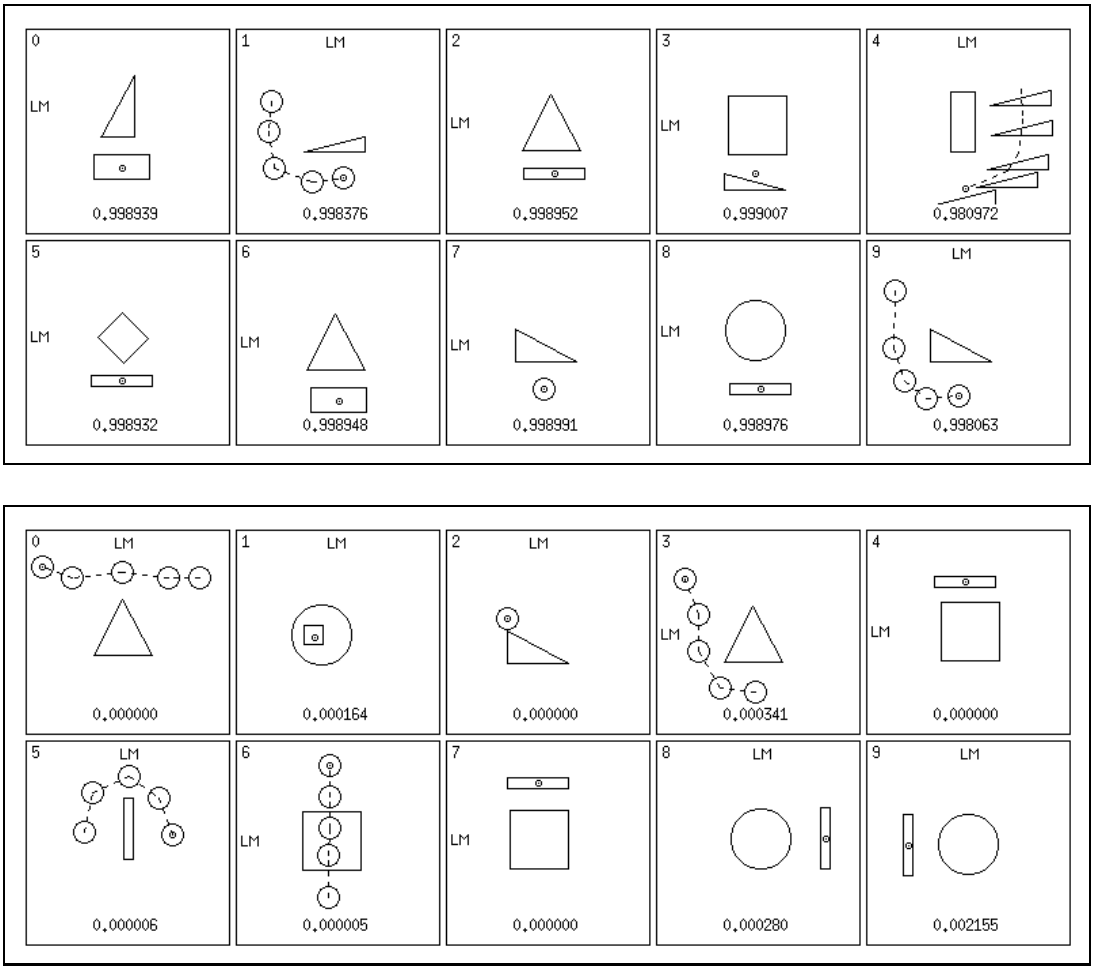


Figure 6.27: Positive and negative examples of English *below*: a test set

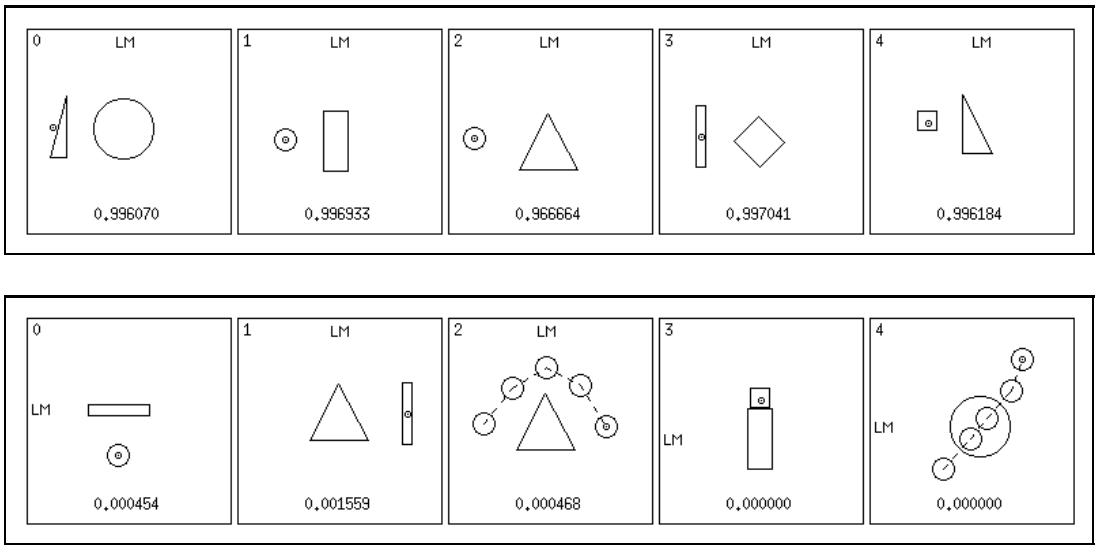


Figure 6.28: Positive and negative examples of English *left*: a test set

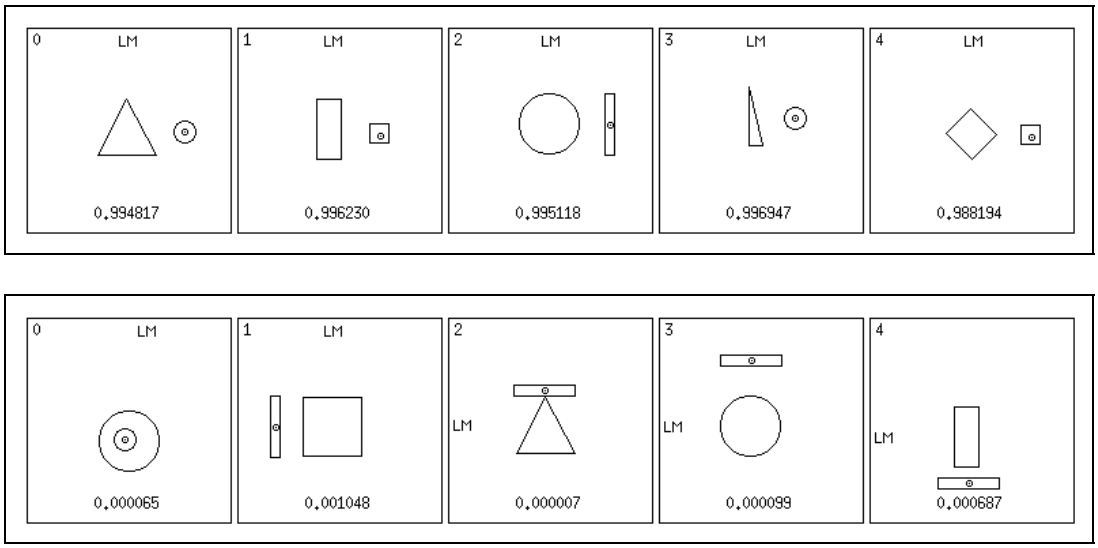


Figure 6.29: Positive and negative examples of English *right*: a test set

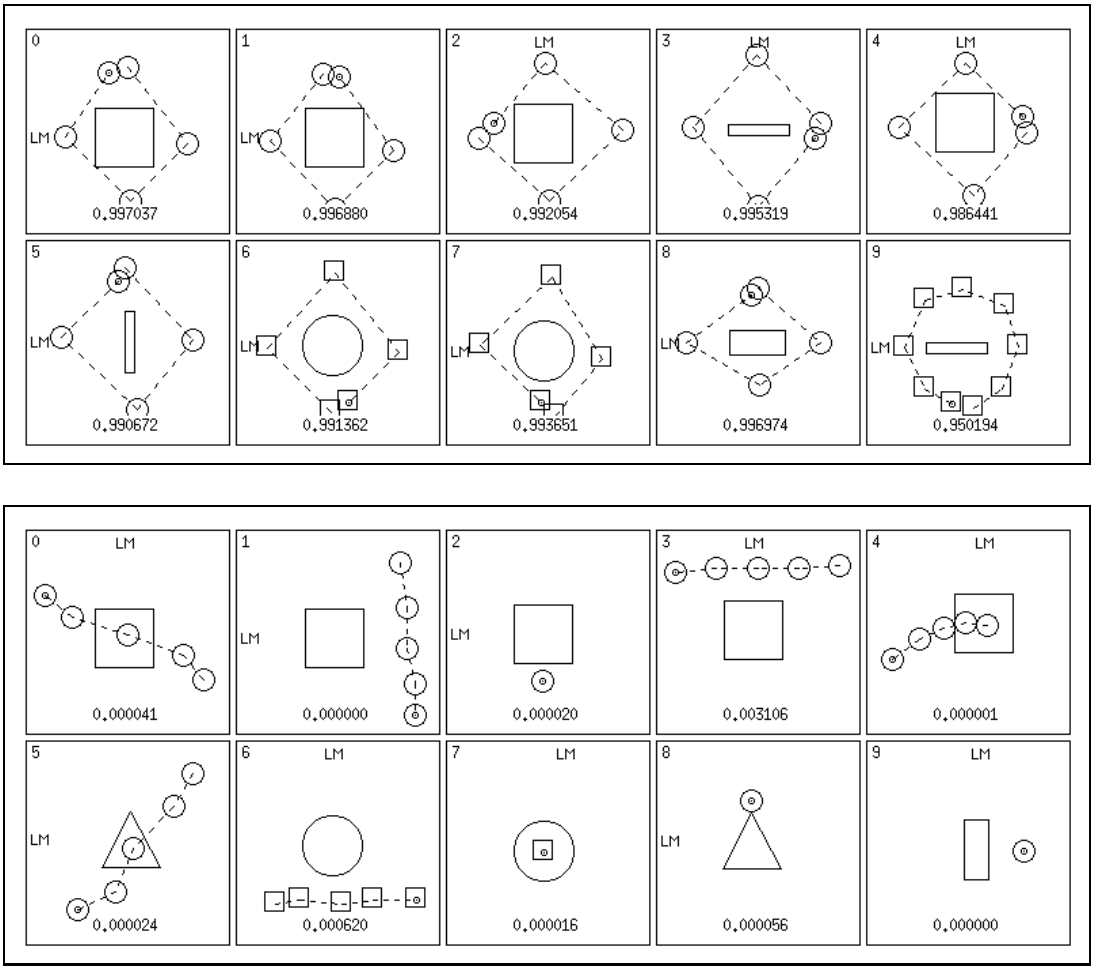


Figure 6.30: Positive and negative examples of English *around*: a test set

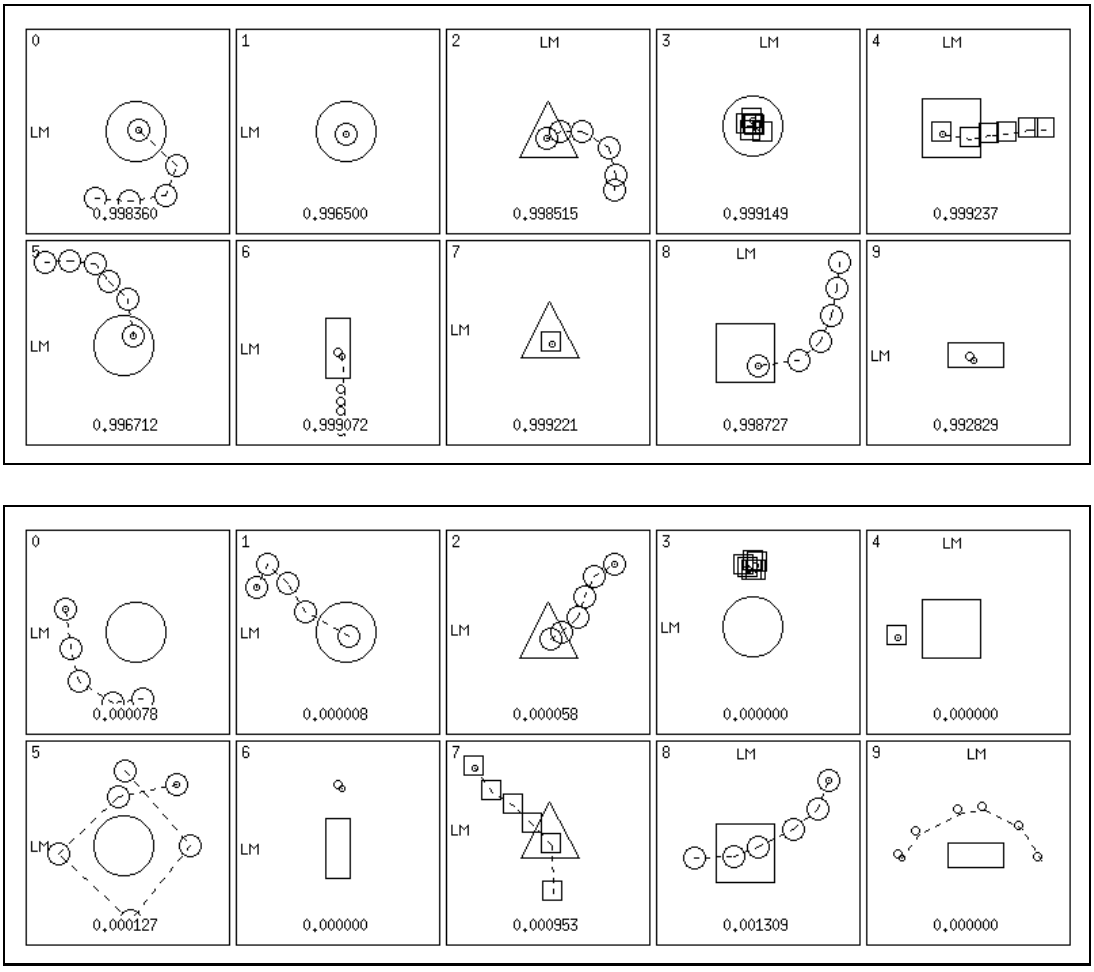


Figure 6.31: Positive and negative examples of English *in*: a test set

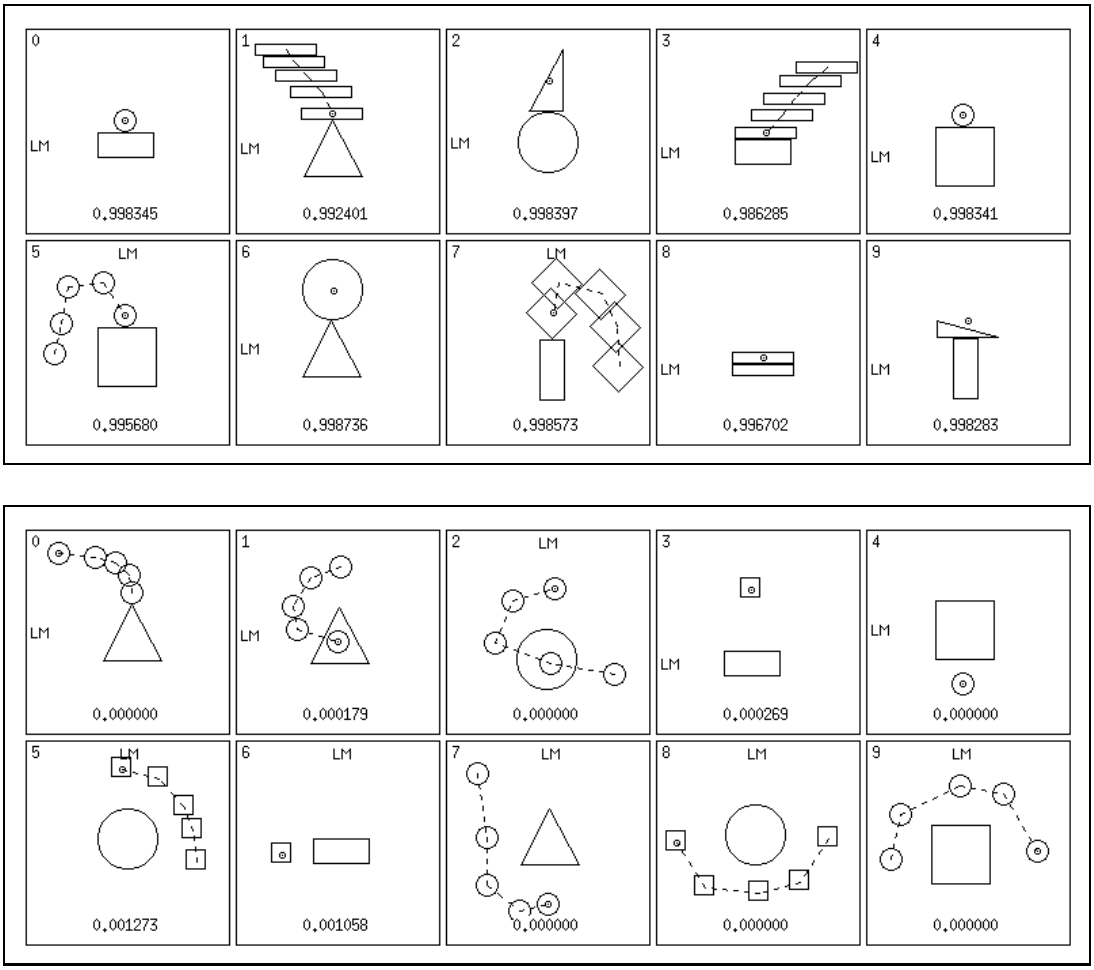


Figure 6.32: Positive and negative examples of English *on*: a test set

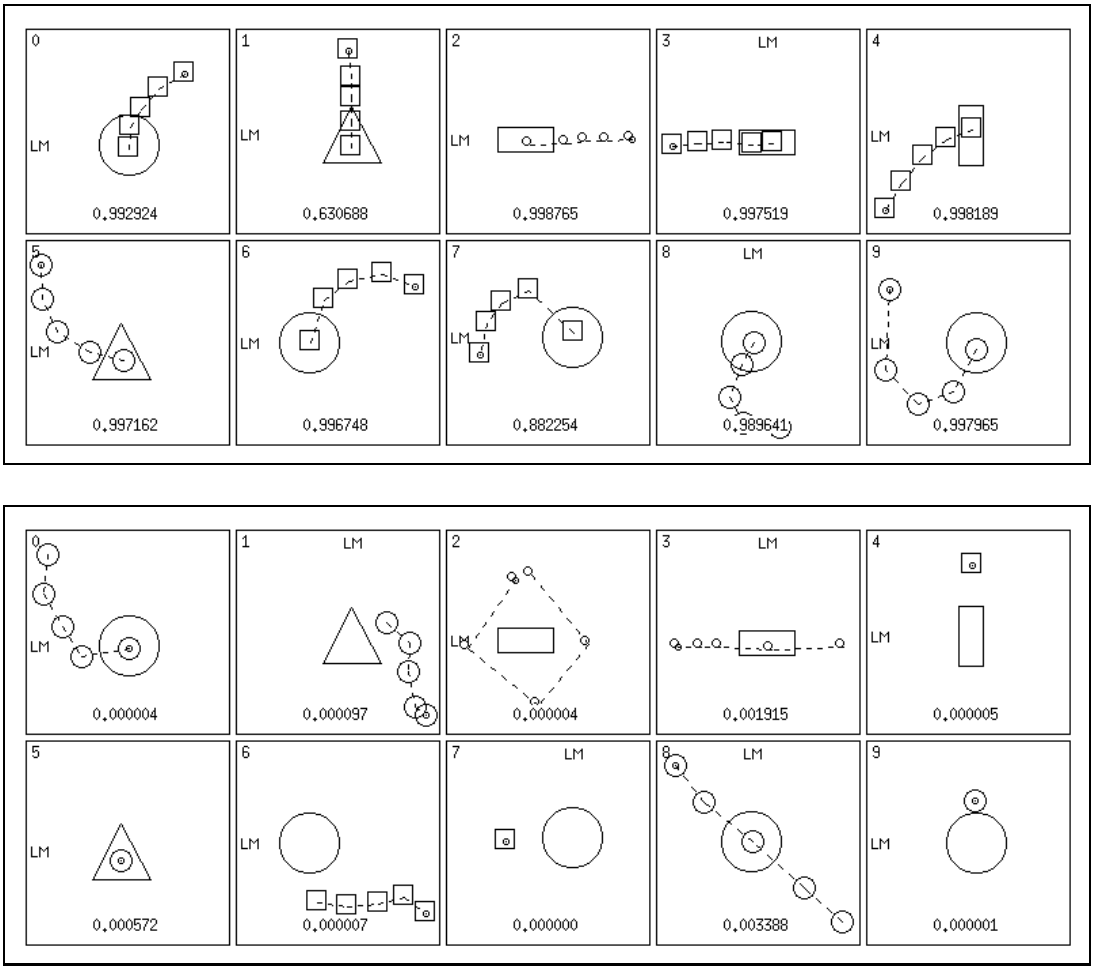


Figure 6.33: Positive and negative examples of English *out of*: a test set

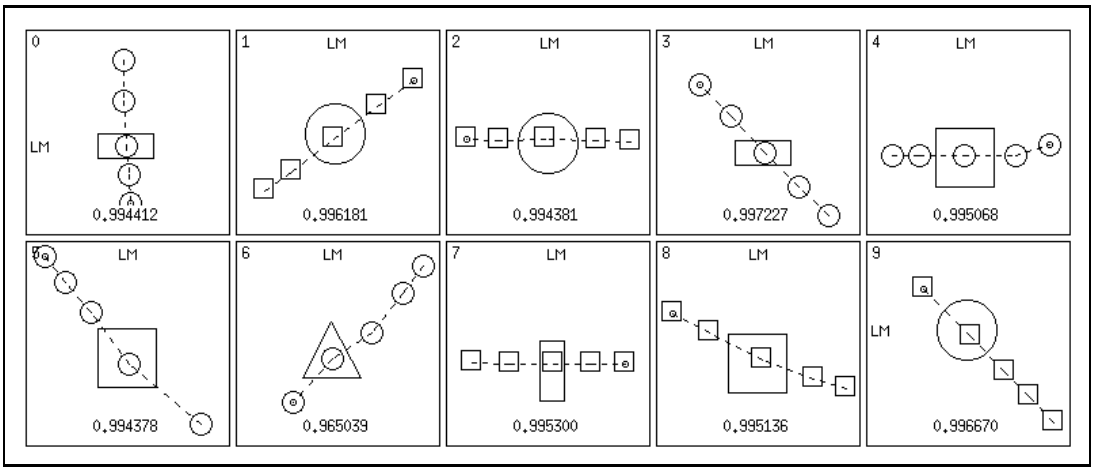


Figure 6.34: Positive examples of English *through*: a test set

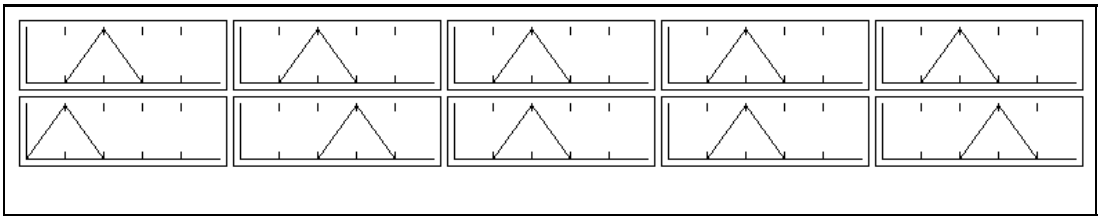


Figure 6.35: Response over time to *in* while viewing positive examples of *through*

portion of the movie viewed so far. For example, we can expect the output node for *in* to respond strongly in mid-event for movies which correspond to *through*, and can similarly expect the output node for *above* to exhibit a strong response in mid-event for movies corresponding to *over*, since we are restricting our attention here to the "over and across" sense of *over*. We now present examples of the system doing just that.

Figure 6.34 presents the system's performance on a test set of positive examples of *through*, while Figure 6.35 shows the system's response over time at the output node for *in* while viewing the positive examples of *through* shown in Figure 6.34. (In addition, Figure 6.36 presents the system's performance on a test set of negative examples of *through*.) By comparing Figure 6.34 and Figure 6.35 one can see that the output node for *in* is strongly activated any time the movie up to and including the current frame is a good example of *in* (here, this is the "into" sense of *in*).

Very much in the same vein, Figure 6.37 shows the system's performance on a test set of positive examples of *over*, while Figure 6.38 presents the system's response over time at the output node for *above* while viewing these positive examples of the over-

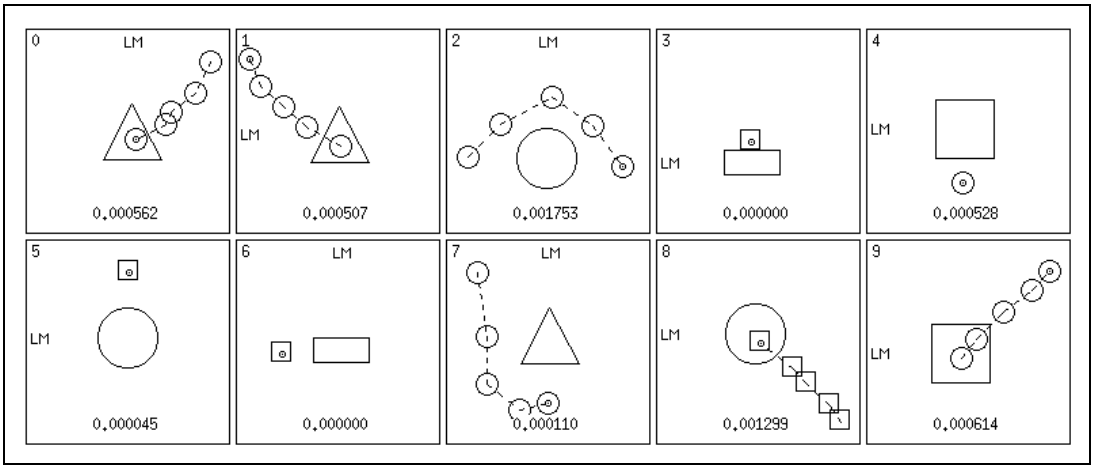


Figure 6.36: Negative examples of English *through*: a test set

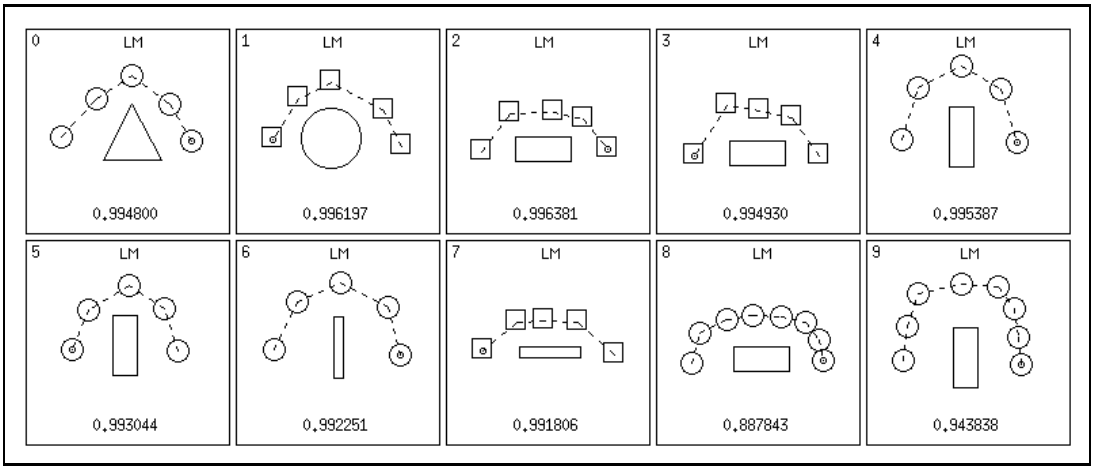


Figure 6.37: Positive examples of English *over*: a test set

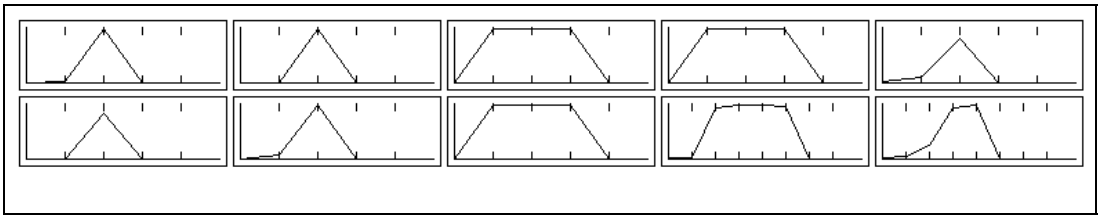


Figure 6.38: Response over time to *above* while viewing positive examples of *over*

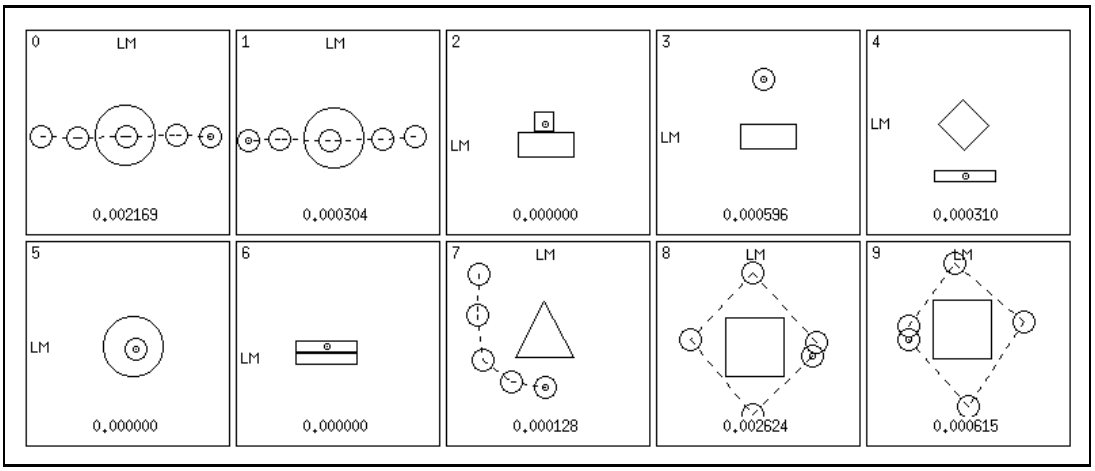


Figure 6.39: Negative examples of English *over*: a test set

and-across sense of *over*. (In addition, Figure 6.39 presents the system's performance on a test set of negative examples of *over*.) Analogously to the situation obtaining for *in* and *through* above, we find here that the output node for *above* is strongly activated whenever the movie up to and including the current frame is a good example of *above*.

Thus, although the network was trained only on the last time step of movies, it responds appropriately on intermediate frames as well. This is attributable to the fact that the network is structured so as to distinguish the initial frame, the intermediate frames taken as a whole, and the final frame. Since at each time step the current frame is viewed as the final frame, if the movie up to and including some intermediate frame is an instance of a term for which the system has been trained, the network responds appropriately at that intermediate time step.

6.6 A Fictitious Spatial Concept: *in/out-of*

It is very common for prepositions to denote both static location in some situation, and motion into that situation. Consider for example the uses of the English prepositions *in* and *under*. Each of these prepositions can be used in both static and motion-into senses, as seen in the following sentences:

- He is *in* the room.
- He walked *in* the room.
- The dog is *under* the bed.
- The dog crawled *under* the bed.

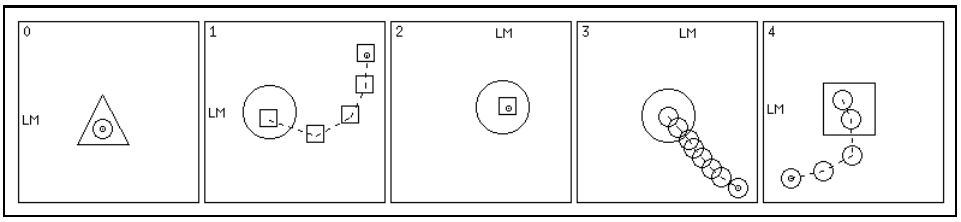


Figure 6.40: *in/out-of*: a fictitious spatial concept which is difficult to learn correctly

English is by no means unique in this regard; many other languages similarly use a single word in these two senses.[12] This phenomenon is easily modeled by the system presented here, as a glance back at Figure 6.31 will show. This figure shows the system's performance on a test set for *in* which includes both static and motion-into uses of the word.

By contrast, I am not aware of any language which uses the same word for both the static situation obtaining at the *beginning* of some motion event, and the motion event itself. This would correspond to using a single word in English to refer to both static *in* and motion *out of* a landmark, as illustrated in Figure 6.40. For lack of a better name, I refer to this concept somewhat awkwardly as *in/out-of*.

Preliminary experiments indicate that the system has difficulty with such concepts. The system was trained with 25 positive and 25 negative examples of *in/out-of*, eventually converging to under 0.01 total summed squared error. However, the generalization from the training set to a test set was quite poor (6.20 total summed squared error over a total of 20 test movies), but excellent in the case of *in* (0.00 error over 20 test movies). The system's performance on test sets of positive and negative examples of *in/out-of* can be seen in Figure 6.41. Here, the top row contains positive examples, while the bottom row contains negatives. Clearly, there are a number of misclassifications. Most notably, in the top row movies 2 and 4, which show motion of the trajector out of the landmark, are considered poor examples of *in/out-of*, while in the bottom row movies 1 and 2, which show motion of the trajector *into* the landmark, are classified as good examples of *in/out-of*, despite the fact that the training set contained positive examples showing motion out of the landmark and negative examples showing motion into the landmark.

Thus the system has failed to pick up on a simple abstraction which could serve to distinguish positive from negative examples of *in/out-of*: if the trajector is inside the landmark at the beginning of the movie, the movie is an instance of *in/out-of*. However, it is able to pick up on the corresponding abstraction in the case of *in*: if

[12]Some Indo-European languages which use the same preposition for both location and motion into a location use the accusative case to mark the motion-into sense, the so-called "accusative of motion", and some other case for the static sense. Examples of such languages are German and Russian.

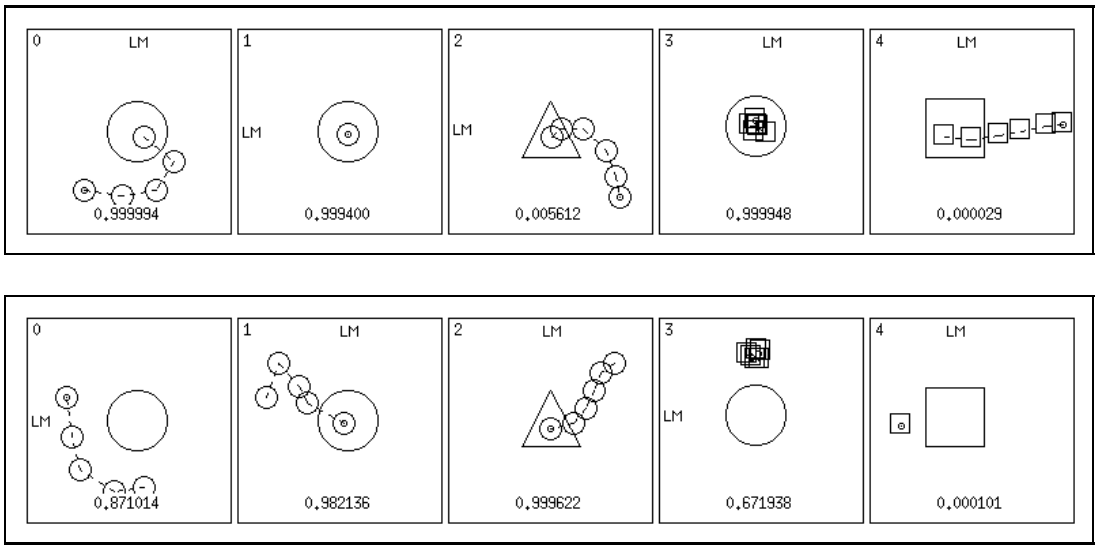


Figure 6.41: Test sets: positive and negative examples of *in/out-of*

the trajector is inside the landmark at the end of the movie, the movie is an instance of *in*. For the time being, it will suffice to simply note this discrepancy. We shall return to this issue below, to discuss why this might be the case, and to examine the ramifications.

6.7 Discussion

Two of the most important distinguishing aspects of the movie-learning task dealt with here are the need to learn to detect static features while learning to respond to sequences of those features, and training only on the last time step of each movie. These two taken together, under the approach taken here, lead to a system which is able to learn to detect static features only if they appear in the final frame of some training movie. This emphasis on the final frame is a central defining feature of the system presented here, and it is worthwhile to note that there is some motivation for giving the final frame a privileged status, and that there are also interesting consequences of doing so.

[Behrend, 1989] presents evidence that children and adults, when learning novel verbs, tend to consider the *result* of the verb the most important component of the verb's meaning. In the domain of motion, this would correspond to a focus on the endpoint of motion. For example, when learning the verb *enter*, the most important component of the verb's semantics would be taken to be the fact that after entering, the trajector is *in* the landmark. This then supports the very general notion that the endpoint of an event has privileged status in language.

We now consider the implications of the system's ability to learn static features only if they appear in the final frame of some training movie. Recall that although we have used explicit negatives for some experiments, the system is meant to be used without such explicit negatives. Rather, positive examples for other terms will be taken as implicit negative evidence for the term in question, as was done in §6.5. Doing away with explicit negative evidence in this way means that the system will be able to learn a static feature only if it appears in the final frame of a movie which is taken as a positive example for some term in the language, since that is the only source of training data, and since training occurs only on the final time step. This in turn means it will only learn a particular static feature if that feature appears at end-event for some term in the language, or appears in a static scene named by some term in the language (recall that static scenes are represented by movies in which the trajector remains immobile, meaning that all frames are identical, so that all static features which appear at all in the movie appear in the final frame.)

Thus, if taken as a rough model of human learning, even if only in broad outline, this gives rise to the following linguistic prediction, a posited implicational universal:

> **Prediction:** *For any language, any static feature which is semantically significant*[13] *will appear either at end-event for some spatial term in that language, or in a static configuration named by a spatial term in that language.*

For example, by this posited universal, the fact that the static feature of inclusion appears in mid-event for English *through* leads us to predict that there will exist a term in English denoting an event or state whose end outcome is inclusion of the trajector in the landmark (such as *in* or *into*). This is so since otherwise the model predicts that the term *through* would not be learnable. This prediction is open to empirical falsification.

Another prediction which falls out of the model concerns spatial concepts along the lines of the fictitious *in/out-of* presented above in §6.6. We have seen that the system is far more successful at learning *in*, with both its static "in" and "motion-into" senses, than *in/out-of*, which combined static "in" with "motion-out-of". Recall that it is easy to articulate simple principles which will discriminate positive from negative examples for each of *in* and *in/out-of*: if the trajector is inside the landmark during the last frame of the movie, the movie is classified as *in*, while if the trajector is inside the landmark during the *first* frame of the movie, the movie is classified as *in/out-of*. However, although these two principles are equally straightforward, the system finds *in* much easier to learn than *in/out-of*. The structure of the system thus appears to implicitly embody a bias towards spatial concepts that use the same word to label both static location and motion into that configuration.

[13] I.e. whose presence or absence is critical to the categorization of at least one spatial term in the language

Reflecting back on the structure of the architecture reveals that it makes sense that this would be the case (recall for example Figure 6.10). Specifically, in order to learn the static feature of inclusion, which is critical to both *in* and *in/out-of*, the system will have to back-propagate error into the frame analyzer, and in order for this error signal to have much of an effect, the weights that link the frame analyzer to the hidden layer above it will need to be substantial, since the formula for computing the error signal involves a multiplication by these weights (see [Rumelhart *et al.*, 1986]). But if this is the case, it means that during the forward pass, the contents of the *Current* buffer of the frame analyzer will in turn have an effect on the representation formed in the hidden layer above it. And since at the end of a movie, the *Current* buffer holds a representation of the final frame, this means that the final frame will have an effect on the representation formed, and thus on the classification of the movie as a whole. These weights could change after the frame analyzer has learned whatever static features it needs, but there is clearly a bias toward incorporating information from the final frame into the hidden layer representation of the movie. Thus, the architecture does in fact embody a bias that would make it more difficult to learn *in/out-of*, in which the final frame is different for the two senses, than *in*, in which the final frame is the same for the two senses. Notice that although this bias is a form of emphasis on the final time step, it does not make it impossible to learn such concepts as Russian *iz-pod* (out-from-under), for which the trajector must be under the landmark at the beginning of the movie, or English *through*, for which the trajector must be inside the landmark in mid-movie: the critical point here is that neither *iz-pod* nor *through* has two separate senses for which the final frames differ, as is the case with *in/out-of*.

If we are to view the system as a cognitive model, this leads us to the following linguistic prediction:

> **Prediction:** *The use of a single word to denote either static location in some configuration or motion into that configuration will be more likely to appear in a language than the use of a single word to denote either static location in some configuration or motion out of that configuration.*

This is in accord with the fact that languages such as German and Russian (and to a degree, English) do in fact use the same word for both a static configuration and motion into that configuration, while none that I am aware of use the same word for both a configuration and motion out of that configuration.

Recall from Chapter 1 that one of the central themes of this thesis is that a focused computational modeling effort in a subfield of linguistic inquiry can give rise both to general techniques which may find applicability outside the original domain for which they were invented, and to falsifiable predictions concerning the nature of language. At the end of Chapter 4, we noted that the technique of using weak implicit negative instances is a very general method that resulted from the work described in this

thesis – that is, it is an example of the way in which such a modeling effort can give rise to computational techniques of general applicability. Here, we note that the two predictions outlined above arise from the particular architecture and training regimen used, and thus provide an example of the manner in which a modeling effort of this sort can give rise not just to techniques, but also to concrete, falsifiable linguistic predictions.

One possible response to this presentation of linguistic predictions would be to suggest that these predictions might be dependent only on the very abstract notion of end-point emphasis, independent of architectural details. That is, one might argue that the predictions follow directly from the notion of end-point emphasis, and that *any* model which embodied this abstract notion would of necessity give rise to these predictions; under this view, the predictions are not so much a product of the particular architecture and training regimen used here as they are a product of this very general feature which characterizes this system among many other possible ones. They could perhaps be arrived at, then, simply by noting the existence of end-point emphasis, and working out the ramifications of the existence of such a phenomenon – without having to build a model at all.

There is a serious problem with this line of argument: the abstract notion of end-point emphasis can certainly make the predictions seem *plausible*, but they do not strictly *follow* from it. For example, consider the first of the two predictions above. It seems plausible, given some abstract notion of end-point emphasis, that static features would only be learned if they appeared at the end of some training movie, but the idea of end-point emphasis is too imprecisely formulated for us to say that this prediction actually follows from it. Notice that the same abstract notion of end-point emphasis also makes it seem plausible that terms which in some way highlight the beginning of the event, rather than the end, could not be learned — but this is not the case at all, Russian *iz-pod* (out-from-under) being a counterexample. This spatial term, which highlights the beginning of the event in that it focuses attention on where the landmark was *before* it starting moving, has been learned by the system described here. This then illustrates the importance, in this instance, of working out the details of a model, rather than dealing with these notions at a more abstract level.

There is, however, another related point: the general phenomenon of end-point emphasis in language could have sprung from any of a number of sources quite separate from the architectural "explanation" given here. For example, it is reasonable to consider the function of language generally to be that of communicating new information, and since starting-points for actions are often known, end-points will be more saliently marked in language.[14] The point here is that since there are alternative possible sources for end-point emphasis in language, the chances are excellent that they too might play a role in the phenomena discussed here, a role that is overlooked if we restrict our attention to the architecture.

[14]Robert Wilensky suggested this way of viewing the issue.

The primary issue associated with this point is the explanatory force of the model. For if there is more to the story than the model would lead one to believe — and there almost certainly is — we cannot view the model as being that which one might desire: a fully explanatory model of the particular set of linguistic phenomena under consideration. Rather, it provides what is at best a partial explanation. I feel that the model presented in this thesis is best viewed in these terms.

Having examined the consequences of using the particular architecture for motion presented here, we can now conclude our presentation of the core system itself. In the next chapter, we shall be examining a number of extensions to this core system.

Chapter 7

Extensions

7.1 Ongoing Work 152
7.1.1 Polysemy 153
7.1.2 Deixis 163
7.1.3 Force Dynamics 168
7.1.4 Reified Paths and Key Events 172
7.2 Possibilities for Future Work 176
7.2.1 Distance 176
7.2.2 Regions as Objects 178
7.2.3 Convex Hulls 180
7.2.4 Implicit Paths 181

This chapter presents a number of extensions to the core system and techniques that were presented earlier. As we will see, a good number of directions for extension suggest themselves – so many, in fact, that the work actually done to date seems, in retrospect, to be in large part a foundation-laying, rather than anything approaching the final word.

We begin by covering extension work that was recently undertaken, and move on to discuss further possibilities.

7.1 Ongoing Work

In the spring of 1992, a seminar was offered at the University of California at Berkeley, cross-listed in the computer science and linguistics departments to attract graduate students from both disciplines. We began by reviewing the work presented in earlier chapters of this thesis, and then considered a number of extensions to that work, several of which were implemented by students as course projects. This section describes several of these extensions, and the issues they raise.

It is here, to my mind, that the preliminary nature of the core work of this thesis becomes most apparent. Some of the most linguistically interesting issues related to the acquisition of perceptually grounded semantics (e.g. polysemy, deixis) are in fact

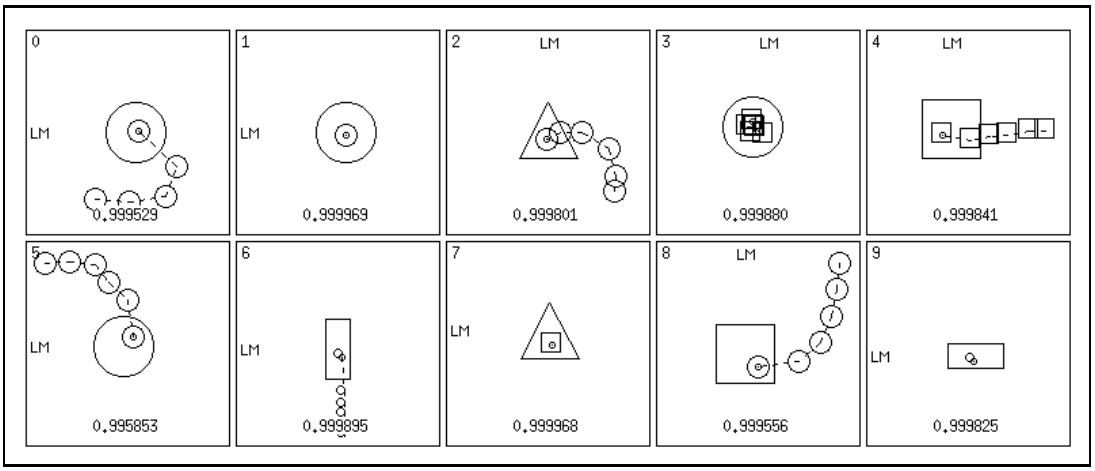


Figure 7.1: A test set for English *in*, learned by itself

the ones addressed by the various extensions, rather than by the bulk of the thesis itself. Under this view, the thesis work has served primarily to provide a framework within which to think about these issues, and a software environment within which to investigate them.

Without the work of the students in the seminar, the issues discussed in the remainder of this section would only have been addressed in a rather speculative fashion in this thesis. With the exception of the discussion of the polysemy of English *in*, the extensions and modifications presented below are entirely theirs.

7.1.1 Polysemy

One issue which was addressed during the course of the seminar was *polysemy*, the phenomenon of a single word having a number of distinct but related senses. In this section, we begin by examining, somewhat more closely than before, the (arguable) polysemy of a spatial term we have already discussed, namely English *in*, and then move on to consider a series of experiments which were run using an extension to the system, addressing the polysemy of English *over* and *under*.

Figure 7.1 presents the system's performance on a test set for English *in*, after it was trained on this spatial term alone. There is arguably a form of polysemy present in this test set, as there was in the training set[1] as well, as movies of two distinct sorts appear here:

- Movies depicting the trajector remaining within the landmark (movies 1, 3, 7, and 9)

[1]The training data was presented at the beginning of Chapter 6.

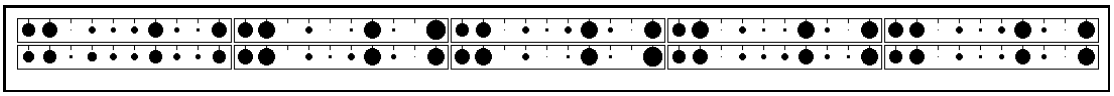


Figure 7.2: Contents of the top hidden layer for *in*, learned by itself

- Movies depicting motion of the trajector *into* the landmark (movies 0, 2, 4, 5, 6, and 8).

The reason this is only arguably an instance of polysemy is that there is in fact a single abstraction over these two cases: the trajector is inside the landmark at the last time step of the movie. This single, simple criterion will serve to distinguish positive from negative examples of *in*, leaving no need for more intricate semantic theorizing.[2] (As we shall see below, there are English prepositions for which no such abstraction appears to exist.) Thus it is not clear on what basis one might say that these two senses are distinct, without starting down a slippery slope that would lead us to start asserting that each and every movie in the set is an instantiation of a "distinct" sense. And yet we do have the persistent intuition that the movies in the above figure break down into the two senses outlined above. We now move on to examine under just what conditions the system will come to the same conclusion.

Figure 7.2 presents the activation vectors for the top hidden layer of the architecture at the last time step of each movie, for the movies in Figure 7.1. There are ten activation vectors shown, one for each movie in the figure; the position of the activation vector within Figure 7.2 corresponds to the position of the movie it represents in Figure 7.1, and the size of the black circles indicates how strongly a particular unit in the hidden layer is activated. It should be apparent that these activation vectors do not differ much from movie to movie; in particular, there is no obvious division into two types of activation "signature", which could be interpreted as corresponding to the two senses outlined above. Similarly, when we examine Figure 7.3, which presents the results of hierarchical clustering on the activation vectors shown in Figure 7.2, we find that the clustering does not correspond to the two-way breakdown which seems so intuitively obvious. Here, the horizontal spacing between nodes in the tree is proportional to the distances between clusters. So the primary distinction made here is between movie 5 and all others, rather than between "remaining-in" and "moving-in".

On the other hand, when *in* is learned in conjunction with *through*, as discussed in Chapter 6, an entirely different picture emerges. Figure 7.4 illustrates the system's performance on the same test set we viewed earlier, and Figure 7.5 and Figure 7.6 present the top hidden layer activation vectors at the last time step, and the results of

[2] At least not within this limited domain. In actuality, the full semantics of *in* are a good deal more subtle than I make them out to be here. Compare for instance "He walked *in* the room", which seems a perfectly normal usage of the word *in* in its motion-into sense, with *"He drove *in* Berkeley", which does not. These examples are due to George Lakoff.

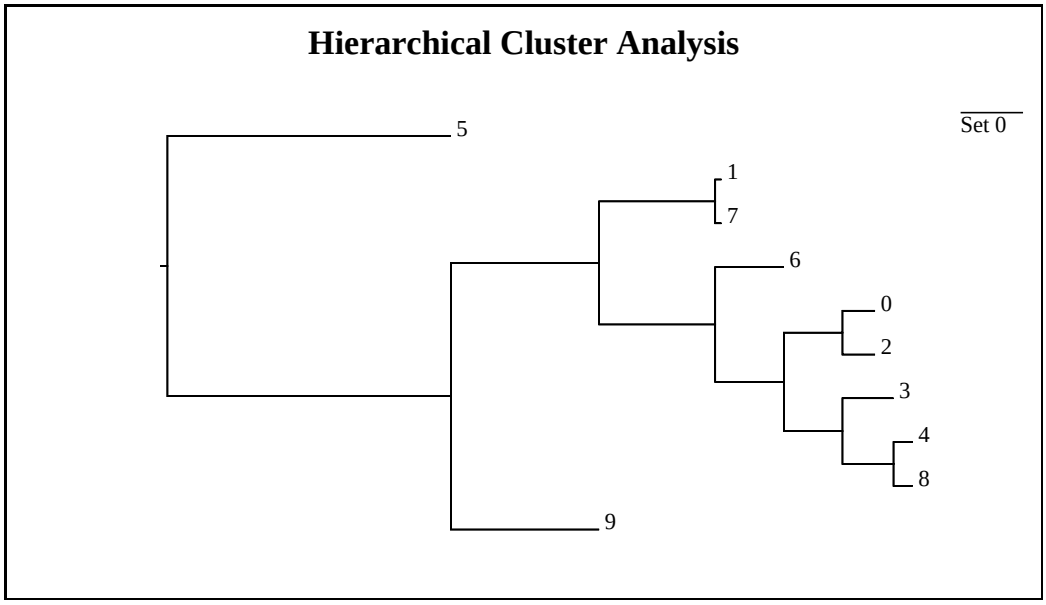


Figure 7.3: Cluster diagram for *in*, learned by itself

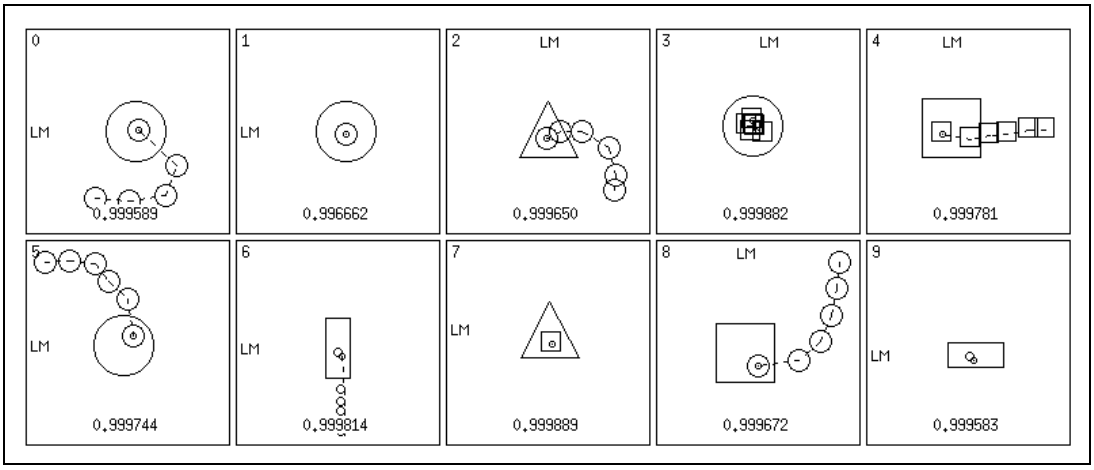


Figure 7.4: A test set for English *in*, learned together with *through*

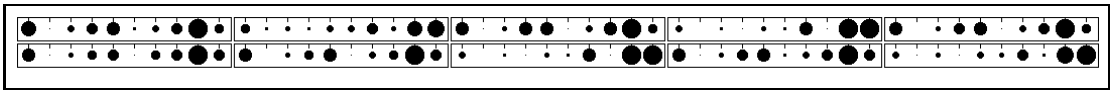


Figure 7.5: Contents of the top hidden layer for *in*, learned together with *through*

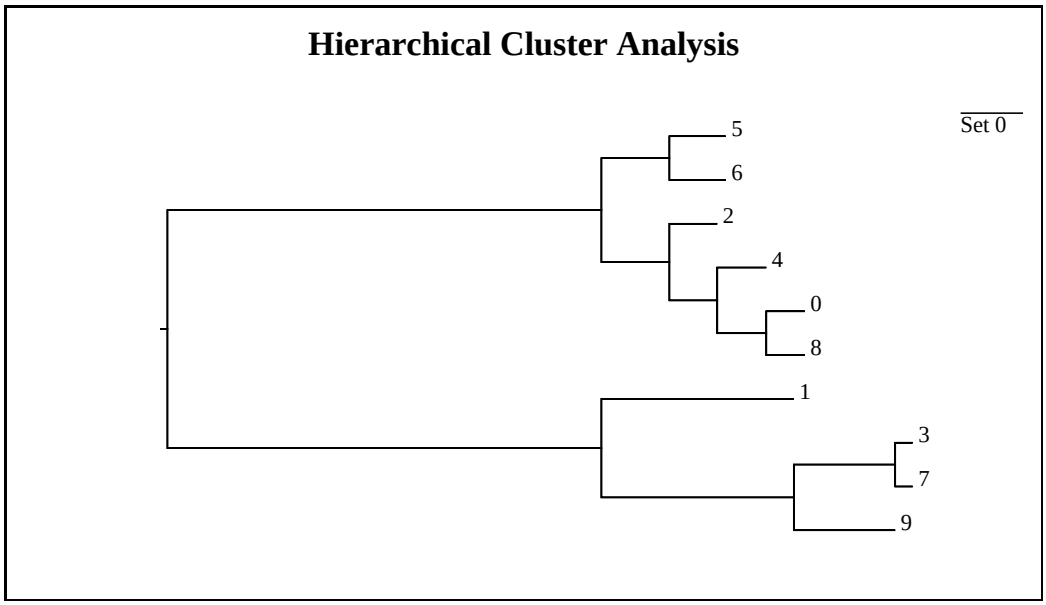


Figure 7.6: Cluster diagram for *in*, learned together with *through*

hierarchical cluster analysis performed on those vectors, respectively. The activation patterns in Figure 7.5 are of two general forms, one of which appears for movies 0, 2, 4, 5, 6, and 8, and the other of which appears for movies 1, 3, 7, and 9: thus, these two forms can be seen as corresponding to the two senses outlined above. Similarly, Figure 7.6 reflects the fact that the primary distinction made here is between these two sets of movies.

So when *in* is learned in conjunction with *through*, the activation patterns of the top hidden layer of the system reflect the polysemous structure of *in* – or, perhaps more accurately, reflect our intuition that the two senses outlined above should be considered distinct even though there is an abstraction which covers the two. And this separation does not occur when *in* is learned alone. While it might not be obvious ahead of time that this would be the case, it makes perfect sense: learning *through* forces the system to bring information concerning inclusion (or lack thereof) from earlier parts of the movie into its top hidden layer representation, since inclusion occurs in mid-movie for *through*. And that would serve to differentiate the two senses of *in*.

This experiment, then, has shown how the *paradigmatic context* of a linguistic form – i.e. that contrast set of linguistic forms that might grammatically be used in its place – can affect the way in which the polysemy of a single word is learned. The rest of this discussion of polysemy will be devoted to describing experiments which study the effect of the *syntagmatic context* as well – those words which might appear together with a given word in an utterance.

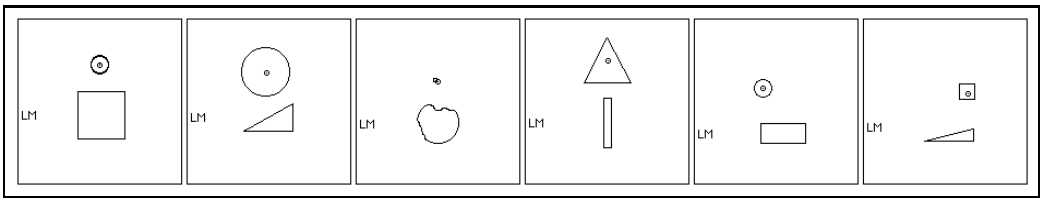


Figure 7.7: Positive examples of *be over*

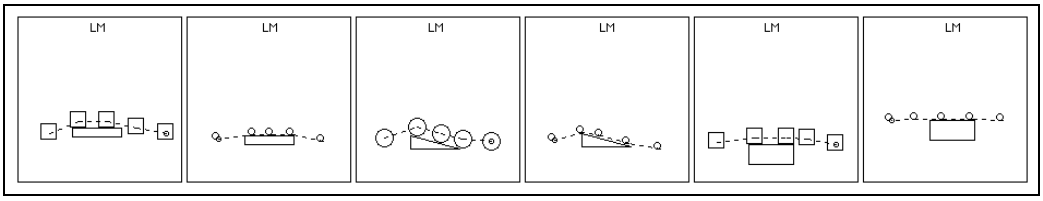


Figure 7.8: Positive examples of *go over*

These experiments were run by Jordan Zlatev, one of the students in the course mentioned above [Zlatev, 1992]. They centered on the acquisition of polysemous semantics for the English prepositions *over* and *under*, when accompanied by the three verbs *be, go,* and *fly*. Zlatev made a number of architectural modifications to the system presented in this thesis, and trained it on the six spatial concepts *be-over, go-over, fly-over, be-under, go-under*, and *fly-under*. Examples of these six concepts are shown in Figure 7.7, Figure 7.8, Figure 7.9, Figure 7.10, Figure 7.11, and Figure 7.12, respectively. Notice that the distinction between *going* and *flying* in this training set is one of contact: *going* allows contact between trajector and landmark, while *flying* does not. The system was trained using explicit negatives for each of these concepts, and these negatives explicitly disallowed contact in the case of *fly*, but not in the case of *go*, so although all the positive instances of *go over* and *go under* shown here involve contact, a valid generalization from these sets need not restrict itself to motion with contact. It is also worth pointing out that there are two "senses" of *go under*: one in which the trajector moves to a position underneath the

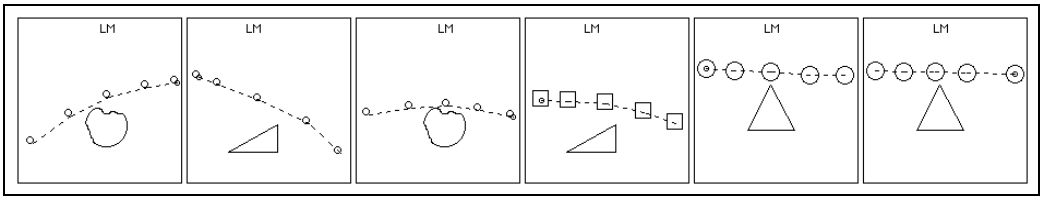


Figure 7.9: Positive examples of *fly over*

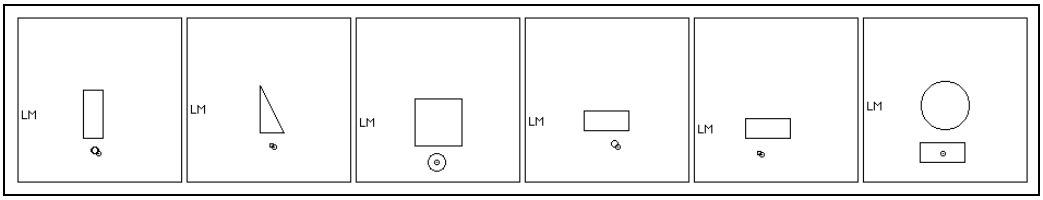


Figure 7.10: Positive examples of *be under*

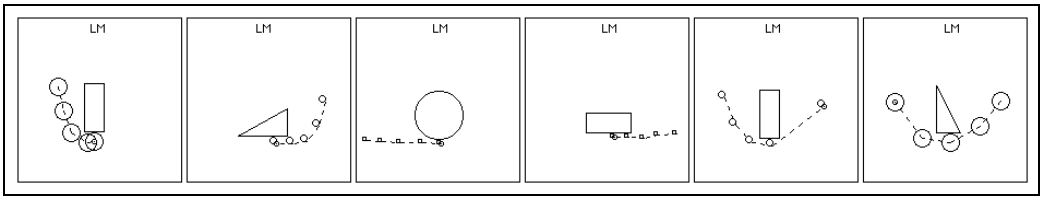


Figure 7.11: Positive examples of *go under*

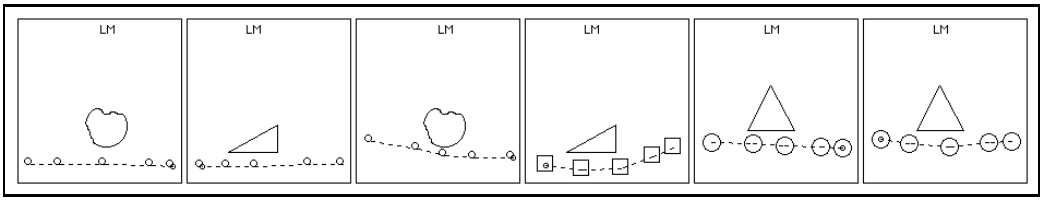


Figure 7.12: Positive examples of *fly under*

landmark, and another in which the trajector moves from one side of the landmark to the other, passing underneath it. Note that there is no corresponding pair of senses for *go over*, which can only mean motion from one side of the landmark to the other, while passing through the region above it.

The polysemy of *over* has been studied in detail [Brugman, 1981], and appears to be exceedingly complex. For our purposes here in this preliminary study, it suffices to note that there is no obvious abstraction that covers all the senses of *over* without also including examples that we would not consider *over*. Rather, *over* can be characterized as a disjunction of related senses, such as being-over, and motion-over (either with or without contact).[3] Thus, the situation here differs from the case of *in*, discussed above, for which such an abstraction does exist.

Figure 7.13 presents the final architecture which was used. This is similar to the standard architecture presented in Chapter 6, but whereas the standard architecture had each output node fully connected to the hidden layer marked *HL*, here there are a number of added structures. Specifically, there are five added hidden layer segments, in parallel, each receiving input from the *HL* hidden layer. These are marked *(over), (be), (go), (fly),* and *(under)* in the figure, since the aim is that representations of the semantics of these individual words will be formed in these hidden layers. Finally, there are six output nodes, with the *be-over* output node fully connected to the two hidden layers marked *(be)* and *(over)*, and not to any others. The other output nodes receive input in an analogous fashion. The central idea here is that if the *(over)* hidden layer supplies input to exactly those output nodes that involve the concept *over* in some way, the representations formed in that layer will reflect the manner in which the system has categorized *over*, and similarly for the buffers.

In addition, there were two Θ-nodes added to the Θ-node layer in the frame analyzer, to assist in detecting motion from one side of the landmark to the other. One of these compares the current proximal orientation to the proximal orientation at the first frame of the movie, and returns a maximal response when the two are diametrically opposed to one another.[4] Thus, if there is a strong response for this feature at the last frame of the movie, this is evidence that the trajector is on the other side of the landmark from its starting position. This situation is illustrated in Figure 7.14, which shows a circular trajector flying over a rectangular landmark, from left to right. Also shown are the proximal orientation at the beginning and then at the end of the event. Here, the proximal orientations at the beginning and end are diametrically opposed, so the Θ-node will respond strongly at the last frame of the movie.

[3]The fact that contact between trajector and landmark is allowed for *over* when the trajector is in motion, but not when it is static, is an indication that no single simple abstraction will serve to distinguish positive from negative examples of *over*.

[4]This is a somewhat unusual use of the proximal orientation, since it has up until now been used strictly as a relational orientation. Here it is both a relational and a reference orientation.

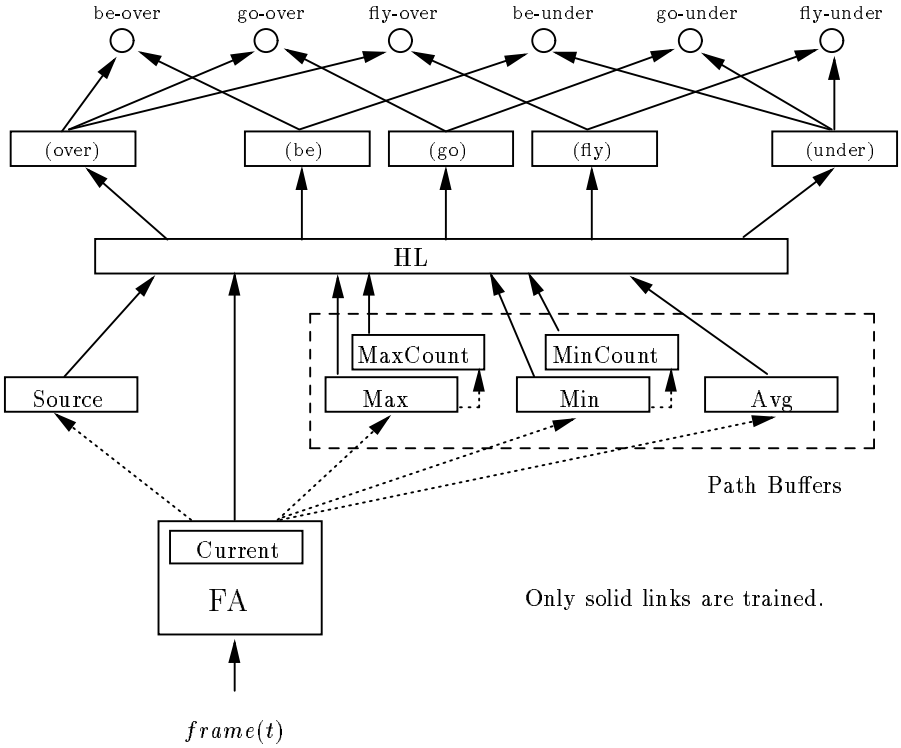


Figure 7.13: Learning *over* and *under* with *be, go,* and *fly*

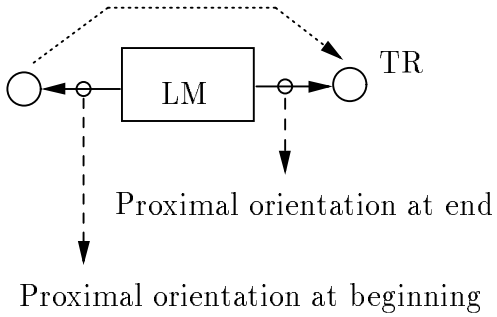


Figure 7.14: Detecting motion from one side of the landmark to the other

---

The other added Θ-node performs an analogous comparison for the center-of-mass orientation.

This architecture was trained on the six concepts presented above. Zlatev does not report the number of epochs to convergence, but states that the system performed well, exhibiting good generalization to novel test cases. For our purposes here, the topic of central interest is the representations formed in the hidden layers, and the manner in which they encode the polysemy of the words learned. We now move on to examine these representations.

Figure 7.15 presents a test set for *over*, and Figure 7.16 presents the results of hierarchical cluster analysis on the contents of the *(over)* buffer, for each of the movies in the test set shown. The primary distinction the system makes, as reflected in the cluster diagram, is between movies that involve the trajector remaining above the landmark, either immobile or moving slightly (this is the upper branch in Figure 7.16), and those that involve motion of the trajector from one side of the landmark to the other (the lower branch). Within the upper branch there are two subdivisions, one corresponding to those movies in which the trajector is immobile, and the other to movies in which the trajector moves slightly while above the landmark. No distinction is made within the lower branch of the diagram between movies that involve contact and movies that do not.

So these representations form an intuitively fairly natural-seeming set of "senses". Recall that these representations were shaped by learning *over* and *under* while they were paired with the verbs *be, go,* and *fly*, so this experiment illustrates the emergence of polysemes when words are learned in their syntagmatic, as well as their paradigmatic, context. Bearing this in mind then, it should not be surprising that the movies cluster as they do, since the upper branch of Figure 7.16 can be seen as corresponding roughly to *be-over*, while the lower branch can be seen as corresponding to *go-over* and *fly-over*. This dependence of the emergence of senses on syntagmatic context can also be found in other computational work on polysemy [Munro *et al.*, 1991; Harris, 1991].

Examination of the *(go)* buffer reveals verbal polysemy of a sort. Figure 7.17 presents a test set for *go*, made up of movies of the following four general types:

- Movies showing motion to a position under the landmark: numbers 0 through 3
- Movies showing motion under the landmark to the opposite side: numbers 4 through 8
- Movies showing motion over the landmark, either with or without contact: numbers 9 through 14
- Movies showing motion to a position above the landmark: numbers 15 and 16

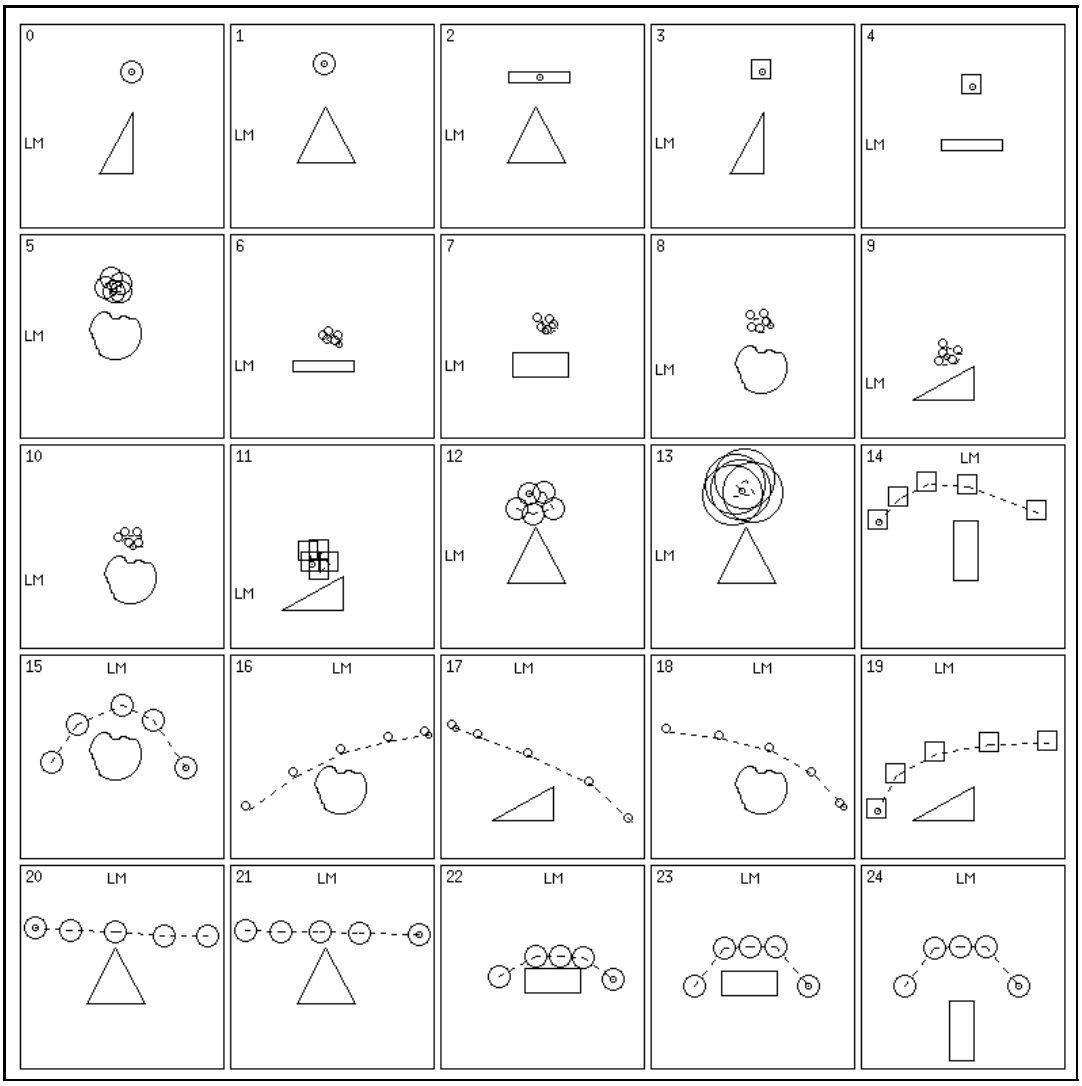


Figure 7.15: A test set for *over*

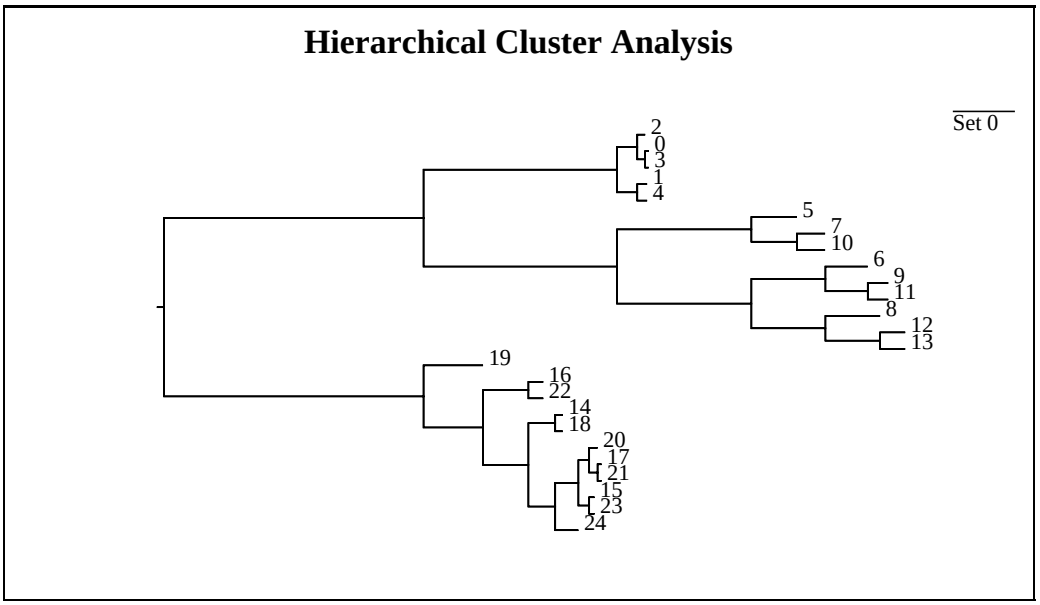


Figure 7.16: Cluster diagram for contents of the *(over)* buffer

Figure 7.18 presents the results of hierarchical clustering on the contents of the *(go)* buffer, for each of the movies just shown. As a glance at the diagram will reveal, the primary distinction being made here is between movies that depict the trajector going *through* a region, such as the region above or below the landmark (this is the top branch in Figure 7.18), and movies that depict the trajector going *into* such a region (the lower branch). Within the top branch, there is a further division into motion through the region above the landmark and motion through the region below the landmark. There is no such natural division within the lower branch.

So just as learning prepositions and verbs together gives rise to polysemy within the representations for prepositions, learning the two together gives rise to polysemous structure in the representations for verbs as well. Thus, the overall lesson from these experiments, together with the earlier ones regarding *in*, is the centrality of context, both paradigmatic and syntagmatic, in the emergence of polysemy.

7.1.2 Deixis

Adam Jacobs and Sarah Taub, two students from the course, extended the system by adding a deictic center to each movie, so that the semantics for the deictically anchored senses of prepositions such as *behind* could be learned.[5]

Figure 7.19 and Figure 7.20 present positive and negative examples of the deictic

[5]Recall from Chapter 2 that there are also uses of *behind* which are not deictically anchored, but which rather make reference to an intrinisic front/back orientation of the landmark object.

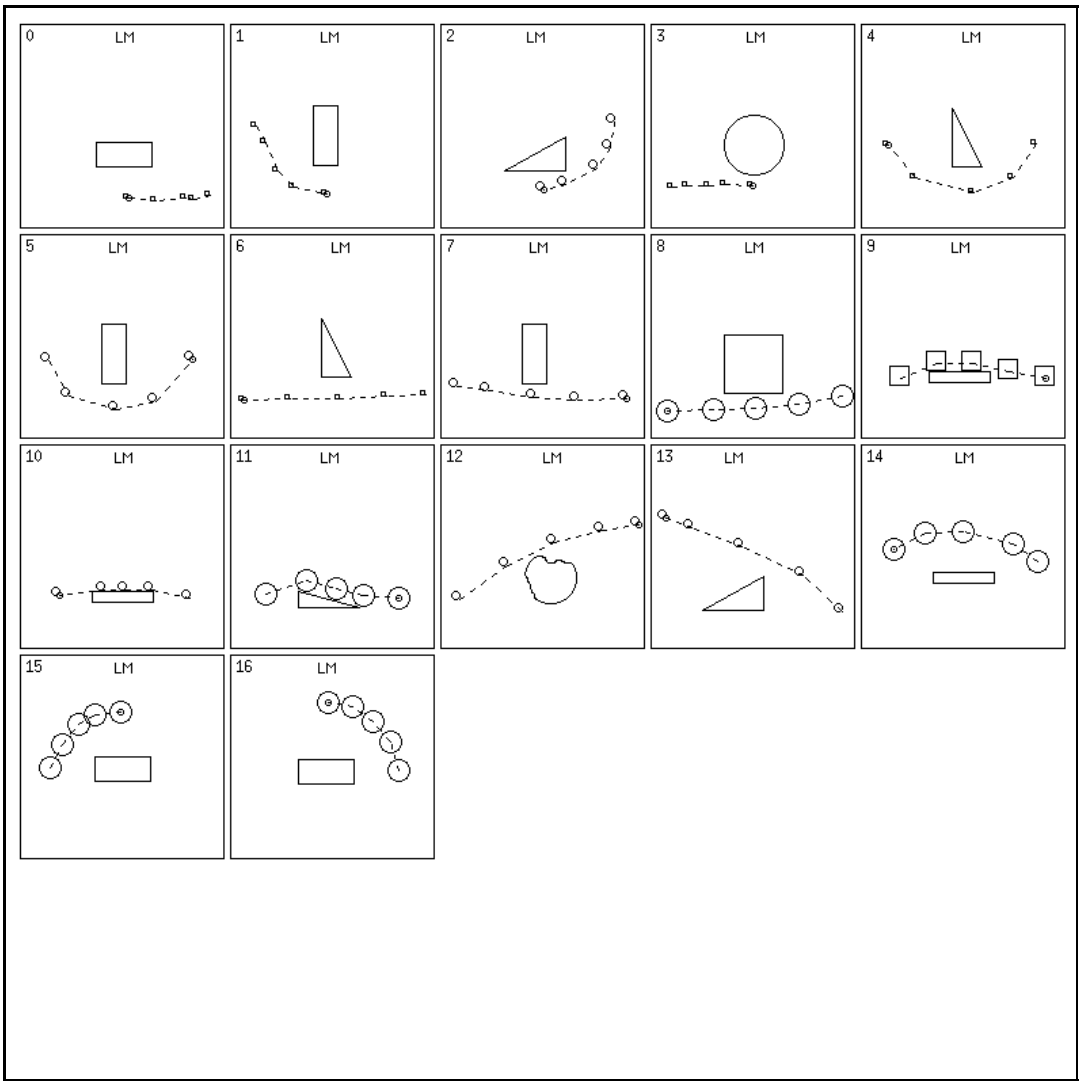


Figure 7.17: A test set for *go*

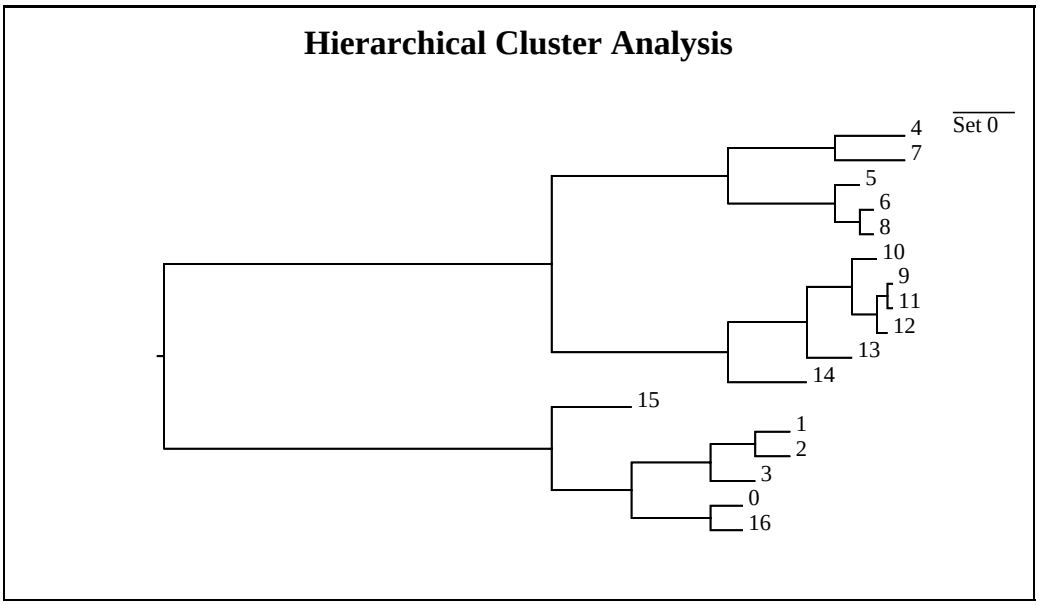


Figure 7.18: Cluster diagram for contents of the *(go)* buffer

sense of the English preposition *behind*. Each of these is a movie of length one, i.e. a static scene. Jacobs and Taub chose to assume a bird's eye view of the world, so the scenes shown here are to be seen as if one were looking down upon a set of objects on a surface. Notice that in addition to the landmark and trajector, each scene contains a deictic center with an inherent front. This deictic center is marked by two lines, labeled "F" and "R", for "front" and "right", respectively, which indicate the orientation of the deictic center. For example, in all the scenes in Figure 7.19, the deictic center is facing toward the landmark, as the line labeled "F" points toward the landmark.

Extending the system to handle deictic centers involved the introduction of a number of new orientational features. Specifically, the following relational and reference orientations were added:

- Relational orientations:
  - Orientation of the directed line segment from the deictic center to the closest point on the landmark
  - Orientation of the directed line segment from the deictic center to the closest point on the trajector
  - Orientation of the directed line segment from the deictic center to the center of mass of the landmark
  - Orientation of the directed line segment from the deictic center to the center of mass of the trajector

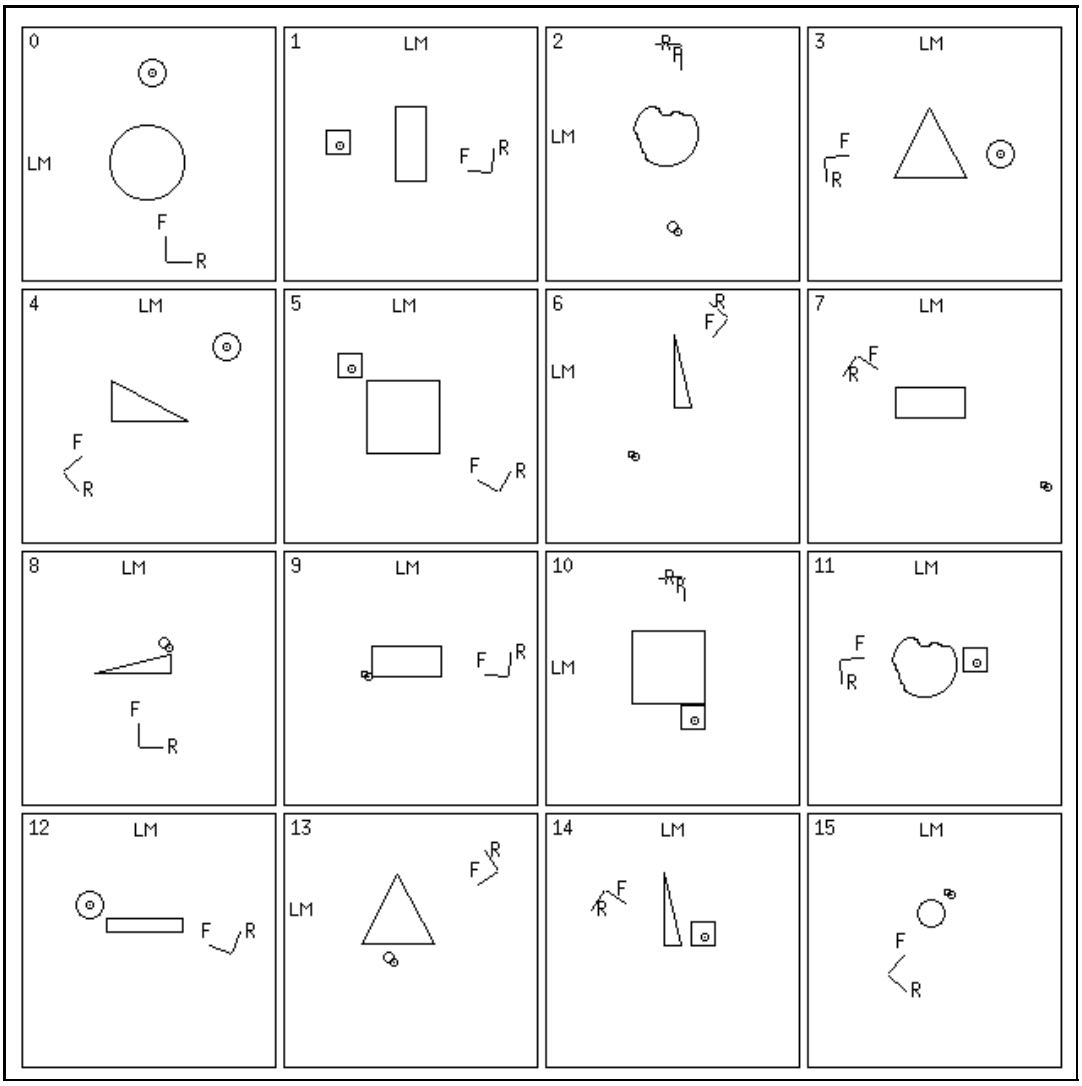


Figure 7.19: Positive examples of *behind*

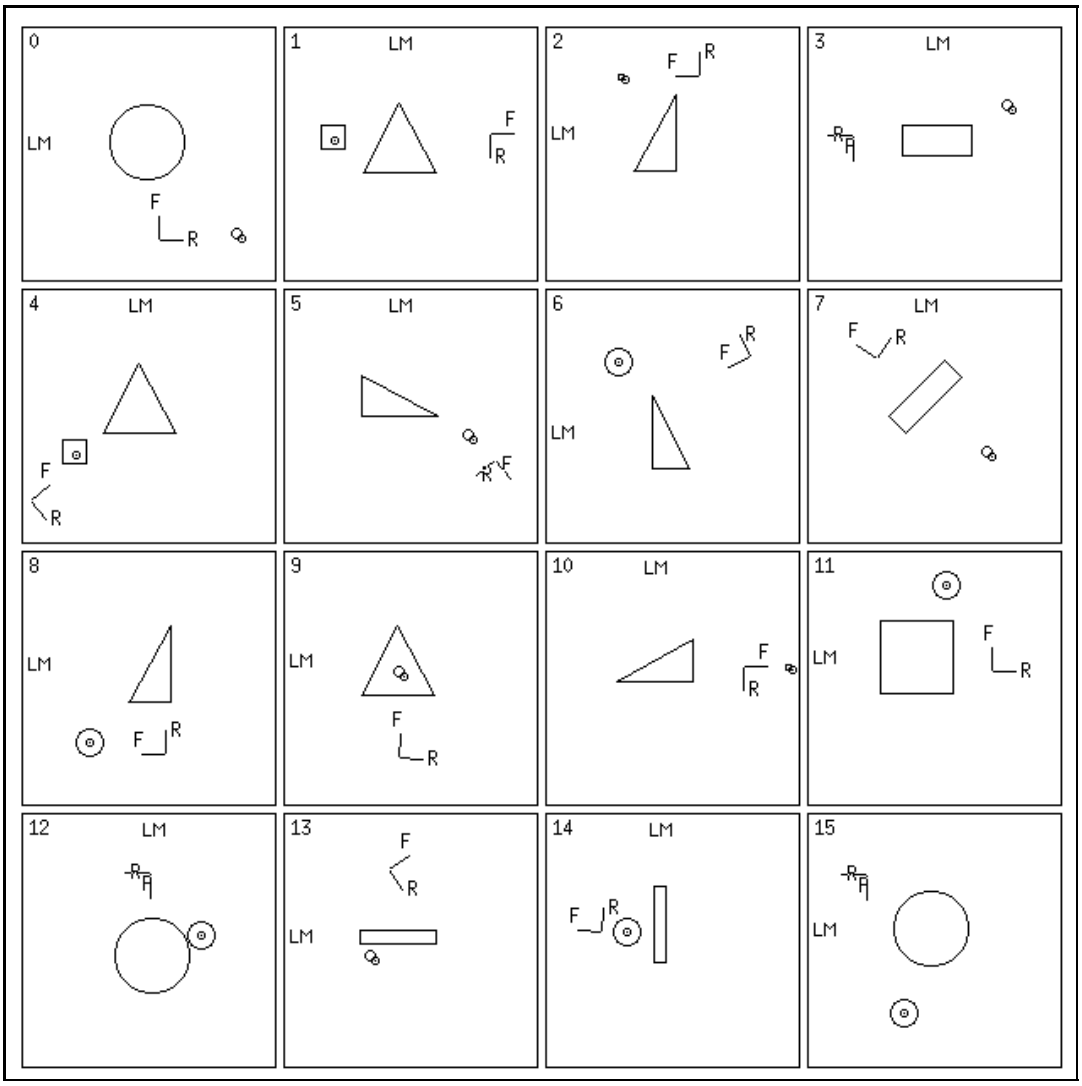


Figure 7.20: Negative examples of *behind*

- Reference orientations:
  - Inherent orientation of the deictic center

Θ-nodes were added so that each relational orientation was compared to each reference orientation. The resulting architecture was trained under quickprop, which yielded convergence to under 0.01 total summed squared error within 40 epochs. Figure 7.21 shows the system's performance on a test set for *behind*. Notice that the deictic center must be facing the landmark and trajector for the scene to be classified as *behind*. Consider scenes 8 and 9 for example; here, the trajector is on the other side of the landmark from the deictic center, but the deictic center is facing the wrong way, so the scene is considered an extremely poor example of *behind*.

One interesting avenue for further research is examining polysemy within a deictically anchored system. For example, there are deictically based senses of *over* that have not been handled yet (as in the sentence "The circle is *over* the square from here"); with the addition of a deictic center to the system, this sense could perhaps be learned along with the other senses that the system is already able to learn.

7.1.3 Force Dynamics

[Talmy, 1987] has focused attention on the "force dynamic" component of semantics: how entities interact with respect to force, noting that concepts such as exertion of force, resistance to force and the like are pervasive across a number of semantic domains, both physical and non-physical. This section examines uses of the system to address the force-dynamic issue of *potential motion* in static scenes, and the role it may play in the semantics of words such as English *on*.

In Chapter 2 we discussed the work of Jennifer Freyd and coworkers, which presents psychological evidence that humans viewing static scenes mentally represent the forces that are acting on objects in the scene, keeping them where they are. For instance, viewing a potted plant sitting on a table, a subject would mentally represent the force dynamics of the support relationship between table and pot, explicitly including some representation of the fact that if the table were not there, the pot would fall. In the words of [Freyd *et al.*, 1988], "subjects represent the underlying dynamics of coherent static scenes, so that when the forces are suddenly unbalanced, the mental representation includes an unfreezing of the potential motion implicit in the scene."

The computational work of Jeffrey Siskind [Siskind, 1991; Siskind, 1992], taking the work of Freyd *et al* as motivation, has incorporated this notion in a program which uses a process of *counterfactual simulation* to detect support in scenes. That is, the system simulates what would in fact happen if objects in the scene were to fall to positions of lower potential energy, and if an object moves under this simulation, the system concludes that that object is not supported.

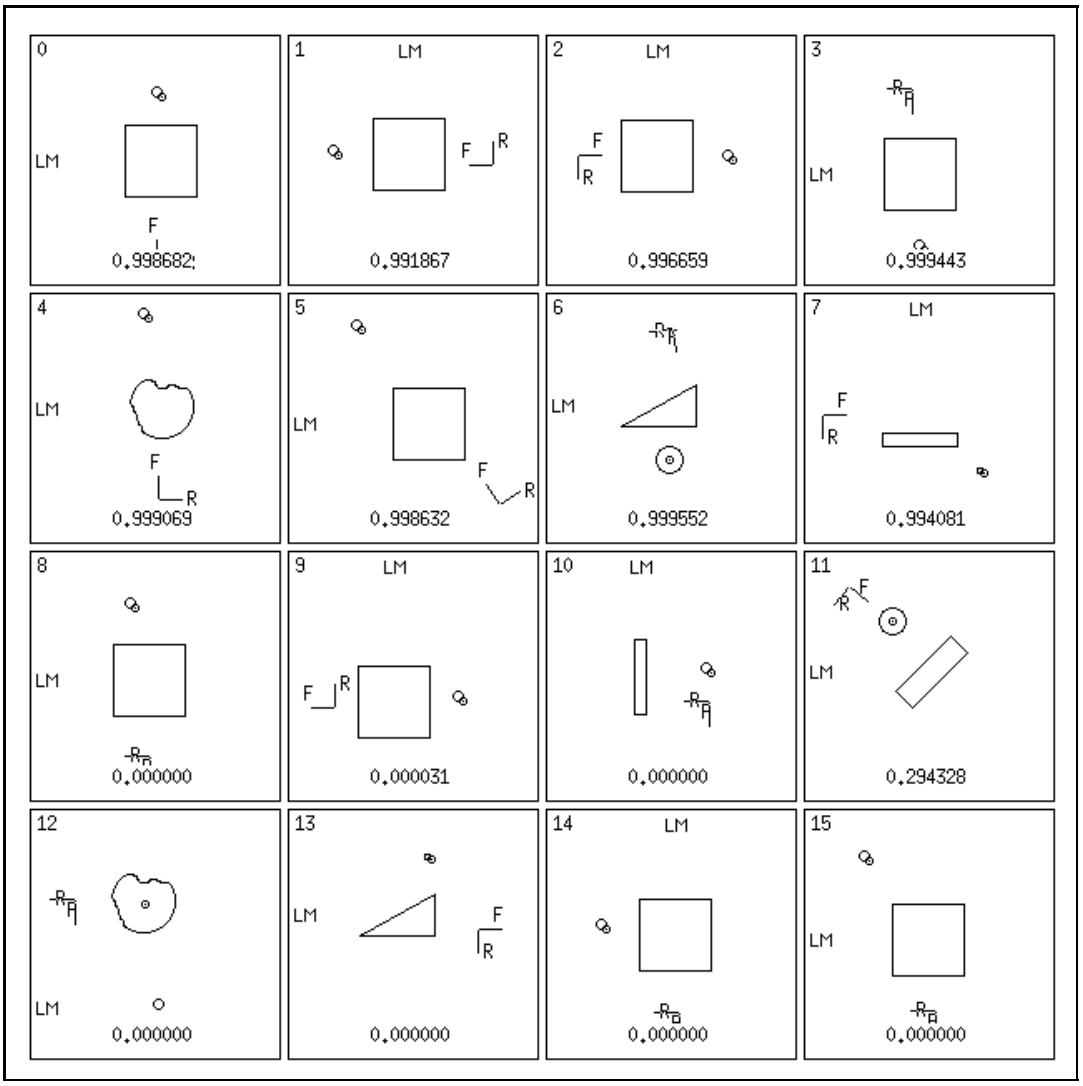


Figure 7.21: A test set: *behind*

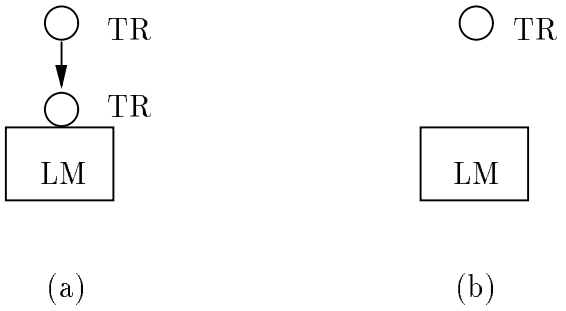


Figure 7.22: Potential motion: English *above*

This section presents the work of Collin Baker, one of the students in the course, that uses the system described in this thesis to model such a counterfactual imaginative capacity. The idea here has been to construct movies in which the motion of the trajector is taken to correspond not to actual motion of an object in the world, but rather to the *unfrozen potential motion* of the trajector. Thus, a movie showing a circle dropping down onto a square (as in Figure 7.22(a)) represents not the fact that a circle actually dropped onto a square, but rather the fact that a circle, which is located above a square (as in Figure 7.22(b)), *would* drop onto the square under the force of gravity. Thus, part of the semantics of static *above* is taken as "the trajector would fall onto the landmark if released".

Similarly, potential motion plays a central role in the semantics of the English preposition *on*. Consider Figure 7.23 and Figure 7.24, which present positive and negative examples, respectively, of *on*. Each of the movies shown is a two-frame movie, although in Figure 7.23, the two frames are always identical. The idea here has been to capture the intuition that if a trajector is supported by a landmark, either via support from below or by adhesion, it can be said to be *on* the landmark. Compare, for example, movie 1 in Figure 7.23 and movie 0 in Figure 7.24. These two depict identical situations, a small circle touching the side of an upright triangle, except in one case the circle adheres to the triangle (movie 1 in Figure 7.23), and is thus considered to be *on* it, while in the other case, we see that the circle would fall if released, and thus is not stuck to the triangle, and is thus not *on* it.

Given training sets somewhat larger than the sets of positive and negative examples shown above (19 positives and 17 negatives total), the system converged to under 0.01 error in 70 epochs. Figure 7.25 presents the system's performance on a test set; all the movies shown are classified correctly. Notice that the role of adhesion can be seen in the responses given for movies 1 and 6. These two show the same situation, but in one case the trajector adheres to the landmark (movie 1), causing

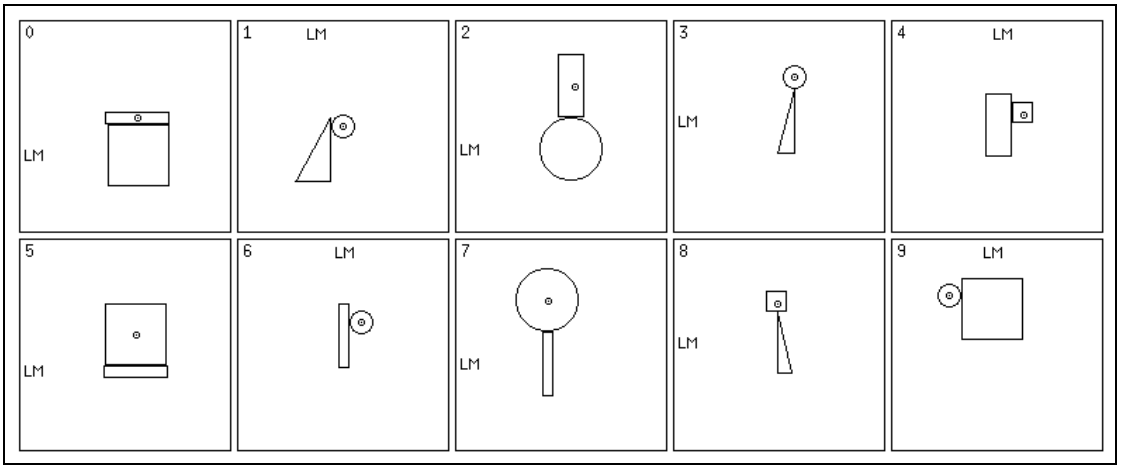


Figure 7.23: Positive examples of *on*

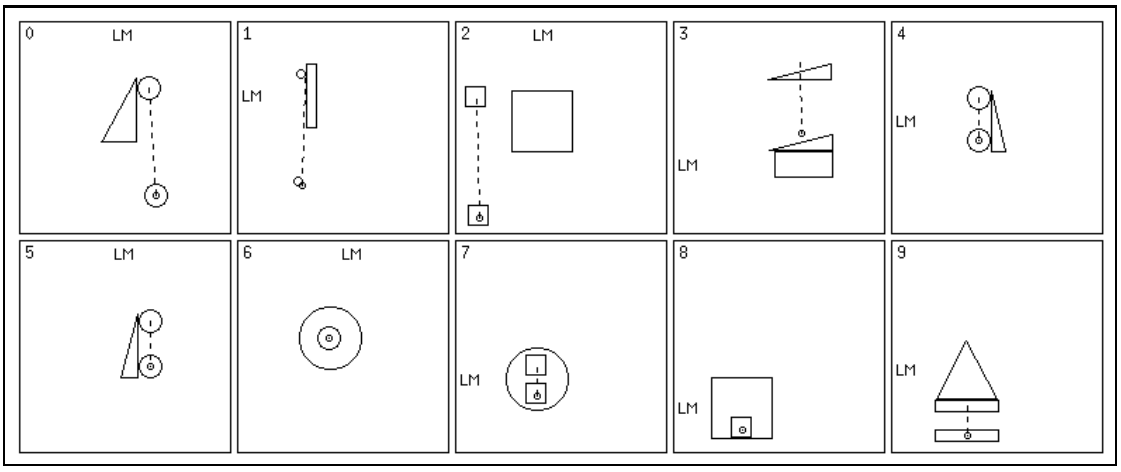


Figure 7.24: Negative examples of *on*

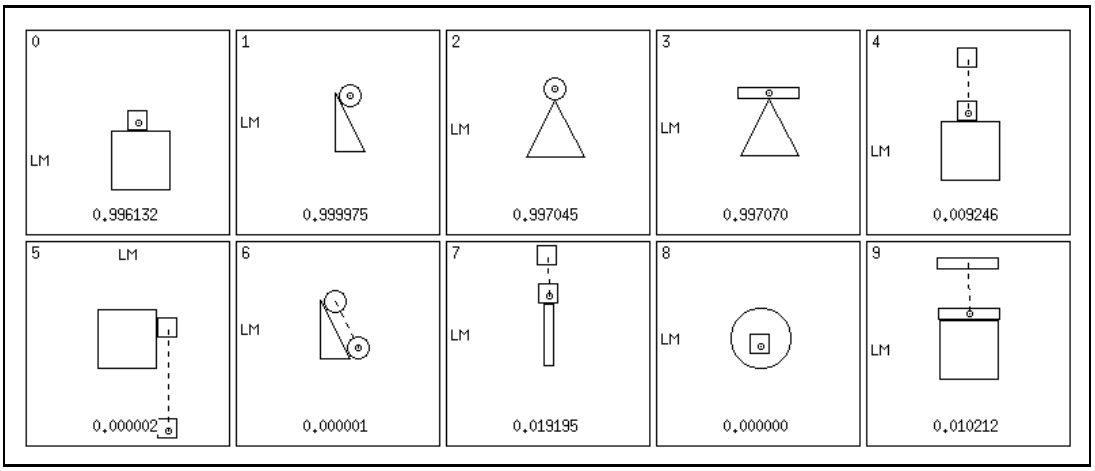


Figure 7.25: A test set for *on*

the situation to be classified as *on*, while in the other case it does not, but rather rolls down the side of the triangular landmark. This latter situation is thus not classified as an example of English *on*.

7.1.4 Reified Paths and Key Events

Two alternative means for handling motion were explored, the method of *reified paths*, and the method of *key events*.

Reified Paths

Narciso Jaramillo and Michael Schiff extended the system to incorporate the idea of *reified paths* for motion. The idea here is that instead of keeping running statistics such as the maximum and the average over static features from individual frames (such as is currently done using the path buffers), one could simply use an explicit, static representation of the path as a whole. This is illustrated in Figure 7.26. Here, (a) shows a small square moving over and around a circle. In the current system, the maximum, minimum, and average of the static features extracted from each frame of the movie would serve as a representation of the path. Under the reified path scheme, however, we construct a bitmap-based representation of the area that the trajector would cover in its trajectory, by "sweeping" a filled version of the trajector from its position in one frame to its position in the next. Figure 7.26(b) depicts this new construct for the event shown in (a), and (c) shows the convex hull of this reified path, which was also used in this modification to the system.[6] The convex hull was

[6]The convex hull of a set X of points is the smallest set of points that is convex and contains X. Intuitively, if one were to stretch a rubber band around a set of points, the rubber band would

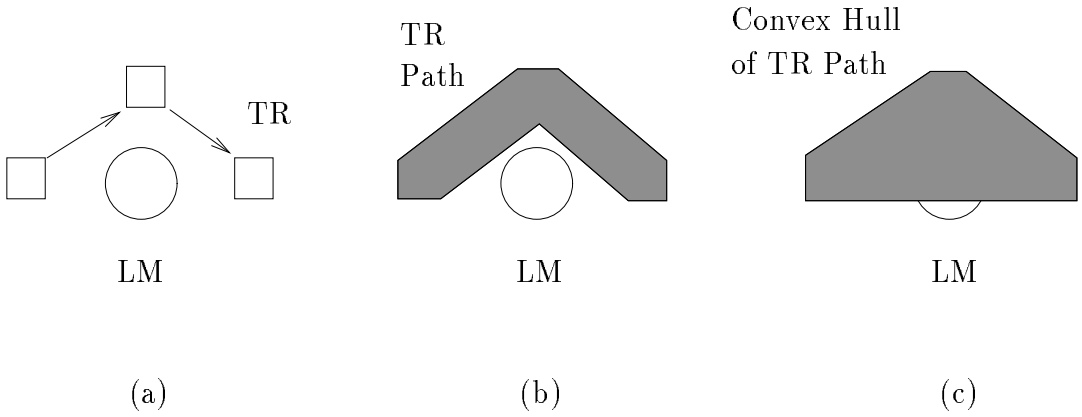


Figure 7.26: Reified paths

---

used since it appears to play a role in the semantics for prepositions such as *around*: one possible partial characterization of *around* is that the convex hull of the path must overlap the interior of the landmark to a large degree, as shown in Figure 7.26(c).

Jaramillo and Schiff added two feature map structures to the frame analyzer, each of which is of the form shown in Figure 7.27(a). These structures are isomorphic to the feature map devices introduced and discussed in Chapter 5, and are characterized by feature maps whose units receive input from highly localized receptive fields in a *viewed map*, gated by units at the corresponding position in a *gating map*. Jaramillo and Schiff added two such feature map structures, so the modified frame analyzer contained a total of three, receiving input as follows:

- Viewed map contains the landmark interior; gating map contains the trajector boundary (this was in the original system)
- Viewed map contains the trajector path (see Figure 7.26(b)); gating map contains the landmark boundary
- Viewed map contains the convex hull of the trajector path (see Figure 7.26(c)); gating map contains the landmark interior

The overall resulting architecture, with these feature map structures in the frame analyzer, is portrayed in Figure 7.27(b). Note that Jaramillo and Schiff have done away with the path buffers altogether, since the feature map structures now take the place of that path representation. There is still a source buffer, however, holding the

provide the outline of the convex hull of the set of points.

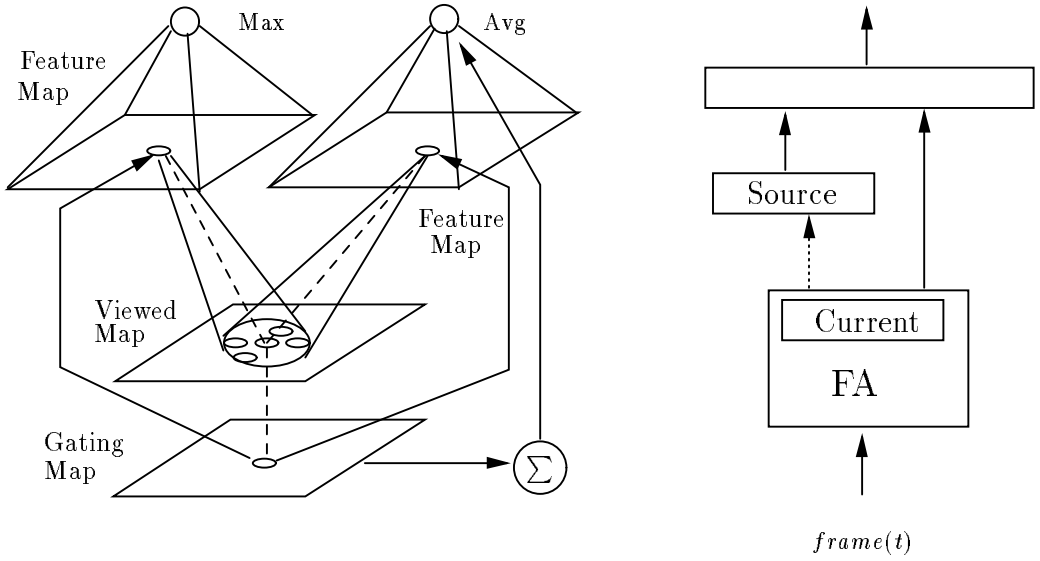


Figure 7.27: Modifying the architecture to incorporate reified paths

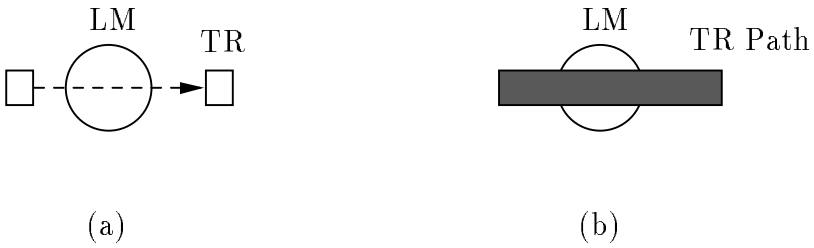


Figure 7.28: Implied intersection in a 2-frame movie for *through*

output of the frame analyzer at the first frame of the movie. Note that although non-directional features of the path, such as contact between the convex hull of the trajector path and the landmark, are detected in the static path representations we have been discussing, directional features such as proximal or center-of-mass orientation are still dealt with on a frame-by-frame basis, and appear in the *Current* buffer of the frame analyzer at each time step, along with the path representation. The source buffer is retained so as to be able to bring into play the values of these directional features at the beginning of the movie.

Jaramillo and Schiff tested this architecture on a number of English prepositions, including *through.* One noteworthy outcome of this testing is that, unlike the original system, the modified system was able to learn to classify two-frame movies such as the one shown in Figure 7.28(a) as *through*, even though there is no frame in the movie which actually shows the trajector inside the landmark.[7] This is possible because the static path representation for a movie like this one would look like the path shown in Figure 7.28(b), which does intersect the landmark. The feature map structures added to the architecture learn to detect this, allowing the system to classify as instances of *through* two-frame movies with implicit intersection, as well as movies with explicit intersection.

Key Events

David Bailey examined the idea of using *key events* in the analysis of motion. Under this scheme, the source and path buffers in the current architecture would be replaced by a representation of a series of key frames in the movies, frames which carry the bulk of the semantic content of the movie. Consider for example Figure 7.29, which portrays two examples of a circle moving into and then out of a square. In this figure, (a) does not appear to be a good example of English *through*, while (b) does,

[7]It is possible that the original system would also be able to learn to do this, if modified with the "side-to-side" detection mechanism discussed above in §7.1.1, and illustrated in Figure 7.14.

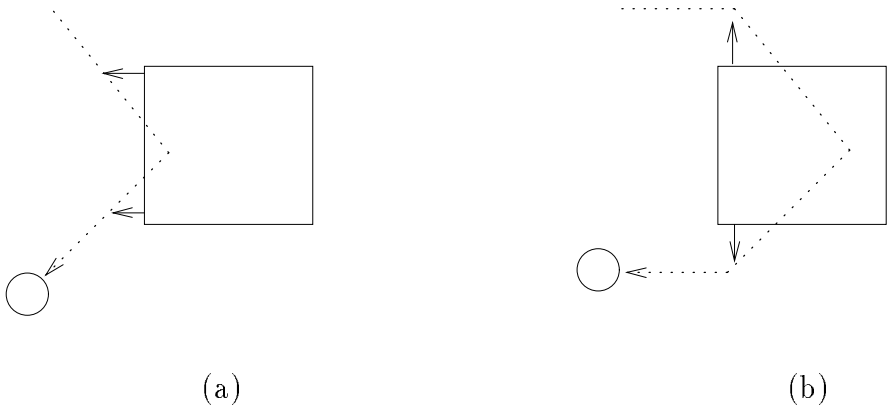


Figure 7.29: The role of key events and proximal orientation in English *through*

---

although the circle starts and ends in the same position relative to the landmark in the two cases, and the angle formed by the trajectory is also the same in the two cases.[8] One thing which does differentiate these two, however, is the relation of the proximal orientation just prior to entering and just after leaving the landmark. These orientations are shown by the small arrows attached to the landmark in the figure. In (a), the proximal orientation right before entry and right after exit is essentially the same (i.e. entry and exit occur on the same side of the landmark), while in (b) they are diametrically opposed (i.e. on opposite sides of the landmark). Under this analysis, entry and exit from the landmark are key events, and the output of the frame analyzer for the time steps right before and right after these events should be saved in buffers, much the way the first frame is stored in the *Source* buffer now, so that comparisons of the sort described above can be effected.

7.2 Possibilities for Future Work

7.2.1 Distance

One obvious – and hopefully easily rectifiable – shortcoming of the system as it stands is that it does not take into account any representation of the *distance* between trajector and landmark, other than the non-directional feature map representations. And these only detect relations in which the landmark and trajector are either touching or interpenetrating. In §5.2.2, we briefly introduced two features which could serve

[8] Note that this implies that the method for detecting motion from side to side which was presented above in connection with Figure 7.14 is inadequate.

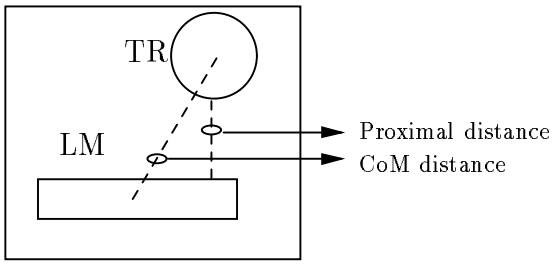


Figure 7.30: Proximal and center-of-mass distances

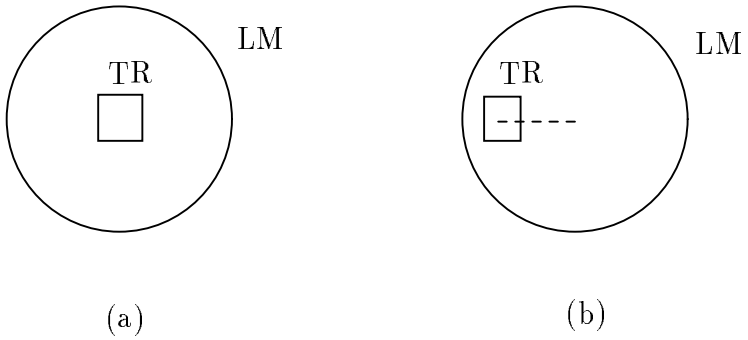


Figure 7.31: Good and poor examples of Russian *posredi* ("in the middle of")

to make up for this deficiency, and we review them once more here: these are the *proximal distance* and the *center-of-mass distance*, illustrated in Figure 7.30. These are the lengths of the directed line segments whose orientations define the proximal and center-of-mass orientations, respectively.

These features can be shown to play a role in the semantics of spatial terms other than *near, far,* and other overtly distance-expressing terms. For example, consider Figure 7.31, which illustrates a good and a poor example of the Russian preposition *posredi*, which expresses the notion "in the middle of". In (a), the good example, the center-of-mass distance between the circular landmark and the small square trajector is essentially zero, and is not shown. In (b), it is shown as a dotted line. So the center-of-mass distance is an excellent indicator of the degree to which the trajector is "in the middle of" – or *posredi* – the landmark.

However, there is more to even some very simple distance-related concepts than actual distance. For example, consider Figure 7.32. This figure illustrates the effect

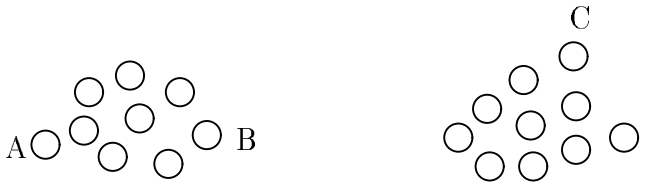


Figure 7.32: The effect of attentional focus on *near* and *far*

---

of the *focus of attention* within a scene on judgments of *near* and *far*. If one were to ask if circles A and B are *near* each other, the response would depend not only on the distance features outlined above, but also on whether or not both clusters of circles lay within the focus of attention, i.e. whether or not both clusters were being considered. If only the leftmost cluster is being considered, A is not very *near* B, but if both are being considered, it is, since A is clearly much nearer to B than either is to C.[9] The point here is that terms such as *near* and *far* refer not to absolute distance, but rather, to distance as compared to some implicit norm. Thus, we can say "Berkeley is *near* San Francisco", but not "Spenger's is *near* Coit Tower",[10] although the proximal distance is essentially the same in the two cases, and the center-of-mass distance is less in the case of Spenger's and Coit Tower. An explanation for this is that when considering distances between cities, a different norm is used than when considering distances between smaller entities such as restaurants and towers. Similarly, in the case of the circles above, a different norm for distance is used when considering just the leftmost cluster than when considering both clusters, presumably because the size of the focus of attention itself is larger in the latter case.

7.2.2 Regions as Objects

Just as we considered the idea of the trajector path as a reified object in §7.1.4, we may also want to consider treating regions of space, such as the region immediately above a landmark, as objects.[11] Once we do this, a number of analogies among motion terms become apparent. For example, we see that to go *over* a landmark is to go *through* the region above it; to go *past* a landmark is to go *through* the region near it; to pass *under* a landmark is to go *through* the region below it, and so on. Similarly, to go *to* a landmark is to go *into* the region near it; to go *under* a landmark is to go *into* the region below it, etc. The basic insight is that once we start viewing regions this

[9] Helmut Schnelle suggested this example.

[10] Spenger's is a restaurant in Berkeley, and Coit Tower is a tower in San Francisco.

[11] For other work in the L_0 project along these lines, see [Weber and Stolcke, 1990].

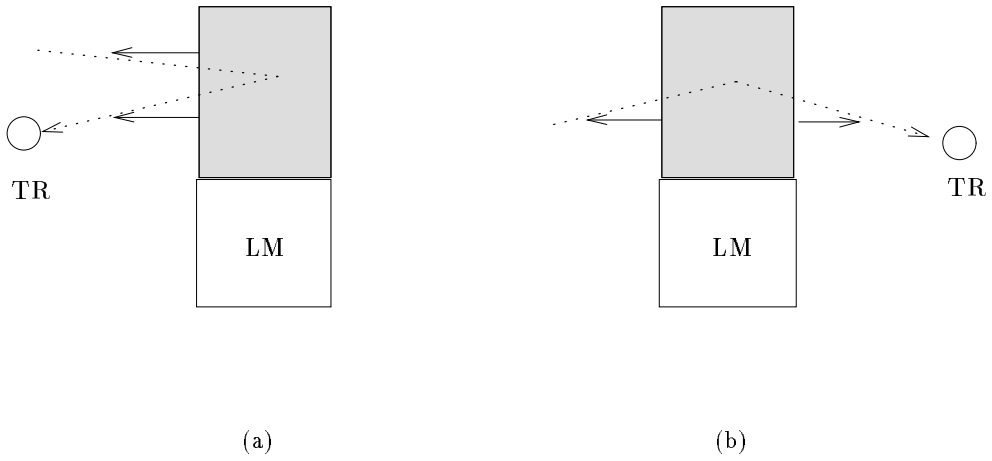


Figure 7.33: The *above* region as an object, and proximal orientations from it to the trajector

---

way, we find that much (though not all) of what language tends to encode regarding trajectories can be expressed as simply the selection of a region, and a trajectory type relative to that region. [Jackendoff, 1983; Jackendoff and Landau, 1991] list five trajectory types relative to regions, corresponding to motion *through, into, out of, toward,* and *away from* a region-object. These cover a good many non-English examples as well, some with no simple English counterparts. For example, Russian *iz-pod*, meaning "out from underneath", can be seen as simply the application of the *out of* trajectory type to the *below* region.

If we do this, it makes sense to detect motion *over*, for example, in a manner analogous to the way we detect motion *through*. Recall from the discussion in §7.1.4 that one way of detecting motion *through* – as opposed to simply motion into and then out of the landmark object, possibly on the same side – is to compare the proximal orientation just prior to entry into the landmark and just after exit (this was illustrated in Figure 7.29). Analogously, one could compare the proximal orientation *from the above-region to the trajector* just prior to the trajector's entry into the above-region, and just after its exit. This can be seen in Figure 7.33. The use of proximal orientations from the reified above-region to the trajector exactly parallels the use of proximal orientations from the landmark to the trajector in our earlier discussion of *through*.

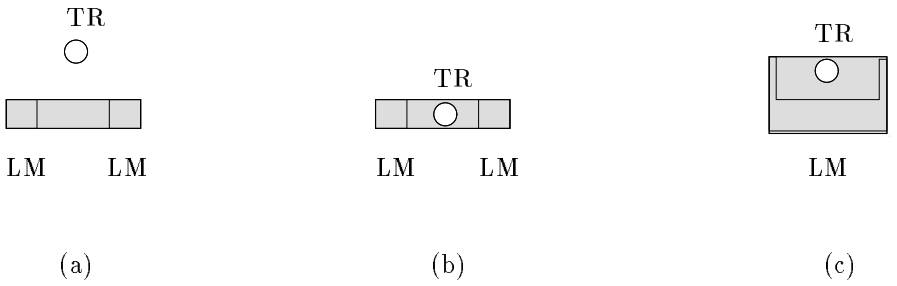


Figure 7.34: Using the convex hull of the landmark(s)

7.2.3 Convex Hulls

Another useful extension would be the incorporation into the system of explicit representations of the *convex hulls* of objects. We have already seen that Jaramillo and Schiff used the convex hull of the trajector path in their extensions to the system, and that this could play a role in the characterizations of such spatial terms as English *around*; we now note that the convex hull of a landmark or set of landmarks is also potentially a useful representation to include.[12]

Consider for example Figure 7.34. Here we see three scenes in which the convex hull of the landmark (or of the co-landmarks, as in (a) and (b)) appears to play a central role in the way in which the spatial relations shown are verbalized. In (a), we see two small square landmarks and a circular trajector above them. This seems an excellent instance of the English description

> The circle is *above* the squares

even though it is not a particularly good instance of the circle being *above* either the left or the right square taken alone. The idea here is that if we take the phrase "the squares" in the above sentence to actually refer to the convex hull of the two squares, analogs to the usual features we consider relevant to *above* will work here as well: the proximal and center-of-mass orientations *from the convex hull* to the trajector both align with upright vertical.

Figure 7.34(b) shows two square landmarks again, and a small circular trajector *between* them. The convex hull of the landmarks provides a simple means of characterizing the region between two objects (or the region in which a trajector could be said to be *among* a number of landmark objects): the degree to which a trajector lies

[12]Note that this is by no means a novel idea; [Herskovits, 1986] has also noted the utility of such a construct, and [Weber and Stolcke, 1990] present earlier work in the L_0 project which used the convex hull in the computation of spatial relations among objects.

(a) (b)

Figure 7.35: Implicit paths: The circle is above and to the left of the square

---

within the convex hull, but not within the landmarks themselves, is an indication of the degree to which it is *between* the landmarks.

Finally, Figure 7.34(c) shows a non-convex U-shaped landmark, and a trajector located such that the proximal and center-of-mass orientations from the landmark to the trajector are both aligned with upright vertical, but also located such that one would not describe it as *above* the landmark. This is another example in which the relevant features for *above* might be the proximal and center-of-mass orientations from the convex hull of the landmark(s) to the trajector, as discussed in reference to (a) above. This thesis has restricted its attention to convex objects, for which the object and its convex hull are co-extensive, so this distinction has not been critical yet. It clearly will be once non-convex objects and multiple landmarks are considered.

7.2.4 Implicit Paths

There are a number of phenomena which are not covered by any of the extensions we have discussed so far, and which seem to call for the introduction of *implicit paths*. Figure 7.35(a) depicts a situation which is well described by the sentence

> The circle is *above and to the left of* the square.

However, the location of the circle is at best a weak example of each of *above the square* and *to the left of the square*. One explanation that seems credible is that the phrase *above and to the left of* in fact describes an implicit path directing the listener how to get to the trajector (from the landmark). This is shown in Figure 7.35(b).

Implicit paths also help to make sense of some senses of a number of English prepositions. Consider for example the use of the preposition *across* in

> Kim is *across* the table (from me).

Here, while Kim is probably not in fact physically lying across the table, the preposition is used because the implicit search path that would take me from where I am to where Kim is passes across the table. This phenomenon is referred to as *end-point focus* [Brugman, 1981; Lakoff, 1987], since the trajector is located at the end-point of an implicit search path.

Another example which seems to involve an implicit search path, although not through end-point focus, is

> His office is *along* the corridor,

or alternatively

> His office is *along* here somewhere.

In these uses of *along*, the only thing that is actually *along* the landmark (the corridor in this case) is the imagined search path that will take the listener from where he is to the trajector.

This is related to Talmy's notion of *fictive motion* [Talmy, 1990]. Talmy notes that sentences such as

> Those rods *go through* the ceiling,

in which a verb of motion and a preposition are used to describe the static relationship of an elongated trajector relative to some landmark, can be explained by positing a mental scanning along the length of the trajector on the part of the speaker. The verb of motion and preposition then in fact refer to the relationship between the landmark and the moving focus of attention of the speaker along the trajector, not simply the trajector itself.

Of the possible directions for extension outlined in this chapter, this latter is the least fleshed-out. It is clear that some form of imputation to the scene of an entity which is not actually there is involved, and that once this is done, the usages of particular words to describe situations appear to make more sense. But the circumstances under which imputation of this sort takes place, the processes by which it happens, and the ramifications these would have for a modeling effort of the sort this thesis has concerned itself with have not been worked out.

In general then, we have seen a number of existing extensions from the core system, as well as an indication of further possible extensions, which would address some of the many relevant linguistic issues not covered by this thesis. Having now discussed the bulk of the thesis work and directions for further inquiry, we move on to a review of related research, and conclusions for the thesis as a whole.

Chapter 8

Perspective and Conclusions

8.1 Related Computational Research 183
8.1.1 Related Non-Learning Systems 184
8.1.2 Related Learning Systems 187
8.2 The Thesis in Retrospect 189

This chapter provides a perspective on the thesis taken as a whole. We begin by reviewing computational research which has addressed similar issues, and noting similarities with and differences from the approach used here. Following that, we conclude with a retrospective discussion of the thesis itself, recapitulating the central ideas, techniques, and results.

8.1 Related Computational Research

Since this thesis concerns the acquisition of visually-grounded semantics for spatial lexemes, it stands at the intersection of a number of subfields of artificial intelligence, most notably learning, vision, and natural language. A complete review of work in each of these fields is clearly beyond the scope of this thesis; I focus, therefore, on work that has taken a perceptually-based approach to the grounding of natural langauge semantics. Some classic concept learning research, such as the work of [Winston, 1977] in learning such geometric notions as the concept of an "arch", could be viewed as lexical semantics learning, but this work is not perceptually grounded – rather, the learning takes place within an ungrounded symbolic system.

Computational modeling of the acquisition of perceptually-grounded semantics is not a particularly time-honored field. Nonetheless, it does seem to have come into its own over the past few years, having fairly solidly established itself as a research program. This is in large measure attributable to a general *zeitgeist* in cognitive science as a whole, which emphasizes the importance of perceptual grounding for natural language semantics. In any event, as we will see below, there has been a flurry of recent computational research activity addressing issues very similar, if

not always identical, to the ones addressed in this thesis. We begin by examining research which is concerned with non-learning systems performing event perception and classification in various visually-based domains, and then cover work on systems which learn to perform such tasks.

8.1.1 Related Non-Learning Systems

There are a number of systems which perceive events with the purpose of producing some linguistic or semantic description of the events viewed, without learning to do so. These include the work of [Thibadeau, 1986] in the perception of actions, based on [Miller and Johnson-Laird, 1976], and the work of [Tsotsos, 1981], which presents an expert system for motion understanding in the context of the functioning of the left ventricle of the human heart. This builds on the earlier work of [Badler, 1975], which outlined a general framework for the production of natural language descriptions of movies of moving objects. In addition, other work [Waltz, 1981; Yuhan, 1991] has focused on the use of spatial language, but in the absence of visual grounding for semantics. More similar, however, to the work described in this thesis, are the work of the Landscan and Vitra projects, Jeffrey Siskind, and David Chapman. We now discuss each of these in turn.

The Landscan and Vitra Projects

[Bajcsy *et al.*, 1985; Hays, 1987] present the Landscan project, which is directed at the answering of natural language questions regarding visual input. This input is received by cameras trained on a model of a city block. [Hays, 1987] in particular focuses on the natural language aspect of the problem, and describes a program which accepts as input a sentence containing a locative expression, and produces an interpretation of the sentence suited for the interface with the vision system. The system is restricted to handling static spatial relationships, and thus does not cover motion. Hays uses as her theoretical framework the work of [Herskovits, 1986], and her program can be viewed as an implementation of some of the ideas put forth in Herskovits' book. One aspect of particular interest in her system concerns object "coercion", so named by analogy to type coercion in programming languages. For example, following Herskovits, the *ideal* or core meaning of the English preposition *at* is "for one point to coincide with another". Thus, when an object is referred to as being *at* another object (e.g. "the bicycle is *at* my house"), both objects are conceptualized as points, or "coerced" down to points. This process of coercion corresponds to part of Herskovits' proposed theory of the interpretation of locative expressions. As we shall see below, others have also picked up on this notion, including myself.

[Andre *et al.*, 1986; Schirra *et al.*, 1987] present the Citytour system, produced by the Vitra project at the University of the Saarland in Germany. This system outputs German descriptions of the motions and positions of such objects as cars, churches,

and post offices relative to one another. Two versions are presented: one using a bird's eye view of a city map with idealized point trajectors moving about among the buildings, and another which accepts as input the output of a vision system which detects such objects as cars, streetcars, buildings, and so on, and detects relations among these objects. In both cases the system handles moving trajectors, and is able to recognize situations and events corresponding to a number of German prepositions and verbs. [Andre *et al.*, 1986] describes in some detail algorithms which detect the applicability of particular prepositions. The usages covered include deictically anchored uses such as *hinter dem Rathaus von hier aus* (in English, *behind the town hall from here*).

One shortcoming of the Citytour system is the lack of any notion of degree of applicability of spatial prepositions. Thus, an object is considerd to be either to the left of the town hall, or not; there is no gradation corresponding to the linguistic forms *sort of to the left of*, *directly to the left of*, and so on. As we have seen, the system presented in this thesis does exhibit graded responses of the sort that might be expressed using the above expressions.

Another Vitra system, named Soccer [Schirra, 1990], which describes a soccer match viewed from above, remedies this situation by introducing so-called "probability clouds" relative to a landmark object. These indicate the degree of applicability of a particular preposition for a trajector located at some position relative to the landmark. This is similar in nature to the use of Gaussians in orientation comparison in the system described here. In addition, Schirra adopts the concept of "object coercion", much along the same lines as Hays, when discussing the distance between two objects. This amounts to the observation that since distance is defined between points, rather than objects, we need to coerce the objects into points when calculating the distance betweenthem. And the points that he coerces objects down to for the purposes of distance calculation are the two points where the landmark and trajector are closest. These are precisely the two points used in this thesis in the calculation of the proximal orientation (recall Chapter 5), although Schirra does not proceed to make use of that orientational feature. He also discusses coercion to the center of mass, which is implicitly done in this thesis as well in the case of the center-of-mass orientation. So although there is no learning in the Soccer system, there are similarities between it and the system described here: specifically, the perceptual primitives Schirra focuses on are quite similar to some of the ones used in this thesis.

Jeffrey Siskind

Some of the work of Jeffrey Siskind [Siskind, 1992; Siskind, 1990; Siskind, 1991] focuses on the perception and linguistic categorization of events in a movie of humanoid stick figures moving back and forth, placing balls on tables, and the like. This work is noteworthy for its emphasis on the force dynamic components of lexical semantics in this domain, taking into account such notions as support and attachment. The fact

that a landmark supports a trajector is detected through a process of counterfactual simulation, whereby the system simulates what would happen to a trajector object if the landmark object were not there: if it would fall, then the landmark supports the trajector; otherwise not. To motivate this means of support-detection, Siskind adduces the work of [Freyd *et al.*, 1988], which suggests strongly that humans in fact perform some sort of "unfreezing of potential motion" in their perception of static scenes involving support. (This phenomenon and Freyd's work are discussed in more detail in Chapter 2.) Siskind's work in counterfactual simulation for the detection of semantically significant primitives such as support was the motivation for work much along the same lines, only involving learning, reported in this thesis (see §7.1.3).

David Chapman

[Chapman, 1990] describes a system which plays a video game, guiding the activities of a character in the game as it tries to achieve its goals (e.g. staying alive, killing ghosts, and the like). The system also accepts instructions in English from a human advisor, and interprets the instructions in a manner appropriate to the situation in which the game's central character finds itself at the time. One of Chapman's central tenets is that language use is best studied when grounded in concrete activity, in the pursuit of some set of goals. In line with this outlook, his system brings current plans and other pragmatic concerns into consideration when determining, for instance, the referent of a noun phrase. This emphasis on situated language use requires that the system be able to perceive spatial relations among the objects in the video game world, and this is accomplished through a set of "visual operators", based on Shimon Ullman's work on visual routines [Ullman, 1984]. An example of such an operator is the flood-fill operation which was presented in Chapter 5, and which is used in my system for determining the interiors of obects. Chapman's system uses the same operator for the same purpose, along with several others, such as finding the convex hull and center of mass of an object. (For some earlier work concerning the implementation of some of Ullman's ideas on visual routines, see [Mahoney, 1987].) While his routines are currently implemented procedurally in Lisp, Chapman does suggest connectionist implementations for them, and in fact the bulk of his system is built in a more or less connectionist fashion; he has made neural plausibility a central focus of his work. Given this, it seems reasonable to expect at least preliminary forays on his part into learning, and he has in fact attempted to build a system which will acquire skills for the video game domain, not just use them. This attempt, based on a combination of Sutton's method of temporal differences [Sutton, 1988] and standard back propagation, was not successful.

8.1.2 Related Learning Systems

A number of researchers have addressed the issue of *learning* to associate linguistic forms with perceptally grounded semantics, and thus have focused on problems much like those presented in this thesis. The systems of [Sopena, 1988], [Allen, 1990], and [Bartell and Cottrell, 1991] represent efforts in this direction, accepting much simplified perceptual input, and learning to associate it with natural language descriptions of that input. In addition, somewhat more closely related to the work described in this thesis is the the work of Valeriy Nenov, Paul Munro and coworkers, and Cathy Harris. We review each of these in turn.

Valeriy Nenov

The work of Valeriy Nenov [Nenov and Dyer, 1988; Nenov, 1991] represents the most ambitious effort in this direction to date. His system, *Dete*, accepts three sorts of input: visual (sequences of scenes, involving simple 2-dimensional moving objects, of the sort we have been considering here), verbal (natural language sentences describing the actions seen), and motor (instructions to shift the focus of attention, or to move objects in the world using a "finger"). The output of the system includes verbal descriptions of the visual input, visual imagination, and simple motor performance.

Input processing is done procedurally, the output of the preprocessing being stored in a set of feature maps. From here, Nenov's novel sequential memory mechanism, which he calls Katamic memory, learns to associate streams of input in the various maps. Katamic memory is noteworthy for its rapid learning: Nenov reports learning, with the ability to recognize and recall, within 4 to 6 presentations of a sequence. In addition, Nenov makes a serious attempt to map structures within his memory mechanism to known neuroanatomical structures in the human brain. The system is being tested on learning to associate visual events with sentences from subsets of English, Spanish, and Japanese.

Clearly, *Dete* is much more ambitious in scope than my system, and is capable of a number of functions of which mine is not, such as syntactic processing, language generation and visual imagination. However – not surprisingly given the impressive breadth of his domain – Nenov's treatment of spatial relations is somewhat less detailed. Objects are in all cases "coerced" down to their centers of mass, and a trajector is considered to be *above* a landmark simply if it lies in the half-plane above a horizontal line passing through the landmark's center of mass. In addition, no languages with spatial systems radically different from that of English – such as Mixtec – are handled by *Dete*.

Paul Munro *et al*

[Cosic and Munro, 1988; Munro *et al.*, 1991] present connectionist systems which learn to encode and decode prepositional phrases into and from feature vectors, where the features used are *TR above LM, TR below LM, TR at edge of LM, TR embedded in LM, TR contained by LM, TR within border of LM, TR touching LM, TR near LM, TR far from LM*, and *TR supported by LM*. The general framework within which the work takes place is due to [Herskovits, 1986]. The systems learn to exhibit sensitivity to the nouns used in the prepositional phrase; for example the meaning of *in* in the two phrases *water in glass* and *crack in glass* is different, and the systems learn to encode or decode appropriately, depending on the identity of the trajector and landmark. [Munro *et al.*, 1991] presents the use of networks of this sort in machine translation, such that a prepositional phrase in English is provided as input to a decoding network, and the resulting feature vector is provided as input to a German encoding network, which produces the appropriate German prepositional phrase.

Like the work presented in this thesis, the work of these researchers seeks to provide a model for the phenomenon of polysemy in prepositional use, and their investigation of the effects of different nouns filling slots in prepositional phrases is more detailed than the investigation in §7.1.1 of the interaction of various verbs with prepositions. However, unlike the work presented here, Munro *et al* make do with a feature vector semantic representation, without actually grounding the features in visual input. In addition, Munro *et al* restrict their attention to static spatial prepositions.

Cathy Harris

The work of Cathy Harris [Harris, 1989; Harris, 1991] is somewhat similar in flavor to that of Munro and coworkers, in that it uses connectionist learning systems to focus on the the polysemy of spatial prepositions in different linguistic contexts, without actual perceptual grounding. Her earlier work [Harris, 1989] was a connectionist approach to Claudia Brugman's analysis of the polysemy of the English preposition *over* [Brugman, 1981].

Like Munro *et al*, Harris uses feature vectors, including such perceptually-motivated features as *above, contact, proximity, motion*, and *support*. Unlike Munro, she also includes motion features, and more abstract features such as *authority*, which are needed to capture some of the metaphorical uses of the prepositions she studies. Harris studies the decoding into feature vectors of sentence templates of the format *[trajector] [verb] [preposition] [landmark]* (for example, "bird flew over tree"). Thus, she considers the effects of neighboring verbs on the semantic interpretation of prepositions, not just the identities of the associated trajector and landmark. Given that Harris is not bound, in her choice of features, to those which can be realistically extracted from an image sequence, she has been able to address the issue of polysemy

in considerably more detail than I (or more precisely, my coworkers) have been able to here.

8.2 The Thesis in Retrospect

At the very outset of this dissertation, in Chapter 1, this work was situated within two very general parallel frameworks, the *technical* and the *scientific*. The technical viewpoint focuses primarily on the methods and structures used, in the hope that they may find widespread applicability outside the particular domain which gave rise to them, while the scientific viewpoint highlights the model itself, and the degree to which it can be said to provide an explanation of a real-world linguistic process: the acquisition of visually-grounded semantics.

A central theme of the work taken as a whole has been the interplay between these two frameworks. Specifically, I have attempted to indicate, while presenting the modeling effort itself, and the techniques used in its service, the ways in which each drives the other. In conclusion here, I review the particular technical and scientific contributions made by this work, and in the process once again highlight the ways in which the two frameworks have influenced each other.

There are three central technical contributions made by this work:

- A method is presented for learning in the absence of explicit negative evidence. The basic idea behind the solution is to learn a set of concepts in parallel, and to take each positive example of one concept as *weak* implicit negative evidence for all others. This deliberate weakening of implicit negative evidence permits learning without explicit negatives even when there is significant overlap between concepts.

- The notion of *partially-structured* connectionism is introduced, the idea being to bring together the best of structured and unstructured network design. In the system presented here, this strategy has taken the form of highly structured feature detecting modules, whose structure reflects knowledge of the domain, with full interconnection to unstructured hidden layers above them, so that the features detected may be flexibly combined as required by the training set. In general, the goal has been to simultaneously capture the flexibility afforded by unstructured network design, and the tractability and improved generalization afforded by a structured design.

- The structural devices introduced into the architecture are trained along with the rest of the network in which they are embedded. In general, the activation functions of the units in these structures are chosen so as to be suitable to the domain; error back propagation then serves to adjust a number of parameters

associated with them, so that the search during learning is restricted to a range of options known to be of possible relevance.

These methods provide examples of the means by which a very specific modeling effort can give rise to methods of potentially wide applicability. Each of the techniques grew out of an attempt to model a very specific linguistic phenomenon: the acquisition of semantics under conditions similar to those which face children. Yet they are all very general in nature, and could well be used in any of a number of learning situations.

The central scientific contribution of this dissertation has been the construction and analysis of a neurally-inspired computational model of the acquisition of lexical semantics for spatial terms. The system is able to learn the semantics of spatial terms from a variety of languages, some with spatial systems quite different from that of English, and is able to learn without the benefit of explicit negative evidence, as children appear to. The system is not an implementation of any existing linguistic theory, its primary commitment being simply to match linguistic data. In the process of doing this, it gives rise to the following two falsifiable linguistic predictions:

- In any language, any semantically significant static feature which appears in mid-event for some spatial term in the language will appear at end-event for at least one spatial term in that language. For example, the fact that the static feature of inclusion appears in mid-event for English *through* leads us to predict that there will exist a term in English denoting an event ending in inclusion of the trajector in the landmark (such as *in* or *into*).

- Some languages use the same word to denote static location in some configuration and motion into that configuration. For example, English *in* can be used to mean either static inclusion, or motion into a state of inclusion, synonymously with *into*. The model predicts that this usage will be more likely to appear than the use of the same word to denote either static location in some configuration or motion *out of* that configuration.

All of this is, of course, critically dependent on the actual architectural structures and techniques used. This illustrates the means by which the general connectionist tools which were used can shape and give rise to particular scientific results.

In the course of this dissertation, then, I hope to have demonstrated that a fairly detailed computational modeling effort in a very specific and perhaps somewhat obscure domain of linguistic inquiry can give rise to architectural principles and training methods of general applicability, principles and methods which are of potential utility to connectionist modelers working in domains that have nothing whatsoever to do with the acquisition of spatial semantics, and perhaps to researchers in other paradigms as well. I also hope to have demonstrated that a modeling effort of this sort, employing connectionist architectural and training techniques, can produce a neurally-plausible explanatory model of a particular linguistic process, and through

it, can give rise to falsifiable predictions concerning the nature of the linguistic categorization of space.

Bibliography

[Ackley *et al.*, 1987] David Ackley, Geoffrey Hinton, and Terrence Sejnowski, "A Learning Algorithm for Boltzmann Machines," In Martin A. Fischler and Oscar Firschein, editors, *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*. Morgan Kaufmann, 1987.

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

[Ahmad, 1991] Subutai Ahmad, "Efficient Visual Search: A Connectionist Solution," In *Proceedings of the 13th Annual Conference of the Cognitive Science Society*, 1991.

[Ahmad, 1992] Subutai Ahmad, (personal communication), 1992.

[Ahmad and Omohundro, 1990] Subutai Ahmad and Stephen Omohundro, "A Connectionist System for Extracting the Locations of Point Clusters," Technical Report TR-90-011, International Computer Science Institute, Berkeley, California, 1990.

[Allen, 1990] Robert B. Allen, "Connectionist Language Users," *Connection Science*, 2(4):279–311, 1990.

[Andre *et al.*, 1986] E. Andre, G. Bosch, G. Herzog, and T. Rist, "Characterizing Trajectories of Moving Objects Using Natural Language Path Descriptions," Technical Report 5, Fachbereich 14 - Informatik, Universität des Saarlandes, 1986.

[Badler, 1975] N. Badler, "Temporal Scene Analysis: Conceptual Descriptions of Object Movements," Technical Report TR-80, Dept. of Computer Science, University of Toronto, 1975.

[Bailey, 1992] David Bailey, "Key Events in Spatial Semantics Acquisition," Unpublished paper from course on connectionist modeling of spatial semantics, University of California at Berkeley, May 1992.

[Bajcsy *et al.*, 1985] Ruzena Bajcsy, Aravind Joshi, Eric Krotkov, and Amy Zwarico, "LandScan: A Natural Language and Computer Vision System for Analyzing Aerial Images," In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, 1985, pp. 919-921.

[Baker, 1992] Collin Baker, "Internalized Gravity, or How to Learn Spatial Concepts from Movies," Unpublished paper from course on connectionist modeling of spatial semantics, University of California at Berkeley, May 1992.

[Ballard and Brown, 1982] Dana Ballard and Christopher Brown, *Computer Vision*, Prentice-Hall, 1982.

[Ballard, 1987a] Dana H. Ballard, "Cortical Connections and Parallel Processing: Structure and Function," In Michael A. Arbib and Allen R. Hanson, editors, *Vision, Brain, and Cooperative Computation*. MIT Press, 1987.

[Ballard, 1987b] Dana H. Ballard, "Parameter Nets," In Martin A. Fischler and Oscar Firschein, editors, *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*. Morgan Kaufmann, 1987.

[Bartell and Cottrell, 1991] Brian Bartell and Garrison Cottrell, "A Model of Symbol Grounding in a Temporal Environment," In *The AAAI Spring Symposium Workshop on Connectionist Natural Language Processing*, March 1991.

[Behrend, 1989] Douglas Behrend, "Default Values in Verb Frames: Cognitive Biases for Learning Verb Meanings," In *Proceedings of the 11th Annual Meeting of the Cognitive Science Society*, pages 252–258, 1989.

[Berlin and Kay, 1969] Brent Berlin and Paul Kay, "Basic Color Terms: Their Universality and Evolution," University of California Press, Berkeley, 1969.

[Beutter, 1992] Brent Beutter, (personal communication), 1992.

[Bolinger, 1965] Dwight Bolinger, "The Atomization of Meaning," *Language*, 41:555–573, 1965.

[Bowerman, 1983] Melissa Bowerman, "How Do Children Avoid Constructing an Overly General Grammar in the Absence of Feedback about What is Not a Sentence?," In *Papers and Reports on Child Language Development*. Stanford University, 1983.

[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.

[Braine, 1971] M. Braine, "On Two Types of Models of the Internalization of Grammars," In D. Slobin, editor, *The Ontogenesis of Grammar*. Academic Press, 1971.

[Bridle, 1990] John Bridle, "Training Stochastic Model Recognition Algorithms as Networks Can Lead to Maximum Mutual Information Estimation of Parameters," *Advances in Neural Information Processing Systems*, 2:211–217, 1990.

[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.

[Casad, 1982] Eugene Casad, "Cora Locationals and Structured Imagery," Ph.D. Dissertation, University of California, San Diego, 1982.

[Chapman, 1990] David Chapman, *Vision, Instruction, and Action*, PhD thesis, Massachusetts Institute of Technology, 1990.

[Chomsky and Lasnik, 1977] N. Chomsky and H. Lasnik, "Filters and Control," *Linguistic Inquiry*, 8:425–504, 1977.

[Chou and Raman, 1987] Paul Chou and Rajeev Raman, "On Relaxation Algorithms Based on Markov Random Fields," Technical Report 212, Computer Science Department, University of Rochester, 1987.

[Clark, 1987] Eve Clark, "The Principle of Contrast: A Constraint on Language Acquisition," In B. MacWhinney, editor, *Mechanisms of Language Acquisition*, Hillsdale, NJ, 1987. Lawrence Erlbaum.

[Comrie, 1981] Bernard Comrie, *Language Universals and Linguistic Typology*, University of Chicago Press, Chicago, 1981.

[Cooper, 1989] Paul Cooper, "Parallel Object Recognition from Structure," Technical Report 301, PhD. thesis, Department of Computer Science, University of Rochester, July 1989.

[Cosic and Munro, 1988] Cynthia Cosic and Paul Munro, "Learning to Represent and Understand Locative Prepositional Phrases," In *Proceedings of the 10th Annual Meeting of the Cognitive Science Society*, 1988, pp. 257-262.

[Cottrell, 1985] Garrison Cottrell, *A Connectionist Approach to Word Sense Disambiguation*, PhD thesis, University of Rochester, 1985.

[Cross and Jain, 1983] G. R. Cross and A. K. Jain, "Markov Random Field Texture Models," *IEEE PAMI*, 5(1):25–39, January 1983.

[Cybenko, 1989] G. Cybenko, "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989.

[Denker *et al.*, 1987] J. Denker, D. Schwartz, B. Wittner, S. Solla, R. Howard, L. Jackel, and J. Hopfield, "Large Automatic Learning, Rule Extraction and Generalization," *Complex Systems*, 1:877–922, 1987.

[Denny, 1980] J. Peter Denny, "Semantics of the Inuktitut (Eskimo) Spatial Deictics," Draft, submitted to *International Journal of American Linguistics*, February 1980.

[Denny and Issaluk, 1976] Peter Denny and Luke Issaluk, "Semantically Organized Tables of Inuktitut Locatives," Technical Report 352, Department of Psychology, University of Western Ontario, March 1976.

[DeValois and DeValois, 1990] Russell DeValois and Karen DeValois, *Spatial Vision*, Oxford University Press, 1990.

[Elman, 1988] J. L. Elman, "Finding Structure in Time," Technical Report 8801, Center for Research in Language, University of California, San Diego, 1988.

[Fahlman, 1988] Scott Fahlman, "An Empirical Study of Learning Speed in Back-Propagation Networks," Technical Report CMU-CS-88-162, Department of Computer Science, Carnegie Mellon University, June 1988.

[Fahlman, 1991] Scott E. Fahlman, "The Recurrent Cascade-Correlation Architecture," Technical Report CMU-CS-91-100, Dept. of Computer Science, Carnegie Mellon University, May 1991.

[Fanty, 1988] Mark Fanty, "Learning in Structured Connectionist Networks," Technical Report 252, Department of Computer Science, University of Rochester, April 1988.

[Feldman and Ballard, 1982] J. Feldman and D. Ballard, "Connectionist Models and Their Properties," *Cognitive Science*, 6:205–254, 1982.

[Feldman *et al.*, 1988] J. Feldman, M. Fanty, and N. Goddard, "Computing with Structured Neural Networks," *IEEE Computer*, 21(3):91–104, 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.

[Feldman, 1986] Jerome Feldman, "Neural Representation of Conceptual Knowledge," Technical Report 189, Department of Computer Science, University of Rochester, 1986.

[Fodor, 1983] Jerry A. Fodor, *The Modularity of Mind*, MIT Press, Cambridge, MA, 1983.

[Freyd *et al.*, 1988] Jennifer Freyd, Teresa Pantzer, and Jeannette Cheng, "Representing Statics as Forces in Equilibrium," *Journal of Experimental Psychology: General*, 117(4):395–407, 1988.

[Geman and Geman, 1984] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE PAMI*, 6(6):721–741, November 1984.

[Goldberg, 1991] Adele Goldberg, (personal communication), 1991.

[Greenberg, 1963] Joseph H. Greenberg, *Universals of Language*, MIT Press, Cambridge, MA, 1963.

[Guyon *et al.*, 1991] I. Guyon, P. Albrecht, Y. LeCun, J. Denker, and W. Hubbard, "Design of a Neural Network Character Recognizer for a Touch Terminal," *Pattern Recognition*, 24(2):105–119, 1991.

[Hanania, 1990] Edith Hanania, (personal communication), 1990.

[Harris, 1989] Cathy Harris, "A Connectionist Approach to the Story of 'Over'," In *Berkeley Linguistics Society*, volume 15, pages 126–138, 1989.

[Harris, 1991] Cathy Harris, "Back Propagation Representations for the Rule-Analogy Continuum: Pros and Cons," (Draft), 1991.

[Hasegawa, 1991] Yoko Hasegawa, (personal communication), 1991.

[Hays, 1987] Ellen M. Hays, "A Computational Treatment of Locative Relations in Natural Language," Technical Report MS-CIS-87-31, Department of Computer and Information Science, University of Philadelphia, 1987.

[Hebb, 1949] D. Hebb, *The Organization of Behavior*, Wiley, 1949.

[Herskovits, 1986] Annette Herskovits, *Language and Spatial Cognition: An Interdisciplinary Study of the Prepositions in English*, Cambridge University Press, 1986.

[Hertz *et al.*, 1991] John Hertz, Anders Krogh, and Richard G. Palmer, *Introduction to the Theory of Neural Computation*, Addison Wesley, 1991.

[Hill, 1977] C. A. Hill, "Linguistic Representation of Spatial and Temporal Orientation," In *Proceedings of the Fourth Annual Meeting of the Berkeley Linguistics Society*. University of California, Berkeley, 1977.

[Hinton, 1986] G. Hinton, "Learning Distributed Representations of Concepts," In *Proceedings of the 8th Annual Cognitive Science Society Conference*. Lawrence Erlbaum, 1986.

[Hinton, 1990] Geoffrey Hinton, "Connectionist Learning Procedures," In J. G. Carbonell, editor, *Machine Learning: Paradigms and Methods*. MIT Press, 1990.

[Hopfield, 1982] J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proceedings of the National Academy of Sciences of the USA*, 79:2554–2558, 1982.

[Hopfield, 1984] J. J. Hopfield, "Neurons with Graded Responses Have Collective Computational Properties Like Those of Two-State Neurons," *Proceedings of the National Academy of Sciences of the USA*, 81:3088–3092, 1984.

[Horn, 1986] Berthold Klaus Paul Horn, *Robot Vision*, MIT Press, 1986.

[Hubel and Wiesel, 1959] D. Hubel and T. Wiesel, "Receptive Fields of Single Neurones in the Cat's Visual Cortex," *Journal of Physiology*, 148:574–591, 1959.

[Hubel and Wiesel, 1962] D. Hubel and T. Wiesel, "Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex," *Journal of Physiology*, 160:106–154, 1962.

[Hummel and Biederman, 1990] J. Hummel and I. Biederman, "Dynamic Binding in a Neural Network for Shape Recognition," Technical Report 90-5, Dept. of Psychology, University of Minnesota, 1990.

[Jackendoff, 1983] Ray Jackendoff, *Semantics and Cognition*, MIT Press, Cambridge, MA, 1983.

[Jackendoff and Landau, 1991] Ray Jackendoff and Barbara Landau, "Spatial Language and Spatial Cognition," In Donna Jo Napoli and Judy Anne Kegl, editors, *Bridges Between Psychology and Linguistics: A Swarthmore Festschrift for Lila Gleitman*. L. Erlbaum Associates, Hillsdale, NJ, 1991.

[Jacobs *et al.*, 1990] Robert Jacobs, Michael Jordan, and Andrew Barto, "Task Decomposition Through Competition in a Modular Connectionist Architecture: The What and Where Vision Tasks," Technical Report COINS 90-27, Dept. of Computer and Information Science, University of Massachusetts at Amherst, 1990.

[Jain *et al.*, 1992] Ajay Jain, Alex Waibel, and David Touretzky, "PARSEC: A Structured Connectionist Parsing System for Spoken Language," In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages I–205 – I–208, 1992.

[Janda, 1984] Laura Janda, "A Semantic Analysis of the Russian Verbal Prefixes *za-*, *pere-*, *do-*, and *ot-*.," Ph.D. Dissertation, University of California, Los Angeles, 1984.

[Jaramillo and Schiff, 1992] Narciso Jaramillo and Michael Schiff, "An Alternate Connectionist Architecture for Learning Motion-Based Spatial Semantics," Unpublished paper from course on connectionist modeling of spatial semantics, University of California at Berkeley, May 1992.

[Johnston and Slobin, 1979] Judith Johnston and Dan Slobin, "The Development of Locative Expressions in English, Italian, Serbo-Croatian and Turkish," *Journal of Child Language*, 6:529–545, 1979.

[Jordan, 1986] M. I. Jordan, "Attractor Dynamics and Parallelism in a Connectionist Sequential Machine," In *Proceedings of the 8th Annual Conference of the Cognitive Science Society*, pages 531–546, 1986.

[Kanizsa, 1979] G. Kanizsa, *Organization in Vision: Essays on Gestalt Perception*, Praeger, New York, 1979.

[Keeler *et al.*, 1991] James Keeler, David Rumelhart, and Wee-Kheng Leow, "Integrated Segmentation and Recognition of Hand-Printed Numerals," Technical Report ACT-NN-010-91, Microelectronics and Computer Technology Corporation, January 1991.

[Koenderink, 1990] Jan J. Koenderink, *Solid Shape*, MIT Press, 1990.

[Kolers, 1972] Paul A. Kolers, *Aspects of Motion Perception*, Pergamon Press, New York, 1972.

[Kruglyak, 1990] Leonid Kruglyak, "How to Solve the N Bit Encoder Problem with Just Two Hidden Units," *Neural Computation*, 2:399–401, 1990.

[Kuffler, 1953] S. Kuffler, "Discharge Patterns and Functional Organization of Mammalian Retina," *Journal of Neurophysiology*, 16:37–68, 1953.

[Lakoff, 1987] George Lakoff, *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*, University of Chicago Press, 1987.

[Langacker, 1987] Ronald Langacker, *Foundations of Cognitive Grammar I: Theoretical Prerequisites*, Stanford University Press, Stanford, 1987.

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

[Legendre *et al.*, 1990a] Géraldine Legendre, Yoshiro Miyata, and Paul Smolensky, "Harmonic Grammar — A formal multi-level connectionist theory of linguistic well-formedness: An application," Technical Report #90-4, Institute of Cognitive Science, University of Colorado, Boulder, Colo., 1990.

[Legendre *et al.*, 1990b] Géraldine Legendre, Yoshiro Miyata, and Paul Smolensky, "Harmonic Grammar — A formal multi-level connectionist theory of linguistic well-formedness: Theoretical foundations," Technical Report #90-5, Institute of Cognitive Science, University of Colorado, Boulder, Colo., 1990.

[Lehky and Sejnowski, 1988] Sidney Lehky and Terrence Sejnowski, "Network Model of Shape-from-Shading: Neural Function Arises from Both Receptive and Projective Fields," *Nature*, 333:452–454, 1988.

[Lindner, 1982] Susan Lindner, "What Goes Up Doesn't Necessarily Come Down: The Ins and Outs of Opposites," In *Papers from the Eighteenth Regional Meeting, Chicago Linguistic Society*, pages 305–323, 1982.

[MacWhinney, 1989] Brian MacWhinney, "Competition and Lexical Categorization," In *Linguistic Categorization*, number 61 in Current Issues in Linguistic Theory. John Benjamins Publishing Co., Amsterdam and Philadelphia, 1989.

[Mahoney, 1987] James Mahoney, "Image Chunking: Defining Spatial Building Blocks for Scene Analysis," Technical Report 980, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1987.

[Markman, 1987] Ellen M. Markman, "How Children Constrain the Possible Meanings of Words," In *Concepts and conceptual development: Ecological and intellectual factors in categorization.* Cambridge University Press, 1987.

[Marr, 1982] David Marr, *Vision*, Freeman, New York, 1982.

[Martin, 1987] James Martin, "The Acquisition of Polysemy," In *Proceedings of the 4th International Workshop on Machine Learning*, 1987.

[Maskara and Noetzel, 1992] Arun Maskara and Andrew Noetzel, "Forced Simple Recurrent Neural Networks and Grammatical Inference," In *Proceedings of the 14th Annual Meeting of the Cognitive Science Society*, 1992, (to appear).

[Matsumoto, 1989] Yo Matsumoto, "Abstract Change and Japanese Resultative Sentences," Unpublished paper, from class on cognitive semantics, UC Berkeley, 1989.

[McCulloch and Pitts, 1943] W. S. McCulloch and W. Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity," *Bulletin of Mathematical Biophysics*, 5:115–137, 1943.

[Merzenich and Kaas, 1982] M. Merzenich and J. Kaas, "Reorganization of Mammalian Somatosensory Cortex Following Peripheral Nerve Injury," *Trends in Neuroscience*, 5:434–436, 1982.

[Métin and Frost, 1989] Christine Métin and Douglas Frost, "Visual Responses of Neurons in Somatosensory Cortex of Hamsters with Experimentally Induced Retinal Projections to Somatosensory Thalamus," *Proceedings of the National Academy of Sciences of the USA, Neurobiology*, 86:357–361, 1989.

[Miikkaulainen, 1990] Risto Miikkaulainen, *DISCERN: A Distributed Artificial Neural Network Model of Script Processing and Memory*, PhD thesis, University of California, Los Angeles, 1990.

[Miikkaulainen, 1991] Risto Miikkaulainen, "Parsing Embedded Clauses with Simple Recurrent Networks," In *The AAAI Spring Symposium Workshop on Connectionist Natural Language Processing*, pages 211–215, March 1991.

[Miller and Johnson-Laird, 1976] G. Miller and P. Johnson-Laird, *Language and Perception*, Belknap Press, 1976.

[Minsky and Papert, 1969] Marvin Minsky and Seymour Papert, *Perceptrons*, MIT Press, 1969.

[Mishkin, 1972] M. Mishkin, "Cortical Visual Areas and their Interactions," In A. G. Karczmar and J. C. Eccles, editors, *Brain and Human Behavior*. Springer, 1972.

[Mitchell, 1980] Tom M. Mitchell, "The Need for Biases in Learning Generalizations," Technical Report CBM-TR-117, Computer Science Department, Rutgers University, May 1980.

[Morgan and Bourlard, 1989] N. Morgan and H. Bourlard, "Generalization and Parameter Estimation in Feedforward Nets: Some Experiments," Technical Report TR-89-017, International Computer Science Institute, Berkeley, California, April 1989.

[Mozer, 1988] Michael Mozer, "A Focused Back-Propagation Algorithm for Temporal Pattern Recognition," Technical Report CRG-TR-88-3, Connectionist Research Group, University of Toronto, 1988.

[Mozer *et al.*, 1991] Michael Mozer, Richard Zemel, and Marlene Behrmann, "Learning to Segment Images Using Dynamic Feature Binding," Technical Report CU-CS-540-91, Dept. of Computer Science, University of Colorado at Boulder, August 1991.

[Munro *et al.*, 1991] Paul Munro, Cynthia Cosic, and Mary Tabasko, "A Network for Encoding, Decoding and Translating Locative Prepositions," *Connection Science*, 3(3):225–240, 1991.

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

[Nenov, 1991] Valeriy Nenov, *Perceptually Grounded Language Acquisition: A Neural/Procedural Hybrid Model*, PhD thesis, University of California at Los Angeles, 1991.

[Nenov and Dyer, 1988] Valeriy Nenov and Michael Dyer, "DETE: Connectionist/Symbolic Model of Visual and Verbal Association," Technical Report UCLA-AI-88-6, University of California, Los Angeles, 1988.

[Nowlan, 1990] Steven J. Nowlan, "Competing Experts: An Experimental Investigation of Associative Mixture Models," Technical Report CRG-TR-90-5, Connectionist Research Group, University of Toronto, September 1990.

[Olson, 1989] Thomas Olson, "An Architectural Model of Visual Motion Understanding," Technical Report 305, Department of Computer Science, University of Rochester, August 1989.

[O'Reilly *et al.*, 1990] R. O'Reilly, S. Kosslyn, C. Marsolek, and C. Chabris, "Receptive Field Characteristics That Allow Parietal Lobe Neurons to Encode Spatial Properties of Visual Input: A Computational Analysis," *Journal of Cognitive Neuroscience*, 2:141–155, 1990.

[Osterholtz *et al.*, 1992] Louise Osterholtz, Charles Augustine, Arthur McNair, Ivica Rogina, Hiroaki Saito, Tilo Sloboda, Joe Tebelskis, and Alex Waibel, "Testing Generality in JANUS: A Multi-lingual Speech Translation System," In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages I–209 – I–212, 1992.

[Patarnello and Carnevali, 1987] S. Patarnello and P. Carnevali, "Learning Networks of Neurons with Boolean Logic," *Europhysics Letters*, 4(4):503–508, 1987.

[Pearlmutter, 1990] Barak A. Pearlmutter, "Dynamic Recurrent Neural Networks," Technical Report CMU-CS-90-196, Dept. of Computer Science, Carnegie Mellon University, December 1990.

[Pinker, 1989] Steven Pinker, *Learnability and Cognition: The Acquisition of Argument Structure*, MIT Press, 1989.

[Poggio and Edelman, 1990] T. Poggio and S. Edelman, "A Network That Learns to Recognize Three-Dimensional Objects," *Nature*, 343:263–266, 1990.

[Pollack, 1990a] Jordan Pollack, "Language Acquisition via Strange Automata," In *Proceedings of the 12th Annual Conference of the Cognitive Science Society*, pages 678–685, 1990.

[Pollack, 1988] Jordan B. Pollack, "Recursive Auto-Associative Memory: Devising Compositional Distributed Representations," In *Proceedings of the 10th Annual Meeting of the Cognitive Science Society*, pages 33–39, 1988.

[Pollack, 1990b] Jordan B. Pollack, "Recursive Distributed Representations," *Artificial Intelligence*, 46:77–105, 1990.

[Regier, 1990] Terry Regier, "Learning Spatial Terms Without Explicit Negative Evidence," Technical Report TR-90-057, 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, pp. 191-196.

[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, pp. 138-145.

[Regier, 1991c] Terry Regier, "Learning Spatial Concepts Using a Partially-Structured Connectionist Architecture," Technical Report TR-91-050, International Computer Science Institute, Berkeley, California, October 1991.

[Regier, 1991d] Terry Regier, "Line Labeling and Junction Labeling: A Coupled System for Image Interpretation," In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, 1991, pp. 1305-1310.

[Regier, 1991e] Terry Regier, "Line Labeling Using Markov Random Fields," Technical Report TR-91-059, International Computer Science Institute, Berkeley, CA, November 1991.

[Renals *et al.*, 1991] Steve Renals, Nelson Morgan, and Herve Bourlard, "Probability Estimation by Feed-forward Networks in Continuous Speech Recognition," Technical Report TR-91-030, International Computer Science Institute, Berkeley, California, August 1991.

[Rosch, 1973] Eleanor Rosch, "On the Internal Structure of Perceptual and Semantic Categories," In Timothy E. Moore, editor, *Cognitive Development and the Acquisition of Language*, pages 111–144, New York, 1973. Academic Press.

[Rosch, 1977] Eleanor Rosch, "Human Categorization," In Neil Warren, editor, *Studies in Cross-cultural Psychology*, volume 1, pages 1–49, London, 1977. Academic Press.

[Rosch, 1978] Eleanor Rosch, "Principles of Categorization," In Eleanor Rosch and Barabara B. Lloyd, editors, *Cognition and Categorization*, pages 27–47, Hillsdale, NJ, 1978. Erlbaum.

[Rosenblatt, 1962] F. Rosenblatt, *Principles of Neurodynamics*, Spartan, 1962.

[Rumelhart *et al.*, 1986] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, "Learning Internal Representations by Error Propagation," In James L. McClelland and David E. Rumelhart, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, pages 318–362. MIT Press, 1986.

[Schirra *et al.*, 1987] J. Schirra, G. Bosch, C. Sung, and G Zimmermann, "From Image Sequences to Natural Language: A First Step towards Automatic Perception and Description of Motions," Technical Report 26, Fachbereich 14 - Informatik, Universität des Saarlandes, 1987.

[Schirra, 1990] Jörg Schirra, "A Contribution to Reference Semantics of Spatial Prepositions: The Visualization Problem and its Solution in VITRA," Technical Report 75, Fachbereich 14 - Informatik, Universität des Saarlandes, 1990.

[Schmidhuber, 1991] Jürgen Schmidhuber, "Neural Sequence Chunkers," Technical Report FKI-148-91, Institut für Informatik, Technische Universität München, 1991.

[Sejnowski and Hinton, 1987] Terrence J. Sejnowski and Geoffrey H. Hinton, "Separating Figure from Ground with a Boltzmann Machine," In Michael A. Arbib and Allen R. Hanson, editors, *Vision, Brain, and Cooperative Computation.* MIT Press, 1987.

[Shastri, 1988] Lokendra Shastri, *Semantic Nets: Evidential Formalization and its Connectionist Implementation*, Morgan Kaufmann, 1988.

[Shastri and Ajjanagadde, 1990] Lokendra Shastri and Venkat Ajjanagadde, "An Optimally Efficient Limited Inference System," In *Proceedings of the American Association for Artificial Intelligence*, 1990.

[Siegelman and Sontag, 1991] Hava Siegelman and Eduardo Sontag, "Neural Nets are Universal Computing Devices," Technical Report SYCON-91-08, Rutgers University Center for Systems and Control, May 1991.

[Siskind, 1992] Jeffrey Siskind, *Naive Physics, Event Perception, Lexical Semantics and Language Acquisition*, PhD thesis, Massachusetts Institute of Technology, 1992.

[Siskind, 1990] Jeffrey Mark Siskind, "Acquiring Core Meanings of Words, Represented as Jackendoff-Style Conceptual Structures, from Correlated Streams of Linguistic and Non-Linguistic Input," In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 143–156, June 1990.

[Siskind, 1991] Jeffrey Mark Siskind, "Naive Physics, Event Perception, Lexical Semantics and Language Acquisition," In *The AAAI Spring Symposium Workshop on Machine Learning of Natural Language and Ontology*, March 1991.

[Sopena, 1988] Josep Maria Sopena, "Verbal Description of Visual Blocks World Using Neural Networks," Technical Report 10, Departament de Psicologica Basica, Universitat de Barcelona, December 1988.

[Stolcke, 1990a] Andreas Stolcke, (personal communication), 1990.

[Stolcke, 1990b] Andreas Stolcke, "Learning Feature-Based Semantics with Simple Recurrent Networks," Technical Report TR-90-015, International Computer Science Institute, Berkeley, CA, April 1990.

[Sutton, 1988] Richard Sutton, "Learning to Predict by the Methods of Temporal Differences," *Machine Learning*, 3:9–44, 1988.

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

[Talmy, 1987] Leonard Talmy, "Force Dynamics in Language and Cognition," Technical Report 49, Institute of Cognitive Studies, University of California at Berkeley, November 1987.

[Talmy, 1990] Leonard Talmy, (personal communication), 1990.

[Taub and Jacobs, 1992] Sarah Taub and Adam Jacobs, "Report on Deixis," Unpublished paper from course on connectionist modeling of spatial semantics, University of California at Berkeley, May 1992.

[Taube *et al.*, 1987] A. M. Taube, I. W. Litvinova, A. D. Miller, and R. C. Daglish, *Russian-English Dictionary*, Russky Yazyk Publishers, Moscow, 1987.

[Thibadeau, 1986] Robert Thibadeau, "Artificial Perception of Actions," *Cognitive Science*, 10(2):117–150, 1986.

[Tomasello, 1987] Michael Tomasello, "Learning to Use Prepositions: A Case Study," *Journal of Child Language*, 14:79–98, 1987.

[Tootell *et al.*, 1982] R. Tootell, M. Silverman, E. Switkes, and R. DeValois, "Deoxyglucose Analysis of Retinotopic Organization in Primate Striate Cortex," *Science*, 218:902–904, 1982.

[Tsotsos, 1981] John Tsotsos, "Temporal Event Recognition: An Application to Left Ventricular Performance," In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 1981, pp. 900-907.

[Ullman, 1984] Shimon Ullman, "Visual Routines," *Cognition*, 18:97–159, 1984.

[Ungerleider and Mishkin, 1982] L. G. Ungerleider and M. Mishkin, "Two Cortical Visual Systems," In D. G. Ingle, M. A. Goodale, and R. J. W. Mansfield, editors, *Analysis of Visual Behavior*. MIT Press, 1982.

[Vandeloise, 1991] Claude Vandeloise, *Spatial Prepositions: A Case Study from French*, University of Chicago Press, Chicago, 1991.

[von der Heydt *et al.*, 1984] R. von der Heydt, E. Peterhans, and G. Baumgartner, "Illusory Contours and Cortical Neuron Responses," *Science*, 224:1260–1262, 1984.

[Waibel *et al.*, 1987] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme Recognition Using Time-Delay Neural Networks," Technical Report TR-1-0006, ATR Interpreting Telephony Research Laboratories, Japan, 1987.

[Waibel, 1989] Alex Waibel, "Modular Construction of Time-Delay Neural Networks for Speech Recognition," *Neural Computation*, 1:39–46, 1989.

[Waibel *et al.*, 1991] Alex Waibel, Ajay Jain, Arthur McNair, Hiroaki Saito, Alexander Hauptmann, and Joe Tebelskis, "JANUS: A Speech-to-Speech Translation System Using Connectionist and Symbolic Processing Strategies," In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 793–796, 1991.

[Waltz, 1981] David Waltz, "Toward a Detailed Model of Processing for Language Describing the Physical World," In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 1981, pp. 1-6.

[Waltz and Pollack, 1985] David Waltz and Jordan Pollack, "Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation," *Cognitive Science*, 9(1):51–74, 1985.

[Weber, 1989a] Susan Hollbach Weber, "Figurative Adjective-Noun Interpretation in a Structured Connectionist Network," In *Proceedings of the 11th Annual Meeting of the Cognitive Science Society*, 1989, pp. 204-211.

[Weber, 1989b] Susan Hollbach Weber, "A Structured Connectionist Approach to Direct Inferences and Figurative Adjective-Noun Combinations," Technical Report 289, Department of Computer Science, University of Rochester, 1989, PhD thesis.

[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.

[Weigend *et al.*, 1990] Andreas Weigend, Bernardo Huberman, and David Rumelhart, "Predicting the Future: A Connectionist Approach," Technical Report Stanford-PDP-90-01, PDP Research Group, Stanford University, April 1990.

[Werbos, 1974] P. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, PhD thesis, Harvard University, 1974.

[Whorf, 1956] Benjamin Lee Whorf, *Language, Thought, and Reality*, MIT Press, Cambridge, MA, 1956, (ed. John B. Carroll).

[Williams and Zipser, 1989] Ronald Williams and David Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," *Neural Computation*, 1:270–280, 1989.

[Winston, 1977] P. H. Winston, "Learning Simple Descriptions," In *Artificial Intelligence*, chapter 2, pages 29–44. Addison-Wesley, Reading, Mass., 1977.

[Yuhan, 1991] Albert Hanyong Yuhan, "Dynamic Computation of Spatial Reference Frames in Narrative Understanding," Technical Report 91-03, Department of Computer Science, State University of New York at Buffalo, 1991.

[Zemel *et al.*, 1992] Richard S. Zemel, Christopher K. I. Williams, and Michael C. Mozer, "Adaptive Networks of Directional Units," Technical Report CRG-TR-92-2, Connectionist Research Group, University of Toronto, June 1992.

[Zlatev, 1992] Jordan Zlatev, "A Study of Perceptually Grounded Polysemy in a Spatial Microdomain," Technical Report TR-92-048, International Computer Science Institute, Berkeley, California, August 1992.