

## ABSTRACT

Title of Dissertation:        INVESTIGATIONS INTO THE NEURAL BASIS  
   OF STRUCTURED REPRESENTATIONS

Carol Susan Whitney, Doctor of Philosophy, 2004

Dissertation directed by: Professor Amy Weinberg  
   Departments of Computer Science and Linguistics

The problem of how the brain encodes structural representations is investigated via the formulation of computational theories constrained from the bottom-up by neurobiological factors, and from the top-down by behavioral data. This approach is used to construct models of letter-position encoding in visual word recognition, and of hierarchical representations in sentence parsing.

The problem of letter-position encoding entails the specification of how the retinotopic representation of a stimulus (a printed word) is progressively converted into an abstract representation of letter order. Consideration of the architecture of the visual system, letter perceptibility studies, and form-priming experiments led to the SERIOL model, which is comprised of five layers: (1) a (retinotopic) edge layer, in which letter activations are determined by the acuity gradient; (2) a (retinotopic) feature layer, in which letter activations conform to a monotonically decreasing activation gradient, dubbed the locational gradient;

(3) an abstract letter layer, in which letter order is encoded sequentially. (4) a bigram layer, in which contextual units encode letter pairs that fire in a particular order; (5) a word layer.

Because the acuity and locational gradients are congruent to each other in one hemisphere but not the other, formation of the locational gradient requires hemisphere-specific processing. It is proposed that this processing underlies visual-field asymmetries associated with word length and orthographic-neighborhood size. Hemifield lexical-decision experiments in which contrast manipulations were used to modify activation patterns confirmed this account.

In contrast to the linear relationships between letters, a parse of a sentence requires hierarchical representations. Consideration of a fixed-connectivity constraint, brain imaging studies, sentence-complexity phenomena, and insights from the SERIOL model led to the TPARRSE model, in which hierarchical relationships are represented by a predefined distributed encoding. This encoding is constructed with the support of working memory, which encodes relationships between phrases via two synchronized sequential representations. The model explains complexity phenomena based on specific proposals as to how information is represented and manipulated in syntactic working memory. In contrast to capacity-based metrics, the TPARRSE model provides a more comprehensive account of these phenomena.

INVESTIGATIONS INTO THE NEURAL BASIS  
OF STRUCTURED REPRESENTATIONS

by

Carol Susan Whitney

Dissertation submitted to the Faculty of the Graduate School of the  
University of Maryland, College Park in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
2004

Advisory Committee:

Professor Amy Weinberg, Chairman/Advisor  
Professor Christopher Cherniak  
Professor Don Perlis  
Professor Colin Phillips  
Professor James Reggia

© Copyright by  
Carol Susan Whitney  
2004

## DEDICATION

To my husband.

## ACKNOWLEDGEMENTS

First, I would of course like to thank my committee members. Amy Weinberg, my advisor, was a perfect fit in terms of computational, cognitive, and psycholinguistic interests. Her down-to-earth attitude and sense of humor have always been a pleasure. Colin Phillips has provided encouragement and detailed comments on my research. Amy and he have both helped me to present my research, identify important issues and implicit assumptions, and to position my work in the big picture. As these matters are not my strong point, hopefully their valuable instruction has rubbed off onto me somewhat. Jim Reggia was my M.S. adviser and has stuck with me on my Phd. committee. David Poeppel served as a committee member through my Phd studies, but was not able to attend my defense. Don Perlis agreed to fill in, and Christopher Cherniak took on the role of Dean's Representative. Corey Washington initially sponsored my application to the Neural and Cognitive Sciences program, and was my adviser for the

early part of my graduate work. My thanks to all these professors for contributing their time and expertise to my Phd studies.

I am also grateful to my European colleagues, for their interest in my work in visual word recognition. Michal Lavidor re-ignited my interest in that work by inviting me to participate in a symposium. Thanks to her willingness to investigate my crazy ideas, I have been able to obtain experimental results to support my computational model. Michal and Tatjana Nazir also arranged a workshop which provided valuable interaction with the organizers themselves, and other researchers such as Marc Brysbaert, Andrew Ellis, Vincent Walsh, and Laurent Cohen, Padraic Monaghan and Richard Shillcock. During that trip, Jonathan Grainger also invited me for an interesting visit to his lab. Recently, I have also had the pleasure of corresponding with Piers Cornelissen, who is propelling me to consider new avenues of research, such as reading acquisition and dyslexia.

I would also like to thank Rita Berndt, for whom I worked prior to starting my Phd studies. She gave me the freedom to investigate some interesting data, which launched me on the present path.

Finally, my appreciation to my husband, Udaya Shankar, whose support has sustained me in myriad ways.

## Contents

<b>List of Tables</b>	<b>xi</b>
<b>List of Figures</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
<b>2 Introduction to The Problem of Letter-Position Encoding</b>	<b>5</b>
2.1 Definition of LPE . . . . .	5
2.2 Why Study LPE? . . . . .	7
<b>3 Neurobiological Constraints on LPE</b>	<b>10</b>
3.1 Terminology and Overview of the Visual System . . . . .	10
3.2 Retina to V1 . . . . .	11
3.3 Higher Cortical Areas . . . . .	14
3.4 Summary . . . . .	18
<b>4 Behavioral Results on LPE</b>	<b>19</b>
4.1 Word-Level Studies . . . . .	20
4.1.1 Masked Form Priming . . . . .	20
4.1.2 Positional Patterns . . . . .	25
4.1.3 Seriality . . . . .	26



4.1.4	Summary . . . . .	30
4.2	Letter-Level Experiments . . . . .	31
4.2.1	Fixation at String Center . . . . .	32
4.2.2	Non-central Fixation within a String . . . . .	34
4.2.3	Unilateral Presentation . . . . .	36
4.2.4	Summary . . . . .	40
<b>5</b>	<b>Models of LPE</b>	<b>41</b>
5.1	Desiderata for an LPE Model . . . . .	41
5.2	Review of Modeling Basics . . . . .	43
5.3	Models of LPE . . . . .	46
5.3.1	Interactive Activation Model . . . . .	46
5.3.2	Print-to-Sound Models Trained by Back-Propagation. . . . .	47
5.3.3	BLIRNET . . . . .	48
5.3.4	A Split Fovea Model Trained by Back-Propagation . . . . .	50
5.3.5	SOLAR . . . . .	52
5.3.6	LEX . . . . .	56
5.4	Summary . . . . .	57
<b>6</b>	<b>The SERIOL Model of LPE</b>	<b>59</b>
6.1	Overview . . . . .	59
6.1.1	Highest Prelexical Orthographic Representation . . . . .	59
6.1.2	Nature of Pre-Bigram representation . . . . .	60
6.1.3	Induction of Serial Encoding . . . . .	61
6.1.4	Creation of the Locational Gradient . . . . .	63
6.1.5	Summary . . . . .	65
6.2	SERIOL model . . . . .	66

6.2.1	Edge Layer to Feature Layer . . . . .	67
6.2.2	Feature Layer to Letter Layer . . . . .	70
6.2.3	Letter Layer to Bigram Layer . . . . .	71
6.2.4	Bigram Layer to Letter Layer . . . . .	73
6.3	Summary . . . . .	73
<b>7</b>	<b>Account and Simulations of LPE Behavioral Results</b>	<b>76</b>
7.1	Word Level . . . . .	76
7.1.1	Bigrams . . . . .	76
7.1.2	Letters . . . . .	86
7.2	Letter Perceptibility Patterns . . . . .	95
7.2.1	Mathematical Model . . . . .	98
7.2.2	Short Strings . . . . .	103
7.3	Summary and Discussion . . . . .	109
<b>8</b>	<b>Asymmetry of the Length Effect</b>	<b>112</b>
8.1	Experimental Data . . . . .	112
8.2	SERIOIOL Account of the Length Effect . . . . .	116
8.3	Length Investigation . . . . .	117
8.4	Discussion . . . . .	124
<b>9</b>	<b>Asymmetry of the N effect</b>	<b>127</b>
9.1	The N effect . . . . .	127
9.2	The SERIOIOL Account of the N effect . . . . .	129
9.3	Predictions . . . . .	133
9.4	N-effect Investigation 1 . . . . .	137
9.5	Further Predictions . . . . .	143

9.6	N-effect Investigation 2 . . . . .	144
9.7	Implications . . . . .	147
9.8	General Discussion . . . . .	153
<b>10</b>	<b>SERIOIOL Speculations</b>	<b>156</b>
10.1	Innate versus Learned Aspects of the SERIOIOL Model . . . . .	156
10.2	Object Recognition . . . . .	158
10.3	Feature-Level Processing and Dyslexia . . . . .	161
10.3.1	Simulation of Learning to Form the Locational Gradient . . . . .	162
10.3.2	Dyslexia . . . . .	163
10.3.3	Magnocellular Deficit . . . . .	165
10.3.4	Possible Experimental Tests of these Proposals . . . . .	167
10.3.5	Summary . . . . .	169
<b>11</b>	<b>The Parsing Problem</b>	<b>171</b>
11.1	Specification of the Problem . . . . .	171
11.2	Computational Constraints . . . . .	173
11.3	Neurobiological Constraints . . . . .	178
<b>12</b>	<b>Behavioral Results on Parsing</b>	<b>181</b>
12.1	Complexity Phenomena . . . . .	182
12.1.1	Center-Embedding versus Crossed-Serial Dependencies . . . . .	182
12.1.2	Different types of English doubly center-embedded clauses . . . . .	183
12.1.3	Interference in Working Memory . . . . .	184
12.1.4	NP-type effects . . . . .	186
12.1.5	The RC/RC V2-drop effect . . . . .	188
12.1.6	V2-drop x N3-type Interaction . . . . .	189

12.1.7	Summary . . . . .	190
12.2	Accounts . . . . .	190
12.2.1	Vosse & Kempen [Vos00] . . . . .	190
12.2.2	Interference in Working Memory . . . . .	191
12.2.3	Dependency Locality Theory . . . . .	192
12.2.4	Summary . . . . .	196
<b>13</b>	<b>Parsing Models</b>	<b>197</b>
13.1	Representation of the Thematic Tree on a Computer . . . . .	198
13.1.1	How . . . . .	198
13.1.2	Difference from Neural Networks . . . . .	201
13.2	Possible Neural Network Representations of the Thematic Tree . . . . .	202
13.2.1	Production of an New Pattern . . . . .	202
13.2.2	Temporal Encoding . . . . .	207
13.2.3	Summary and Conclusions . . . . .	210
13.3	Parsing Models . . . . .	211
13.3.1	SRNs . . . . .	211
13.3.2	LTSMs . . . . .	215
13.3.3	Pulvermuller . . . . .	217
13.3.4	Summary . . . . .	219
<b>14</b>	<b>The TPARRSE Model</b>	<b>220</b>
14.1	RR encoding . . . . .	222
14.1.1	Primitives . . . . .	222
14.1.2	Representation of the Thematic Tree . . . . .	223
14.1.3	Generating the RR encoding . . . . .	228
14.2	Temporal Working Memory . . . . .	233

14.2.1	Primitives . . . . .	235
14.2.2	Representation of Syntactic Information . . . . .	240
14.3	Processing Center-embedded Clauses . . . . .	243
14.4	Partial Deletion . . . . .	246
14.5	Arbitrary Hierarchical Structure . . . . .	248
<b>15</b>	<b>Computational Demonstrations</b>	<b>250</b>
15.1	Decoding an RR encoding . . . . .	250
15.2	Temporal WM . . . . .	253
15.3	Parsing Algorithm . . . . .	255
15.3.1	Implementation . . . . .	255
15.3.2	Stimuli . . . . .	256
<b>16</b>	<b>Complexity</b>	<b>265</b>
16.1	Center Embedding . . . . .	266
16.1.1	RC/RC . . . . .	266
16.1.2	Noun Complements . . . . .	269
16.1.3	Summary . . . . .	271
16.2	Crossed-Serial Dependencies . . . . .	272
16.3	Interference in Working Memory . . . . .	274
<b>17</b>	<b>Conclusion</b>	<b>276</b>
17.1	Future TPARRSE Research . . . . .	276
17.2	Conclusion . . . . .	278
	<b>Bibliography</b>	<b>280</b>

## List of Tables

4.1	Results from Exp. 1a - 1c of [Hum90]. Each group of rows represents a sub-experiment. $Fac = (\text{accuracy for prime} - \text{accuracy for control})$ , where 0 denotes no significant facilitation. Stimuli with the same facilitation were not statistically different from each other; the given value reflects their average. . . . .	21
4.2	Results from experiments 4 through 6 from [Hum90]. Each group of rows represents the results from a single experiment. $Fac = (\text{accuracy for prime} - \text{accuracy for control})$ , where 0 denotes no significant facilitation. Stimuli with the same facilitation were not statistically different from each other; the given value reflects their average. . . . .	23
7.1	Simulated and experimental results for priming conditions from [Gra04a]. $Act$ denotes the activation of the target node in the simulation for the given prime. $Fac$ denotes the the facilitation for that prime in the experimental results (difference between reaction times for the control condition (dddd or ddddd) and the prime condition), where * denotes facilitation is statistically significant. The top group is five-letter targets; the middle group is seven-letter targets and the bottom group is nine-letter targets. . . . .	81
8.1	Results for word targets. . . . .	122

8.2	Results for non-word targets. . . . .	123
9.1	Stimuli for N-effect investigations. . . . .	138
9.2	Results for N-effect investigation 1. . . . .	141
9.3	Results for N-effect investigation 2. In the dimmed condition, the outer two letters were dimmed for RVF and LVF presentation, while only the first letter was dimmed for CVF presentation. . . . .	146
14.1	WM variables after each item <b>x</b> is processed from sentence 39. The relative pronoun <i>that</i> introduces the predicate <b>C</b> and starts a new clause, giving <b>TotRR</b> = <b>sue</b> + <b>likes@(the</b> + <b>vase)</b> . It also causes its referent, <b>the</b> + <b>vase</b> , to be stored, so that it can be accessed when a gap is encountered. During processing of the relative clause, the parser determines that the object of <i>bought</i> is a gap, corresponding to the referent of the relative pronoun. At the end of the sentence, chunking is invoked, yielding the final value of <b>TotRR</b> given in the text. . . . .	232

## List of Figures

4.1	Results from [Wol74], with LVF/RH on left and RVF/LH on right. Each line represents a fixed retinal location. As string position is increased (i.e., more letters occur to the left), performance decreases. The pattern of decrease varies with visual field. . . . .	37
4.2	Results from [Est76], for the 2400 ms exposure duration. . . . .	38
5.1	Basic components of an implemented model. Each node has an activation value (shown in the center of the node). At the lowest level of the model, activation values are clamped to particular values. Each connection has an associated weight. The input to a node . . . . .	44
6.1	Interaction of input level and timing of firing for a cell with undergoing a sub-threshold oscillation of excitability. When a relatively high level of input (top curving line) is added to the base oscillation, the cell crosses threshold at time 1 (action potential not illustrated). If less input were received, the cell would cross threshold later in the cycle, such as at time 2. . . . .	62



6.2	Architecture of the letter, bigram, and word levels of the SERIOL model, with example of encoding the word CART. At the letter level, simultaneous graded inputs are converted into serial firing, as indicated by the timing of firing displayed under the letter nodes. Bigram nodes recognize temporally ordered pairs of letters (connections shown for a single bigram). Bigram activations (shown above the nodes) decrease with increasing temporal separation of the constituent letters. Activation of word nodes is based on the conventional dot-product model. . . . .	64
6.3	Formation of the locational gradient at the feature layer, for the centrally fixated stimulus CASTLE. The horizontal axis represents retinal location, while the vertical axis represents activation level. The bold-face letters represent bottom-up input levels, which are higher in the RH than the LH. In each hemisphere, activation decreases as a eccentricity increases, due to the acuity gradient. The italicized letters represent the effect left-to-right inhibition within the RH, and RH-to-LH inhibition. In the RH, C inhibits A, and C and A inhibition S, creating a decreasing gradient. The RH inhibits each letter in the LH by the same amount, bringing the activation of T lower than that of S. As a result, activation monotonically decreases from left to right. . . . .	69
7.1	Comparison of simulated score and amount of facilitation using data from Table 7.1 ( $r=.87$ ; $p < .0001$ ). . . . .	82

7.2	Experimental [Whi99] and simulated results for the aphasic error pattern. The <i>percent retained</i> refers to the percentage of erroneous trials in which the letter in the $i$ th position in the target occurred in the $i$ th position the response ( $n = 201$ for experiment; $n = 363$ for simulation). Data are collapsed over target lengths of three to six. (In the both the experimental data and the simulation, there was also a decreasing pattern within each target length.) . . . . .	84
7.3	Simulation results under backward scoring, and no inhibition. In backward scoring, the target and response are aligned at the final letter, and scored from right to left. In this case, position 1 corresponds to the final letter, 2 corresponds to the next-to-last letter, etc. The backward results are from the same simulation run as Figure 7.2. For the no-inhibition condition, a new simulation was run with $C_{inh} = 0$ , and scored in the forward manner. Because backward scoring yielded a relatively flat pattern, and no inhibition yielded a V-shaped pattern, this shows that the decreasing pattern in Figure 7.2 was not merely an artifact of the scoring method. . . . .	85
7.4	Experimental reaction times (in milliseconds) for the rotated-string lexical-decision task. Each line represents one angle of rotation, where the lower lines correspond to $0^\circ$ through $80^\circ$ , and the upper lines correspond to $100^\circ$ to $180^\circ$ . . . . .	87
7.5	Simulated reaction times for the rotated-string, lexical-decision task. Notation is the same as Figure 7.4. . . . .	94

7.6	Schematic of locational gradients for the stimulus CART at three different presentation locations. The vertical axis represents activation, while the horizontal axis represents retinal location. For central presentation, the gradient is smoothly and rapidly decreasing. For RVF presentation, the gradient is shallower because the acuity gradient is shallower. For LVF presentation, the initial letter strongly inhibits nearby letters, but the gradient flattens out as acuity increases. . . . .	97
7.7	Experimental (top) and modeled (bottom) results of [Wol74], with LVF presentation on the left and RVF on the right. Each graph shows the effect of string position on perceptibility at a given retinal location (specified in $R$ units of letter width). . . . .	99
7.8	Experimental results from [Est76] for a four-letter string embedded in \$'s, occurring at two different retinal locations in each visual field. Exposure duration was 2400 ms. (Subjects were trained to maintain central fixation, and their gaze was monitored.) . . . .	104

- 7.9 Locational gradient and resulting firing pattern for LVF/RH presentation (normal font) and RVF/LH presentation (bold italics). Top: Comparison of locational gradient for string CDFG under RVF/LH presentation and LVF/RH presentation. Bottom: Cartoon of resulting firing pattern at the letter level. The point in the oscillatory cycle at which the down phase prevents further firing is marked \*. In the LVF/RH, the first letter fires faster and longer than the other letters, because it receives a much higher level of input. The variations in the amount of bottom-up input create decreasing activation across the string. The final letter starts firing late in the cycle, and is soon cut off by the end of the oscillatory cycle, giving no final-letter advantage. In the RVF/LH, each letter rapidly cuts off firing of the previous letter, allowing the final letter to fire a long time. As a result, activation is flat across the string and rises for the final letter. These firing patterns account for the perceptibility patterns at the larger eccentricities in Figure 7.8. . . 106
- 7.10 Results from Experiment 2 of [Leg01] for the two largest eccentricities, grouped by exposure duration, with 95% confidence intervals. 108

8.1	<p>Example of proposed LVF/RH locational gradient for normal presentation (bold face) and under contrast manipulation (italics, shifted to the right for clarity) for a six-letter word. Horizontal axis represents retinal location, while vertical axis represents activation level at the feature layer. For normal presentation, the locational gradient is not smooth, becoming quite flat near fixation. Increasing the contrast of the second and third letters raises their activation levels, and decreases the activation levels of the fourth and fifth letters due to increased left-to-right inhibition. Decreasing the contrast of the sixth letter decreases its activation level. As a result, the locational gradient is more smoothly decreasing.</p>	118
8.2	<p>Results for word targets. . . . .</p>	124
9.1	<p>Outer dimming in the LVF/RH. The normal locational gradient is shown in bold-face. The results of outer dimming are shown in italics (shifted to the right for clarity.) Reducing the contrast of the first letter reduces its activation level, and decreases inhibition to the second and third letters, increasing their activation levels. As a result, the locational gradient is shallower across the first three letters. Reducing the contrast of the fourth letter reduces its activation level. As a result, the locational gradient is smoother across the last three letters. . . . .</p>	134
9.2	<p>Predicted pattern for Experiment 2. . . . .</p>	136
9.3	<p>Results for N-effect investigation 1. . . . .</p>	142
9.4	<p>Results for N-effect investigation 2. . . . .</p>	146

- 11.1 Examples of finite state machines (FSMs). Each recognizer consists of a start state,  $S$ , and an accept state,  $A$ , and intermediate (numbered) states. Transitions occur between states for specific input tokens, where  $e$  represents the end-of-string token. The top FSM accepts strings of the form  $a^n b^m$ , for  $n \geq 1$  and  $m \geq 1$ . For example, the string  $a_1 b_1 b_2 b_3$  would activate the following sequence of states:  $S, 1, 2, 2, 2, A$ . The bottom FSM accepts strings of the form  $(ab)^n$ , for  $n \geq 1$ . For example, the string  $a_1 b_1 a_2 b_2$  would activate the following sequence of states:  $S, 1, 2, 1, 2, A$ . . . . . 175
- 11.2 Example of using a stack to recognize strings of the form  $a^n b^n$ . A stack  $S$  provides the  $push(S, x)$  operation, which puts  $x$  on the top of the  $S$ , the  $pop(S)$  operation, which removes the top item from  $S$  and returns it, and the  $empty(S)$  operation, which is true only if there are no items on  $S$ . The string  $a^n b^n$  can be recognized using the following algorithm for token  $x$ : . . . . . 176

13.1 Example of encoding *Mary knows that Ted likes Sue* in computer memory. The left column represents memory addresses, which systematically increase. The right column represents registers. The programmer would declare a record having *Agent*, *Verb*, and *Theme* variables. For each instance of this record the compiler would map these variables onto specific consecutive addresses. Here the record *Main* starts at 1200 and the record *Sub* starts at 1392. The value of *Main's* *Theme* variable is a pointer to *Sub*. *Mary*, *knows*, *Ted*, *etc.* correspond to numbers that have been associated with each token. (For simplicity, the problem of how to determine whether a register's value should be interpreted as a memory address is ignored. ) . . . . . 200

13.2	Example of network that learns to form an RR encoding. Each box represents a group of nodes of the same size, and each arrow represents full interconnectivity between two groups of nodes. For each training item, the input and output layers are set to the same value. Using the back-propagation training algorithm, the network learns to recreate the input on the output layer. As a result, the hidden layer (in conjunction with the learned weights) forms a condensed representation of the input. This condensed representation could then be used as one of the values on the input layer. For example, in the <i>Mary knows Ted likes Sue</i> example, the patterns for <i>Ted</i> , <i>likes</i> , and <i>Sue</i> would first be activated over the corresponding sets of input nodes. The resulting pattern on the hidden layer constitutes an RR encoding of this information. Then the input layer is set to <i>Agent = Mary</i> , <i>Verb = knows</i> , and <i>Theme = the hidden layer pattern</i> . The new hidden layer pattern then represents the encoding of the entire sentence. Such an encoding is decoded by activating the pattern on the hidden layer to get the component values on the output layer. An output item that is itself an RR encoding can then be fed back to the hidden layer again to be decoded. . . . .	204
13.3	Example of bind and merge operations. . . . .	205
13.4	Example of temporal encoding of <i>Ted = Agent</i> and <i>Sue = Theme</i> . The lines to the right of each node represent the firing pattern for that node. For simplicity, each word and role is represented here as a single node. However, the same type of encoding could be used for a distributed representation of each item. . . . .	208



13.5	Architecture of a recurrent network. The hidden units connect into the context units, which feed back to the hidden units. Thus the hidden units' previous activations can affect their subsequent activations. . . . .	212
13.6	Example of detector S which recognizes sequence A B, from [Pul03].	218
14.1	Basic algorithm for generating the RR encoding of a sentence having only right-branching clauses. . . . .	234
14.2	Illustration of timing of firing of list elements A, B, and C. Each new element is activated at the peak of the oscillatory cycle. Previously activated items move forward with respect to the cycle, due to the ADP. Over time, A, B, and C come to fire successively within a single cycle. . . . .	236
14.3	Proposed architecture for a WM list, illustrated for positions N to N+2. In this example, 100, 110, and 001 are encoded across those positions on successive oscillatory subcycles. Each large circle represents a bank of nodes coding for the same value and position. A subset of those nodes is shown by the small circles. Each column represents a vector position. The top row encodes 0's, while the bottom row encodes 1's. The number in each node reflects the oscillatory subcycle in which it fires. Fast connections coordinate firing within a sub-cycle, while slower inhibitory connections separate subcycles. . . . .	239

14.4	Proposed architecture of deletion network. The tag field is comprised of syntactic features F1, F2, F3 ... Fn, with multiple instances of each feature (two instances shown here). Each feature has inhibitory connections to the corresponding feature in Dtag, and each feature in Dtag inhibits the node which drives the deletion process. When the tag-field features inhibit all of the Dtag features, the perform-deletion node is activated and deletion is initiated. Deletion is sustained via the self-excitatory connection. The gating node becomes activated only if it receives excitation from both the perform-deletion node and the list node. In that case, the list node is inhibited. Thus inhibition only applies to active list nodes, and does not affect list nodes that fired prior to the initiation of deletion. (Only a single list node is shown. A similar circuit is required for each list node.) . . . . .	247
15.1	Chunking and branching procedures for the full RR encoding algorithm . . . . .	257
15.2	Full RR encoding algorithm, using Chunk and Branch operations specified in Figure 3. . . . .	258

# Chapter 1

## Introduction

### 1.1 Overview

The ultimate goal of computational neuroscience is to specify how cognition arises from neural activity. This requires understanding how neurons represent information about the world. One of the more challenging aspects of such an investigation is the question of how structured information is represented. That is, how are the sub-parts of an entity encoded? It is not sufficient to simply encode their identities. Rather, the *relationships* between sub-parts must be represented. My overarching interest is to investigate the nature of such structured representations. I will first address this question in a limited visual domain (visual word recognition) and then in a linguistic domain (creation and representation of a syntactic tree).

In this work, I will distinguish between implemented and theoretical models. An *implemented model* refers to a simulation or a mathematical demonstration. In contrast, a *theoretical model* is a framework specifying the nature of the computations that are carried out in the brain. All or part of a theoretical model can be implemented to demonstrate the validity of related claims, essentially by offering an existence proof. This requires choosing specific functions and parameters

for the implementation. Thus an implemented model is but a single instantiation of a more general theoretical framework. Reasoning about a theoretical model can often be more fruitful than building an implemented model. I am primarily interested in formulating theoretical models, using implementations of portions of the resulting models as a proof of concept. Throughout this work, *model* will refer to either a theoretical model or an implemented model when there is no ambiguity as to which is meant. Otherwise, the type of model will be specified.

In particular, I seek to understand what representations and transformations are used by the brain once a task has been learned. Formulation of such a theoretical model requires in-depth consideration of all available sources of information. Behavioral data indicate what algorithms the brain is using. Neurobiological and anatomical data constrain how these algorithms are realized in tissue. The goal is to create a model that explains the behavioral data and can be mapped onto a neural network. Ideally, such a theory should lead to novel, experimentally verifiable predictions.

I have used this approach in two domains. The first is the question of how the brain encodes letter position in a string during visual word recognition. This model specifies how a retinotopic representation is progressively converted into an abstract encoding of letter order. Location-invariance is achieved by creating a temporal (serial) representation of letter order. The model is consistent with the neuroanatomy of the lower levels of the visual system, and explains a wide range of letter perceptibly and form-priming data. Moreover, the model has generated precise predictions concerning the source of visual-field asymmetries, which have been experimentally confirmed.

As we will see, there are many novel aspects to this model:

- The effect of visual acuity is explicitly considered.

- The retinotopic representation is initially split (across the hemispheres).
- Hemisphere-specific processing is proposed at the feature level.
- A location-invariant representation is created by mapping space onto time.
- Representational units based on ordered letter pairs are proposed.
- The model has provided new insights into the source of visual-field asymmetries at both the letter and word levels.

The second domain that I have investigated is the question of how the brain creates the representation of a parse of a sentence. The letter-position model has informed this parsing model; there is a serial representation of phrases in stored working memory. In addition to this serial representation, there is also a distributed representation of sentence structure. The interaction of these two types of representations allows a comprehensive account of phenomena related to sentence complexity. Novel aspects of the model include:

- The proposal that working memory uses dual, synchronized sequences to encode syntactic information.
- Specifications of a parsing algorithm and a hierarchical representation which are based on the computational properties of a predefined distributed representation.
- An account of complexity phenomena that is not based on storage limitations, but rather arises from the way in which syntactic information is encoded in working memory.

These computational theories demonstrate the feasibility of bridging the neural and cognitive levels via the close integration of modeling and experimental work.

A similar method of presentation will be used for both models. First, the computational problem is introduced and specified. Then the anatomical and neurobiological constraints are addressed, followed by a review of the relevant behavioral data. Previous models are presented, and their ability to meet these constraints is discussed. My model is then overviewed at a high level, and then given in detail. Implementations of portions of the model are presented, followed by experimental results (for the letter-position encoding model). A description of future work concludes the discussion of each model.

## Chapter 2

### Introduction to The Problem of Letter-Position Encoding

In this chapter, I define the problem of letter-position encoding, and discuss why it is an excellent problem for investigating structured neural representations.

#### 2.1 Definition of LPE

Letter-position encoding (LPE) is required during visual word recognition due to the existence of anagrams. That is, letter identities are not sufficient to uniquely identify a word because there may be several words comprised of the same letters. For example, the letters A,I,R,L can be used to form the words LAIR, LIAR, RAIL (and others). Thus there must also be some encoding of the position or order in which the letters occurred. This encoding of the input is then compared against stored representations in order to recognize the word.

Therefore, at the highest level, the problem of LPE is the question of what type of sublexical orthographic encoding maps onto the lexical level during visual word recognition. For example, such an encoding might be *position-specific*, with separate representations for each letter in each position. Under this scheme, the input *liar* is represented by activating units  $L_1$ ,  $I_2$ ,  $A_3$  and  $R_4$ , whereas the input *rail* is represented by activating  $R_1$ ,  $A_2$ ,  $I_3$  and  $L_4$ . Here  $L_1, L_2, L_3$  etc.,

are encoded by different sets of neurons. In contrast, there may be position-independent letter units which can dynamically represent positional information in some way. Alternatively, a letter's position may not be represented explicitly, but rather its context may be encoded. For example, the input *liar* could activate *\*LI*, *LIA*, *IAR* and *AR\** units, where *\** denotes a word boundary. Thus, this representation specifies relationships between letters (i.e., each unit specifies the letters immediately to the right and left of the central letter).

However, in understanding how the brain represents structured information, it is not sufficient to merely address the nature of the high-level representations. It is also necessary to understand how those representations are formed. Thus, taking a comprehensive approach to LPE, I also investigate the question of how such an abstract sub-lexical representation is created from an early, location-specific representation. It is well-known that the earliest cortical visual levels are retinotopically organized. That is, each letter occurs at a specific spatial location on the retina, and this spatial organization is maintained into the cortex. How is an abstract representation of letter position created from input that is tied to specific retinal locations? In the following, *position* will refer to a letter's position within a string, while *location* will refer to a letter's location in the visual field and hence on the retina.

One aspect of processing not included in the question of LPE is how letters are recognized. Rather, the question is more abstract. Given the ability to recognize letters, how is string-centered positional information calculated, and combined with letter identity information?

This work assumes an experienced reader, under the assumption that brains solve the problem in a standard way, as discussed below. The details of how the relevant transformations and representations are learned during reading acqui-



tion is left for future work, although some speculations on this topic are included in Chapter 10.

Note also that a full model of visual word recognition is not being sought. For example, phonological and frequency effects will not be considered. Rather, the focus is on an orthographic route to the lexicon. However, I do note that the encoding of letter order must also subserve the learning of grapheme-phoneme correspondences, and thus it must also be suitable for this task. This topic is also briefly addressed in Chapter 10.

A viable theoretical model of LPE should be consistent with relevant neurobiological and behavioral data. The lowest level of the model should employ a retinotopic representation, and the highest should employ a lexical representation. The model should specify the nature of the representations at in-between levels, and the transformations between levels. An important criterion is that the transformations should be biologically plausible. That is, they should employ known neural encoding mechanisms, or be compatible with the type of local, numeric computations that can be carried out by a network of simple, neuron-like units.

## **2.2 Why Study LPE?**

LPE is an ideal arena for studying structured neural representations, because it is complex enough to be interesting, but simple enough to be tractable. Moreover, the processing must tap into basic neural mechanisms because there can be no specific adaptation for reading due to its recent appearance on the cognitive scene.

One central, outstanding issue in cognitive science is the *binding problem*.

How are separate features combined within a single object? For example, consider color and shape. It is known that these attributes of an object are processed in separate areas of the brain. When you see a red square and a yellow triangle, how are the correct associations between color and shape encoded (so that you don't perceive a yellow square and a red triangle)? A similar problem exists in LPE. How are a letter's identity and its position bound together?

Another important issue is how location invariance is obtained during object recognition. How are we able to recognize the same object at different locations and sizes? Given that the input to the visual system is retinotopic, either a recognizer must be duplicated over and over for differing retinal locations, or there must be a mechanism to abstract away from retinal location before input reaches the recognizer. It is clear that the replication method is used by the visual system for low-level features, such as edges. Such replication is highly inefficient for complex objects, since so many different objects are possible. Thus it is unlikely that it is employed for high-level object recognition.

However, it has been claimed that a single detector could recognize an object in a location-invariant way without an explicit abstraction capability, as receptive-field size and featural complexity increase through the processing hierarchy. This approach has been implemented for recognition of simple objects, where the identity of low-level features (i.e., types of line intersections) is sufficient to recognize an object [Fuk88, Ore00]. It has also been used in implemented model of visual word recognition [Moz91], but as discussed in section 5.3.3, this model relied on an unrealistic jump in receptive-field size. As discussed by Hummel and Biederman, a recognizer that is based on feature conjunctions has the inherent limitation that it is susceptible to illusory recognition, wherein recognition is erroneously triggered by a set of jumbled features that have the correct

identities, but not the correct relationships to each other [Hum92]. This difficulty occurs because relationships between sub-parts are not explicitly represented. In contrast, an abstraction mechanism that specifically maintains relational information while removing locational information would not have this problem. As discussed above, a similar problem arises in LPE. How is a representation that is initially tied to retinal location converted to an abstract letter-order encoding that can be matched against a stored representation?

Thus the question of LPE involves key problems in cognitive science. At the same time, it is a very circumscribed problem, allowing ease of investigation. It involves a small number of known, basic units (i.e., letters) which can be organized along a single dimension (i.e., string position). Bottom-up aspects of stimuli can be experimentally varied, by manipulating retinal location, letter order, and contrast levels. Top-down factors can also be selected for, such as lexicality, frequency, length, reading direction, etc. Thus, the problem is easily investigated experimentally.

Recent brain imaging studies have identified a left-hemisphere, inferotemporal cortical area that seems to be involved in the abstract encoding of letter order, dubbed the Visual Word Form Area (VWFA) [McC03]. Interestingly, there is very little variation across subjects in the location of the VWFA [Coh02]. This suggests that brains solve the problem of LPE in a standard way [McC03]. However, this solution must rely on general representational mechanisms, due to the recency of reading on an evolutionary timescale. Thus, understanding how the brain solves the problem of LPE should reveal binding and abstraction mechanisms that are relevant to other domains.

## Chapter 3

### Neurobiological Constraints on LPE

The architecture of the visual system from the retinas to early cortical areas determines the characteristics of the input into the functional LPE network. I first discuss these constraints, and then review brain-imaging and neurological studies on higher cortical areas implicated in visual word recognition.

#### 3.1 Terminology and Overview of the Visual System

The cortex is divided into the right and left hemispheres, and the fibers which connect the hemispheres are known as the *corpus callosum*. Each hemisphere is comprised of four lobes. *Occipital* cortex lies at the back of the head. *Parietal* cortex lies above the occipital area, while *temporal* cortex lies forward of the occipital area. *Frontal* cortex lies in front of the parietal and temporal areas.

The visual image is initially projected onto the retinas. This visual information is processed through several layers of cells, and leaves the retina via ganglion cells. Ganglion cell axons extend through the optic tracts to the lateral geniculate nucleus (LGN), the visual area of the thalamus. LGN cell axons then extend to the cortex.

In the retina, there are two major classes of ganglion cells, magnocells and par-

vocells. The larger magnocells process information more quickly than the smaller parvocells. Magnocells are sensitive to motion and low spatial frequencies (i.e. overall shape), while parvocells are sensitive to color and high spatial frequencies (i.e., fine detail). Separate magnocellular and parvocellular pathways are maintained through the LGN into the cortex. The first cortical area to receive visual inputs lies in occipital cortex and is known as V1. V1 connects to V2, and then the visual pathway splits into two streams. The *ventral* stream extends through region V4 into lower temporal (inferotemporal) cortex. The ventral stream handles object recognition, and receives inputs from both the parvocellular and magnocellular pathways [Fer92]. The *dorsal* stream extends through region V5 into parietal cortex. The dorsal stream handles motion processing, spatial localization, and attention, and receives inputs primarily from the magnocellular pathway [Mau90].

We next consider the connectivity from the retina to the cortex in more detail, because the architecture of the early part of the visual system has ramifications for visual word recognition.

## 3.2 Retina to V1

Light coming into the eye is focused onto the retina, where it is transduced by photoreceptor cells (rods and cones) into electrical signals. Cones provide the high spatial resolution which is necessary for letter identification during reading. The center of the retina, the *fovea*, only contains cones and is free of blood vessels. Therefore, this area provides the highest acuity. It corresponds to about  $1.5^\circ$  of visual angle. (For reference, 4 or 5 letters occupy about  $1^\circ$  under normal reading conditions). Cone density (and therefore visual acuity) is highest at

the very center of the fovea (corresponding to the fixation point), and rapidly falls off away from the center. For example, at an eccentricity of  $0.17^\circ$  from fixation, cone density is decreased by 25% [Wes87]. The rate of decrease in cone density is highest closest to fixation, and falls off as eccentricity increases [Wes87].<sup>1</sup> Resolution remains elevated into the *parafovea*, the retinal region surrounding the fovea, corresponding to a diameter of about  $5^\circ$ .

Each cone cell projects to about three ganglion cells [Was95]. Ganglion cell axons from both eyes converge in the optic chiasm. There, the fibers from each eye split. Imagine a vertical line dividing each retina in half through the center of the fovea. Those fibers originating from the nasal side of this line cross the optic chiasm to enter the contralateral (opposite) optic tract, while those originating from the outer side of this line remain in the ipsilateral (same) optic tract. Therefore, after the optic chiasm, information is split by visual field, not by eye. Information from the left half of the visual field (LVF) is carried in the right optic tract, and information from the right half of the visual field (RVF) is carried in the left optic tract.

The spatial relationships between cells are maintained from the retina through the LGN and V1. Thus V1 is retinotopically organized, with nearby cells representing nearby points in space. Due to the routing of fibers at the optic chiasm, each visual field is projected onto the contralateral cortical hemisphere. That is, the LVF projects to the right hemisphere (RH) portion of V1, while the RVF

---

<sup>1</sup>This acuity pattern is commonly misrepresented as “acuity falls off rapidly outside the fovea”, implying that acuity is uniformly high across the fovea and then falls off. This is not the case. Rather, acuity falls off rapidly within the fovea, so that acuity is substantially reduced by the fovea / parafovea boundary (but still remains higher than outside the parafovea). The rate of decrease in acuity is actually sharper across the fovea than the parafovea.

projects to the left hemisphere (LH) portion of V1. The pattern of spatial resolution is magnified into V1. The number of cells representing a fixed amount of visual space is highest at fixation and decreases as eccentricity increases. As a result, a disproportionate amount of V1 is devoted to representing the fovea and the parafovea [Ino09, Bri68].

There has been some controversy regarding whether information is precisely split by visual half-field in humans, primarily due to the phenomenon of *macular sparing*. Often brain damage to one hemisphere of V1 obliterates vision in the contralateral visual field except for the foveal area. This suggests that the entire fovea may be bilaterally represented. That is, foveal vision may be spared because it also also represented in the undamaged hemisphere [Ino09]. Alternatively, foveal vision may be spared because damage to the lesioned hemisphere is incomplete, due to the large cortical area devoted to representing the fovea. This issue is highly relevant to visual word recognition, because if the visual fields do not overlap, the representation of a fixated string is initially split across the cerebral hemispheres, requiring downstream integration of the representations of the two halves of the string [Bry04]. (In fact, a special edition of *Brain and Language* was devoted to this topic [Bra04]).

Mounting evidence from several lines of investigation indicates that the representation of the fovea is indeed initially split across the hemispheres [Lav04b]. Behavioral experiments have shown that length and orthographic-neighborhood effects, which occur in the LVF but not the RVF under unilateral presentation, are also specific to the LVF portion of the string under central presentation [Bry94, Bry96, Lav01a, Lav04a]. Transcranial magnetic stimulation was used to disrupt neural function over either left or right V1 during processing of centrally presented strings. Unilateral disruption caused effects specific to the half of

the word presented in the contralateral visual field, as would be expected under a split fovea [Lav03]. Leff [Lef04] discusses several arguments against bilateral representation. There is no evidence in humans for the white matter pathways that would be required for such a representation. Also, “there has been no direct demonstration of this extra representation of ipsilateral central vision in human visual cortex, which, given the resolution of modern non-invasive techniques and the amount of cortex these regions must occupy if they are to support high acuity vision, is damning.” ([Lef04], p.276.) Furthermore, about 30% of hemianopia victims do not experience macular sparing, and so suffer from complete obliteration of a visual half-field [Lef04]. If the fovea were truly bilaterally represented, such a deficit pattern should not occur under unilateral damage. Thus, there is no positive evidence for bilateral representation of the fovea. The most likely source of macular sparing is incomplete damage to the affected hemisphere, due to the extensive cortical area devoted to representing the fovea [Lef04, Lav04b].

Therefore, available evidence indicates that the representation of the fovea is initially split across the hemispheres. This information must then be integrated into a unitary representation of a letter string. Recent studies indicate that specific cortical areas become specialized for this task.

### **3.3 Higher Cortical Areas**

Neuroimaging studies have provided converging evidence that areas of left occipital and inferotemporal cortex play a special role in reading. In an EEG study, normal readers showed a LH-specific increase in theta-band power (5 to 10 Hz) at occipital sites during reading, while dyslexics show reduced, bilateral theta-band activity [Kli01]. An MEG study [Tar99] has identified an early string-



specific response at approximately 150 ms post-stimulus in the posterior region of occipitotemporal cortex, where activation was stronger for letter strings than for strings of symbols. Response strength and latency in this area correlated with the speed with which subjects were able to read words aloud. For dyslexic subjects, this area did not show preferential activation for letter strings [Hel99].

EEG and fMRI studies have revealed a more anterior string-specific response in the LH beginning at about 180 ms post-stimulus. This activity has been localized to specific cortical coordinates ( $x = -43$ ,  $y = -54$ ,  $z = -12$  mm to the left, posterior and below the anterior commissure, respectively) [Coh02]. This area corresponds to an activation peak in about 90% of subject scans, with a standard deviation of 5 mm [McC03]. Thus the location of this response is remarkably uniform across subjects.

This area has been dubbed the Visual Word Form Area (VWFA) [McC03]. Although there is some debate on whether this area should be so labeled since it also responds to other types of stimuli and other areas also respond to letter strings [Pri03], there is strong evidence that this area becomes preferentially tuned to processing letter strings [McC03].

VWFA response is preferentially tuned to letter strings (as compared to arrays of pseudo-letters) [Nob94], but is insensitive to surface features of letters, such as font and case [Pol02, Deh04], and to their string position and retinal location [Deh04]. Activation is also insensitive to lexical features, such as frequency [Fie02], and whether a orthographically legal string is a real word or a pseudoword [Deh02]. However, activation is reduced in response to strings consisting only of consonants [Coh02]. VWFA activation is modality-specific, showing no response to passive listening of spoken words [Deh02]. The activation of the VWFA is independent of the location of the stimulus. For unilaterally presented strings,

fMRI showed contralateral activation up to an area probably corresponding to V4. Then, starting at the VWFA, activation was lateralized to the LH, independently of stimulus location [Coh00].

Damage to the region of the VWFA is associated with pure alexia, wherein lexical access via orthography is selectively impaired [Bin92, Bro72, Ges95, War80]. This impairment often does not cause a total inability to read, but rather causes slowed reading that is abnormally sensitive to word length (dubbed letter-by-letter reading). The abilities to write and to recognize orally spelled words are preserved. Lesions that are limited to the callosal connections between RH visual areas and the VWFA result in pure alexia that is specific to LVF stimuli [Coh00].

Thus the VWFA seems to convert a visually presented letter string into a location-invariant representation based on abstract letter identities. The results of [Coh00, Deh04] indicate that this prelexical representation is assembled in the LH, and that lexical access occurs in the LH. It is thought that letter-by-letter readers perform lexical access by representing a letter sequence in verbal working memory, rather than by the more efficient, direct route usually provided by the VWFA [Coh03]. Because writing and lexical access via indirect routes are preserved, it seems that the VWFA does not actually encode how words are spelled.

A pattern of acquired dyslexia (i.e., resulting from brain damage) observed in two Hebrew subjects suffering from left occipitoparietal lesions suggests that the encoding of letter identity can be separated from the encoding of position. These subjects made reading errors that were characterized by migration errors within a word; that is, errors were predominately anagrams of the target word [Fri01]. Such a dyslexia has not been encountered in more commonly studied languages, such as English. However, Hebrew orthography is particularly conducive

to revealing a deficit of this sort, since vowels are not explicitly represented. Therefore, if letter order is misperceived, there is a high probability that a word corresponding to the erroneous ordering exists for some combination of vowels. Thus, lexical constraints are reduced, allowing a pure deficit in position-encoding to be revealed.

The lesions in the above subjects occurred along the dorsal route of the visual system. A role for the dorsal pathway in encoding letter order is consistent with a study showing that ability to detect coherent motion in two-dimensional arrays of moving dots was correlated with accuracy in letter-position encoding (for lexical decision involving nonwords formed by transposing two letters of actual words) [Cor98]. This result is also consistent with evidence that developmental dyslexia is associated with subtly impaired magnocellular function [Lov93, Ste97]. However, it remains unclear whether such visual impairment is a causal factor in developmental dyslexia.

In contrast to these patterns of dyslexia, damage to the left angular gyrus results in complete illiteracy (global alexia). Such patients cannot read or write, or even name letters [Bro72, Ges95]. The angular gyrus is located at the junction of the occipital, temporal, and parietal cortices. Thus it seems to be a multi-modal association area. In the case of reading, the left angular gyrus is thought to subserve the translation of the orthographic encoding of a word into its phonological and semantic representations [Dem92, Ges95]. Therefore, this area seems responsible for encoding how words are spelled, and may provide the orthographic lexicon.

### 3.4 Summary

Letters to the right of fixation are initially projected to the LH, and letters to the left are projected to the RH. The representation in V1 is location-specific; each cell represents a stimulus occurring at a specific retinal location. The number of cortical cells representing a letter depends on the letter's eccentricity, following an *acuity gradient* which originates in the density of cones in the retina. Acuity is highest near fixation, and falls off as eccentricity increases. The rate of decrease in acuity decreases as eccentricity increases. At about 150 ms post-stimulus, cortical activation becomes left lateralized in response to letter strings (in normal readers). Areas of occipitotemporal and occipitoparietal cortex encode an abstract representation of letter order, which may contact lexical representations via the angular gyrus.

Thus, the location-specific representation of a string, which is initially split across the hemispheres in V1, is integrated into a location-invariant, letter-based encoding in the LH. However, neurological investigations and brain-imaging techniques cannot reveal how this transformation is performed. For clues to the answer to this question, I turn to the results of behavioral experiments.

## Chapter 4

### Behavioral Results on LPE

In this chapter, I review experimental evidence from behavioral studies. I first consider those studies relevant to the issue of what type of prelexical representation contacts the word level. In these studies, the target stimuli were words. I then consider studies in which targets were random letter strings. Such studies can reveal patterns at the letter level under reduced lexical influences.

As we will see, the studies indicate the following:

- The relative order of letters is important in word recognition.
- There are position-independent letter units. That is, there are abstract letter representations that are not specific to string position or retinal location.
- Letter perceptibility varies with string position and retinal location, and these patterns differ from those of non-letter symbols.
- The presence or absence of a length effect on RTs cannot reliably indicate whether lexical access proceeds serially or in parallel.
- There is a serial readout of the visual image.

## 4.1 Word-Level Studies

### 4.1.1 Masked Form Priming

The most informative experiments on the nature of the prelexical encoding have used the masked-priming procedure, wherein a mask (visual noise) is displayed, then a briefly presented lower-case prime (for 40 ms or less), then a mask, and then an upper-case target word [Eve81]. Such brief prime exposures lead to orthographic priming, but not semantic priming. Thus such experiments are ideal for investigating the nature of orthographic encoding. In the description of such experiments, the following notation is used for describing the relationship of the prime to the target. A target of length  $n$  is represented by  $123\dots n$  where 1 denotes the first letter, 2 the second letter, etc., and each letter is unique. The prime is specified in terms of these numbers, with “d” representing a letter not in the target. For example, the prime “rqgdzn” for the target GARDEN is denoted 3d14d6. This means that the first letter of the prime is the third letter of the target, the second letter of the prime is not in the target, etc.

Humphreys, Evett and Quinlan carried out an extensive series of masked form-priming experiments where the task was perceptual identification [Hum90]. The target word was briefly presented (for approximately 40 ms), and performance was measured in terms of accuracy in identifying the word, where responses were typed. In Experiment 1, absolute-position effects were investigated. All targets and primes were four letters. Facilitation was measured with respect to a dddd prime. Primes with 1, 2, and 3 matching letters in differing positions were used. The significant effects are given in Table 4.1. In summary, when 1 letter matched, priming was only observed when the match occurred in the first position. When 2 letters matched, priming was strongest when they were the first and fourth

Prime	Fac (% points)
1ddd	8
d2dd	0
dd3d	0
ddd4	0
12dd	6
d23d	6
dd34	6
1dd4	15
123d	20
1d34	20
12d4	20
123d	20

Table 4.1: Results from Exp. 1a - 1c of [Hum90]. Each group of rows represents a sub-experiment.  $Fac = (\text{accuracy for prime} - \text{accuracy for control})$ , where 0 denotes no significant facilitation. Stimuli with the same facilitation were not statistically different from each other; the given value reflects their average.

letters; matches in other positions gave reduced, equivalent levels of priming. When 3 letters matched, priming was independent of position.

In experiment 2, the effects of scrambled letters were investigated. Primes in which order was completely violated (e.g., 3142) produced no facilitation. Primes of the form 1324 and 1dd4 produced equivalent levels of facilitation.

An analysis of the errors to the dddd trials from Experiments 1 and 2 showed that letters in positions 1 and 2 of the target were more likely to be correctly

retained than letters in positions 3 and 4. Thus, while there was an external letter advantage for primes matching on two letters (in Experiment 1), this pattern was not replicated in the error data, where the final letter had no advantage.

Experiments 4 through 6 employed primes and targets of differing length, in order to investigate the effect of maintaining letter order, but not absolute position. For example, a prime of the form 1245 includes the fourth and fifth letters in the correct order, but in the incorrect positions. The results are displayed in Table 4.2. In summary, priming was greatest when the first and final letters remained in those positions and order was maintained among the internal letters. Primes matching on two contiguous letters gave equivalent levels of priming, while a prime matching on two non-contiguous internal letters did not produce priming. Letters did not have to match on absolute position in order for priming to occur. These results are considered evidence for a relative-position encoding, where the first and last letters are encoded as such, and the order of the internal letters is represented.

Peressotti and Grainger [Per98] investigated the properties of relative-position priming further. They used the lexical-decision task, wherein the subject specifies whether or not the target string is a word. Priming was measured in terms of decreased reaction times (RTs). This task is now generally preferred to the perceptual task because it is taken to isolate processing at the lexical level. That is, in the perceptual task, priming may occur at the letter level because letters are difficult to perceive due to the short exposure duration. This is not a factor in lexical decision, where target exposure duration is on the order of 200 ms. Thus any effects that do occur are more likely to be at a higher level.

For six-letter (word) targets, they found that a prime of the form 1346 speeded RTs as compared to a dddd control prime, whereas primes of the form 1436 and



Prime	Fac (% points)
1245	14
1425	7
1dd5	7
d24d	0
12dd	11
d23d	11
d34d	11
dd45	11
1ddd5	9
d1d5d	0

Table 4.2: Results from experiments 4 through 6 from [Hum90]. Each group of rows represents the results from a single experiment.  $Fac = (\text{accuracy for prime} - \text{accuracy for control})$ , where 0 denotes no significant facilitation. Stimuli with the same facilitation were not statistically different from each other; the given value reflects their average.

6341 did not yield facilitation. Thus unlike the results from Experiment 4 of [Hum90], where 1425 yielded some facilitation for five-letter words, 1436 did not yield facilitation for six-letter words. This may be a result of using different tasks, or may reflect the larger percentage of retained letters for five-letter targets. To test whether maintaining absolute position yields any advantage, they compared primes of the form 1346 with primes which included the “-” character in positions 2 and 5 (i.e., 1-34-6). There was no difference in the amount of facilitation provided by these two types of primes. Thus priming only occurred when relative-position was respected, and absolute-position information did not increase the facilitation.

In further investigations, Granier and Grainger explored positional effects in longer targets (seven- and nine-letter words) [Gra04a]. Primes consisting of the initial or final four or five letters of the target all produced facilitation with respect to dddd or ddddd primes. Across five experiments, a small numerical advantage for initial primes over final primes always occurred (ranging from 3 ms to 8 ms), but this difference was not statistically significant. They also performed a series of experiments with five-letter primes and seven-letter targets in which primes matched on the first and last letters and the positions of the missing letters was varied. Those primes having no more than one positional gap within the three central letters (3,4, and 5) induced priming; those primes which included more than one such gap did not. For example, 12457 (gap at position 3) and 13467 (gap at 5) produced facilitation, while 12467 (gaps at 3 and 5), and 12367 (gaps at 4 and 5) did not. Thus the proximity of the internal letters to each other seems to be important. This is line with the finding that d23d and d34d primed five-letter words, while d24d did not (Experiment 4 of [Hum90]).

In other experiments, the effects of transposing letters were investigated. For five-letter targets, primes of the form 12435 produced facilitation with respect

to 12dd5 primes [Per03]. This is in contrast to the results of [Hum90], where 1324 and 1dd4 were equivalent for four-letter targets, and 1425 and 1dd5 were equivalent for five-letter targets. However, primes of the form 12354 did not produce priming, indicating a special status for the final letter (and presumably the initial letter, but 21345 was not actually tested) [Per03]. Transposition of non-contiguous letters can also produce priming [Per04]. For six-letter targets, 125436 provided facilitation, while 12d4d6 did not. However, this result only held when 3 and 5 were both consonants. It is not clear if this specificity for consonants is due to a qualitative difference in processing consonants and vowels, or to statistical differences stemming from the fact that there are only 6 possible vowels.

Overall, these results show that it is unlikely that the brain uses a prelexical encoding based on absolute position. Rather, the encoding represents the relationships between letters. There does seem to be some positional influence, with greater priming when external letters are matched (as compared to internal letters) and an advantage for the first letter over the final letter when only one letter matches the target. In contrast to the priming data, the error data shows an retention advantage for the first and second letters over the third and fourth letters.

### **4.1.2 Positional Patterns**

Another potential source of information about how letter-position is encoded is the error pattern in a perceptual task. The probability of retaining a target letter in the response may be related to how the position of that letter is encoded. We have already seen that this probability was higher for the first and second letters than for the third and fourth letters in Experiments 1 and 2 of [Hum90].

In an experiment where words were presented very briefly (33 ms) without a prime, retention probability decreased monotonically across the string for five- and six-letter words [Mon98]. For longer words, there was an advantage for the final letter. Thus for four- to six- letter words, retention probability decreases across the string, showing no advantage for the final letter.

This pattern has also been observed in aphasic patients suffering from acquired dyslexia. An analysis of their reading errors showed that retention probability decreased with increasing letter position [Whi99]. This pattern was robust under a number of different scoring measures, and did not obtain when the response and target were aligned at the final letter and scored from right to left. Because this pattern is similar to normals' performance under very brief presentations, it likely reflects some aspect of normal processing, rather than being a result of altered processing due to brain damage.

### **4.1.3 Seriality**

A key question is whether lexical access proceeds serially (letter by letter) or in parallel (all letters activating the word level at the same time). It has generally been assumed that this issue can be decided via the presence or absence of a length effect. That is, if RTs were to increase with word length, this would indicate serial access; if RTs were independent of word length, this would indicate parallel access. Before discussing the experimental results, I wish to point out that this assumption is not necessarily warranted. Length may contribute multiple, even opposing, influences to RTs. For example, serial access could yield constant RTs with word length if the increased time that it takes for the final letter to fire for longer words is canceled out by decreased settling time at the word level. That is, for longer words, it may take less time for the lexical network to settle (following

activation by the final letter) than for shorter words, possibly due to an increased amount of bottom-up input from more letters. Conversely, if a length effect were observed, it could be a result of parallel access in conjunction with some other factor. For example, the reduced acuity of the outer letters in longer words could lead to increased RTs despite parallel access. Thus, the presence or absence of a length effect cannot definitively inform us as to whether lexical access proceeds letter-by-letter or in parallel.

For centrally presented words of three to six letters, it was found that string length has no effect lexical-decision RTs [Fre76]. This finding, in conjunction with the popularity of parallel-processing models (e.g. the Interactive Activation Model [McC81]), has led to the general assumption that lexical access proceeds in parallel.

However, a recent study has yielded a more complicated picture. New et al. [New04] undertook an investigation of the length effect based on the English Lexicon Project [Bal94], which is a on-line database of lexical-decision RTs for over 40,000 words. Once the effects of frequency, number of syllables and orthographic-neighborhood size [Col77] were factored out, they found that RTs actually *decrease* with increasing string length for words of three to five letters<sup>1</sup>, are constant with string length for words of five to eight letters, and increase with string length for words of eight or more letters. Thus string length has

---

<sup>1</sup>It is likely that the reason that this facilitatory effect of word length has not been previously observed is that the effect of orthographic-neighborhood size (N) was not factored out. N is the number of words that can be formed by changing one letter of the target to another letter [Col77]. High N is actually facilitatory [And97, New04] for words in lexical decision. Because longer words generally have lower N values than shorter words, the lack of N facilitation longer words may have masked the facilitatory effect of more letters. The N effect is discussed in more detail in Chapter 9.

differing effects over different lengths. It is highly unlikely that these differing effects reflect differences in the method of lexical access. Rather, these results most likely indicate that the effect of length is the sum of opposing forces, where the relative strength of the opposing forces varies with length. For example, increased bottom-up input (from more letters) may contribute a facilitatory effect, which reaches a ceiling level beyond a certain word length. Serial access and/or decreased acuity may contribute an inhibitory effect, which dominates at longer lengths.

The results of an EEG study [Hau04] are also consistent with the notion that there are differing components in the effect of length. In occipital regions, longer words gave increased amplitudes (as compared to shorter words) during the 80-125 ms period. After about 150 ms, this pattern reversed, with shorter words giving larger amplitudes than longer words. Effects of word frequency were seen at about 150 ms, indicating that lexical access had begun by that point. This suggests that there may be differential effects of string length for prelexical versus lexical processing.

The influence of length can also be varied by presenting stimuli in non-canonical formats. An inhibitory length effect occurs when the letters of a word are not horizontally aligned, or when they are presented in MiXeD cAsE [Lav02c]. A length effect can also be induced by rotating the stimuli. This phenomenon was examined in a Hebrew lexical-decision experiment where two- to five-letter strings were centrally presented and rotated as a whole in increments of  $20^\circ$ , from  $0^\circ$  (normal presentation) to  $180^\circ$  (upside-down) [Kor85]. For angles of  $60^\circ$  or less, RTs did not increase with string length. For  $80^\circ$ , RTs were similar for two- to four-letter words, but increased for five-letter words. For  $100^\circ$ , two- and three-letter words had similar RTs, with increasing RTs for four- and five-letter words.

For angles of 120° to 180°, RT varied approximately linearly with word length, with each additional letter adding about 200 ms. The non-word data showed a similar pattern with rotation angle, but with larger length effects.

Due to the size of the per-letter increment at the larger angles, it is likely that this increment *does* actually reflect serial processing. However, the data cannot be explained by supposing that processing switches from parallel to serial at some rotation angle, due to the intermediate region (80° and 100°) where RTs are neither constant nor linear with rotation angle. Note that it cannot be supposed that a such a switch occurs at differing angles for differing lengths. If that were the case, the RT for an  $n$ -letter word should either be close to that of the smaller angles or the larger angles, but not in between<sup>2</sup>. In fact, the authors state, “it is difficult to propose an interpretation of the results in terms of one unitary principle” (p. 504).

For canonical presentation conditions, the best way to investigate the issue of seriality is to use time directly. Harcum and Nice [Har75] used this approach in a clever experiment in which pairs of eight-letter compound words were very briefly sequentially presented. The pairs were selected to allow meaningful blends. For example, the words *headache* and *backrest* could be recombined to give *headrest* or *backache*. When fixating on the center of the string, subjects tended to report the first half of the first word, and the second half of the second word (e.g. for *headache* then *backrest*, *headrest* was reported). This result unambiguously shows sequential readout. The first half of the first word was processed first. By the

---

<sup>2</sup>This assumes a unimodal distribution of RTs; if RTs were bimodally distributed between these two extremes, their average *would* fall in between the two values. Although the authors do not explicitly state that RTs were unimodal for 80° and 100°, the nature of their discussion implies that they were.

time that the second half of the stimulus was reached, the stimulus had changed and the second half of the second word was processed.

They also included trials where fixation fell within the first half or the second half of the stimulus. For fixation within the second half, the same response pattern was observed as for central presentation. However, for fixation within the first half, the pattern reversed (e.g., *backache* tended to be reported instead of *headrest*). The authors took these results as evidence for left-to-right processing for central fixation, and peripheral-to-central processing for non-central fixation. However, there is a more parsimonious explanation, based entirely on left-to-right processing. As we discuss in more detail in section 8.1, fixation within the first half of a word provides the Optimal Viewing Position (OVP) and the fastest processing, as compared to other fixation locations [Ore84]. When fixation was at the OVP, there may have been time to process the first word in its entirety. Then the second word would have been processed starting at the beginning, overwriting the representation of the first word. The second word was presented more quickly than the first, so there may only have been enough time to process the its first half. Therefore, the response was comprised of the first half of the second word, and the second half of the first word.

Thus RT patterns cannot faithfully inform us whether lexical access occurs serially or in parallel. In contrast, the Harcum and Nice experiment provides direct evidence of serial processing.

#### 4.1.4 Summary

These studies indicate that the highest prelexical representation encodes relationships between letters, rather than the absolute position of individual letters. Priming showed an advantage for external letters, but error patterns showed



monotonically decreasing letter retention from left to right (for strings of six or fewer letters).

By using a stimulus that varied over time, it was directly shown that read out of the visual image occurred in a left-to-right manner for central presentation. Conflicting results for non-central fixation can be explained by faster processing at the OVP, allowing the spatial extent of the stimulus to be processed 1 1/2 times. In contrast, length effects are not suitable for diagnosing serial versus parallel processing, as the effect of length varies with length, while it is unlikely that the type of processing varies with length.

## 4.2 Letter-Level Experiments

Next I consider results from experiments which involved letter identification in briefly presented strings that were not orthographically legal. Such experiments should reflect bottom-up processing to the letter level in the absence of top-down lexical and phonological influences. Although some have argued that patterns evoked by processing of non-word strings are not relevant to word recognition (Grainger, pers. comm.), I argue in the following review that observed patterns must be a result of processing specific to visual word recognition.

First I focus on studies in which the target string was fixated in the center, and move to studies in which the target appeared in a single visual field. I will use the following notation to specify retinal location. A location is specified in units of letter widths (with fixation at 0), where the LVF has negative values. The absolute value of a location gives the eccentricity (distance from fixation). A string's location will be given by the locations of the first letter and the last letter, separated by a double colon. For example, if fixation falls on the final letter of

a five-letter string, the string is at -4::0. If fixation falls on the first letter, the string is at 0::4.

### 4.2.1 Fixation at String Center

In a priming study [Per95], Peressotti and Grainger investigated whether letter units are position-specific or position-independent by testing whether there was priming for the same letter across different string positions. Subjects were asked to perform an alphabetic decision task, where they determined whether or not strings of three characters consisted solely of letters (e.g., “TBR” versus “TB\$”). Critical primes were also trigrams, consisting of characters from the target string, either in the same order (same-position primes) or in a different order where no letter stayed in the same position (scrambled primes). The dependent variable was RT. In order to assure that facilitation did not result from physical overlap, prime strings and test strings were presented in fonts of different sizes.

To examine temporal effects, prime exposure duration was varied. The results varied with this duration. For durations of 33 ms, scrambled primes yielded no facilitation, while same-position primes sped RTs by 22 ms. For exposures of 50 ms and 67 ms, cross-position priming did occur, with facilitations of 9 ms and 14 ms, respectively, for the scrambled primes, while the level of same-position priming stayed roughly the same.

Thus priming was observed for the scrambled primes at the longer durations. This is in contrast to word targets, where no priming is observed for completely scrambled primes. Thus it seems that priming can occur at the letter level when relative position is not respected, but not at the word level. Because priming occurred at the shortest duration for same-position but not scrambled primes, the authors took this as evidence for activation of position-specific letter detec-

tors followed by position-independent letter units. However, the assumption of position-specific letter units may not be warranted. I have shown that these position-specific results could be accounted for by location-specific units with overlapping receptive fields [Whi01b]. Alternatively, the results might reflect an advantage for maintaining the relative, not absolute, position of the letters. In contrast, the cross-position priming results provide strong evidence for position-independent letter units; it is difficult to see how such priming could occur in the absence of such units.

Next I consider error patterns in tasks where letters in briefly presented strings are to be identified. Studies have generally shown that letter perceptibility decreases as string position increases, with the exception of the final letter (and possibly the fixated letter) [Wol74, Lef78, Ham82, Mas82].

The observed final-letter advantage is often taken as arising from reduced lateral inhibition at a low level of processing. That is, because the final letter is not masked by a letter to its left, it is more easily perceived. However, this account is in conflict with data from studies of non-letter symbols [Lef78, Mas82]. When a string of such symbols is fixated at its center, perceptibility is lowest at the first and last symbols, as would be expected from the effect of acuity. Thus, there is no advantage from reduced lateral masking. Therefore, reduced lateral masking cannot account for the final-letter advantage. Strings of numbers also show an advantage for the external numbers [Mas82]. Thus, characters that commonly occur in strings display an external character advantage, but other symbols do not. This indicates that the advantage arises from the way that strings are processed and encoded.

However, this pattern differs from the error patterns observed for degraded *word* identification (from aphasics and very brief presentations to normals), where

there is no final-letter advantage [Hum90, Mon98, Whi99]. One potential source of this difference is exposure duration. The letter identification tasks have used durations of 80 ms or more, while the word tasks for normals have employed durations of around 40 ms. (This difference reflects the fact that letter identification in non-words is more difficult than word recognition.) It may be the case that a final letter advantage only emerges at the longer durations. Alternatively, it may be the case that the nature of the stimuli themselves (words versus letters) underlies the difference.

Letter-level experiments have also given evidence for serial, left-to-right processing. Using the same paradigm as [Har75] (discussed in section 4.1.3), Nice and Harcum [Nic76] performed a letter-based experiment, where two six-letter strings were very briefly sequentially presented. Subjects tended to report the first letter of the first string, and the second to sixth letters of the second string. (The position of this cross-over point varied with subject.) This provides unequivocal evidence of serial processing; there was only time to process the initial letters of the first string before the stimulus changed to the second string.

### 4.2.2 Non-central Fixation within a String

For unilateral presentation, it is well known that there is an RVF advantage in visual word recognition. Thus, the advantage for early string positions (those falling in the LVF) over the later string positions (those falling in the RVF) under central fixation contrasts with the generally poorer performance observed for words presented entirely within the LVF. This pattern is also in evidence at the letter level. One study used strings consisting of a target letter embedded in a string of X's, where the task was to identify the non-X letter [Naz04a]. For five-letter strings at -4::0 versus 0::4, there was an RVF advantage. However

a nine-letter string at -4::4 produced a LVF advantage; the letters were in the same locations as in the two five-letter conditions, yet the VF advantage reversed. This suggests that the first half of the string affects the second half under central presentation, perhaps reflecting integration of the two halves. If so, there should be an RVF advantage for central presentation (-4::4) in a right-to-left language. A similar experiment in Hebrew confirmed this [Naz04a]. Also in contrast to English, performance did not differ for the two locations of the five-letter strings. The task used in these experiments did not require encoding of letter position; the single target letter could pop out from among the background letters. Despite this, differences did emerge across languages, which therefore probably reflect highly automatic processing. This processing must be related to word recognition because it is sensitive to reading direction.

Stevens and Grainger performed an experiment that used the target-letter-in-X's task for five- and seven-letter strings, where fixation location was systematically varied across all string positions [Ste03]. Although the average recognition probability across the string was symmetric with respect to visual field, the Position x Location curves showed an asymmetry. An external letter in the LVF (which was necessarily an initial letter) was better perceived than an external letter in the RVF (which was necessarily a final letter). Internal letters were better perceived at -1 and -2 than at 1 and 2, respectively. Thus like [Naz04a], there was an LVF advantage when the string straddled both visual fields.

An earlier study investigated the interaction of retinal location and string position for a wider range of locations and positions [Wol74]. The stimuli were nine-letter consonant strings presented for 200 ms, and the task was to report as many letters as possible in a left-to-right report order. The location of the first letter of the string was systematically varied from -12 to 5. This yielded separate

retinal location curves for all string positions, and separate string position curves for retinal locations -4 to 4.

An analysis of the data showed a significant interaction of string position with visual field. That is, for a given string position and distance from fixation, the result varied with visual field. These experimental data are displayed in Figure 4.1, in terms of perceptibility at a given retinal location as position varies. To summarize, in the LVF, perceptibility initially drops off quickly with increasing string position, and then levels off. However, in the RVF, perceptibility is more slowly and smoothly decreasing. For example, when a letter at -3 is in the first position, accuracy is 100%, but accuracy is only 35% when that letter is in the third position. Perceptibility decreases to 20% for position 4, but stays roughly constant as position increases from 4 to 7. In contrast, at the analogous location in the RVF/LH (3), perceptibility drops from 95% for position 1 to 55% for position 3; a smaller drop than in the LVF/RH. Perceptibility drops to 30% for position 4, and continues to decrease to 5% for position 7 (rather than stabilizing as in the LVF/RH). Thus the effect of increasing the number of letters to the left of a given eccentricity varies with VF.

### 4.2.3 Unilateral Presentation

This positional interaction with visual field is also in evidence for short strings presented within a single VF. In one study, four-letter strings were embedded in a masking array [Est76]. The array was comprised of 9 \$'s, an x, and 9 more \$'s, and was presented so the x appeared at fixation. A consonant string replaced the \$'s in one of four locations: -8::-5, -5::-2, 2::5, or 5::8. Thus a letter at -5 or 5 could be either an initial or final letter. Exposure duration was either 150 ms or 2400 ms. In the longer duration, eye position was monitored, and the trial was

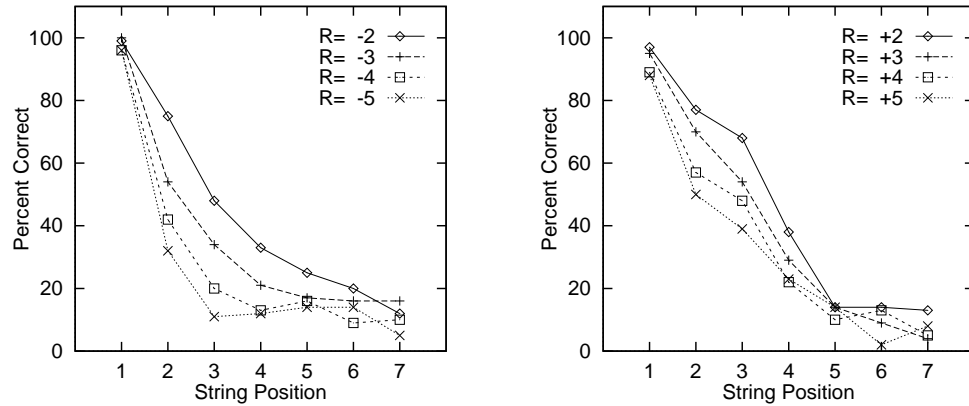


Figure 4.1: Results from [Wol74], with LVF/RH on left and RVF/LH on right. Each line represents a fixed retinal location. As string position is increased (i.e., more letters occur to the left), performance decreases. The pattern of decrease varies with visual field.

terminated if fixation strayed more than  $0.33^\circ$  from the central fixation marker. For both durations, the data displayed similar patterns (see Figure 4.2). For  $-5::2$  and  $2::5$ , performance on the external letters was better than the internal letters. This could not have been a result of a lack of lateral masking because the external letters were always surrounded by \$'s. For  $-8::5$ , accuracy decreased with string position. For  $5::8$ , accuracy was flat across the first three positions, and rose for the final position. Thus, for the larger eccentricity, the letters farthest from fixation were the best perceived in both VFs. This striking pattern has also been observed in studies in which strings were not presented within a masking array [Bou73, Leg01].

This asymmetry between position and VF was also apparent within a single location. At  $-5$ , accuracy was much higher when it was an initial letter. At  $5$ , accuracy was much higher when it was a final letter. There seems to be a general advantage for initial letters in the LVF and final letters in the RVF. These patterns

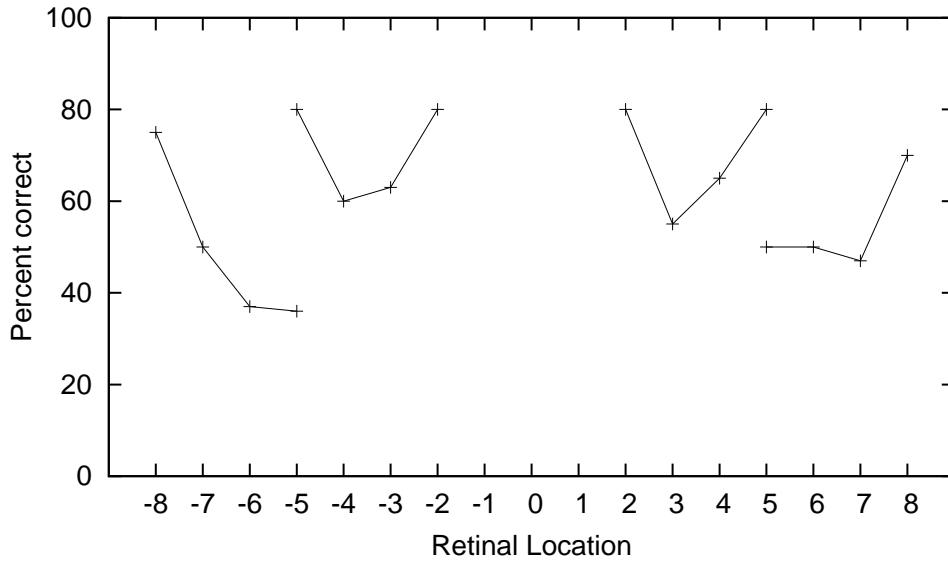


Figure 4.2: Results from [Est76], for the 2400 ms exposure duration.

must result from how a letter string is encoded; they are considerably different from what would be expected on the basis of acuity. It cannot be the case that this occurs simply because initial letters fall in the LVF and final letters in the RVF for fixated words, because short words are often processed without being directly fixated, in which case the first letter falls in the RVF.

However, Jordan and colleagues have claimed that there is no such asymmetry [Jor03]. Using stringent fixation control, they got the same deep, U-shaped patterns (i.e., better accuracy at the external letters) for various locations in both visual fields. Without fixation control, they got a pattern in line with the above experiment: decreasing accuracy with increasing position in the LVF, and shallow U-shaped patterns in the RVF. Therefore, they claim that any observed asymmetry is an artifact of erroneous fixations falling outside the required fixation point. However, this does not explain the observed asymmetry. If there is no asymmetry, the same pattern should emerge no matter where fixation falls. Note



that the above study [Est76] controlled fixation, and yet yielded strong positional asymmetry with VF even when the subjects had 2.4 seconds to examine the stimulus. However, the tolerance in the Jordan experiments was much smaller. In order for the stimulus to be initially displayed, fixation could not deviate from center by more than  $0.125^\circ$  for 1 second. Movements this small are on the order of microsaccades, which occur reflexively at intervals of less than a second in order to keep an object stabilized on the retinas. As Nazir [Naz03] points out, maintaining fixation under this constraint is a demanding task, requiring focused attention prior to presentation of the stimuli. Therefore, performance of this additional fixation task is likely the source of the differing results, rather than uncontrolled fixation errors.

In other studies, consonant-vowel-consonant trigrams were presented vertically [Hel95, Hle97]. For LVF presentation, subjects made many more errors involving the last letter than the first letter of the string. For RVF presentation, this finding was greatly attenuated: there were more errors on the first letter, and fewer errors on the last letter (relative to LVF presentation), resulting in a more even distribution of errors across the string. Correct recognition of the entire string was better in the RVF/LH than the LVF/RH. These patterns were taken to be additional evidence of parallel processing of strings by specialized linguistic modules in the LH, and less efficient, serial processing in the RH.

However, a counterintuitive result arose when the input was directed to both hemispheres simultaneously. Under bilateral presentation, the error pattern is more similar to the LVF/RH than the RVF/LH pattern [Hle97]. Thus, even though the LH was more effective than the RH at performing the task, the RH's mode of processing (i.e., error pattern) dominated when the stimuli were presented to both hemispheres simultaneously. Under a dual processing modes ac-

count, it is unclear why this should be the case.

Similar experiments in languages read in other directions have also cast doubt on a dual modes account of these data. For Hebrew readers, the pattern reversed [Evi99]. Final letter errors were more likely in the RVF/LH than the LVF/RH, and the bilateral pattern was the same as the RVF/LH. A study of Japanese kana, for which the vertical orientation is normal, showed no differences between LVF/RH, RVF/LH, and bilateral presentation patterns [Hell99]. Thus, the patterns vary with reading direction, indicating that they are not a result of hemispheric dominance.

#### 4.2.4 Summary

Priming experiments have shown that facilitation at the letter level can occur across string positions. By using a temporally varying stimulus, it was demonstrated that letters are processed from left to right (in English speakers).

We have seen the letter perceptibility varies in a way that is contrary to acuity. Under central fixation, there is a external letter advantage, where perceptibility decreases from left to right, and rises for the final letter. (This rise at the final letter contrasts with the lack of a final-letter advantage observed in perceptual word-recognition tasks.) Under LVF presentation, the initial letter is perceived the best. Under RVF presentation, the final letter is perceived the best. Letter perceptibility patterns are also sensitive to reading direction. These factors indicate that letter tasks are influenced by processing specific to visual word recognition. Thus a comprehensive model of letter-position encoding should explain these effects.

## Chapter 5

### Models of LPE

In this chapter, I first summarize the desired properties of an LPE model, as constrained by the neurobiological and behavioral data reviewed in the previous two chapters. I then give a brief overview of how artificial neural networks are usually implemented. In the next section, I present other researchers' models, which are all implemented models. These models are evaluated with respect to how well they fulfill the desired properties.

#### 5.1 Desiderata for an LPE Model

- *Neurobiological plausibility.* The lowest level of the model should represent V1, while the highest level should represent the lexical level. The lowest level should be characterized by a retinotopic mapping, where activation levels correspond to visual acuity. At the lexical level, the word representation corresponding to the input should become the most active. Transformations within and between levels should be involve functionality that could be carried out by real groups of neurons.
- *Lowest level is split across hemispheres.* Available evidence indicates that the fovea is not bilaterally represented. Therefore the lowest level of the

model should incorporate a split representation of the input, and the model should integrate this information into a unified LPE.

- *Convert a retinotopic encoding to location-invariant encoding.* The model should solve the problem of how a location-specific representation at the lowest level is transformed into a location-invariant LPE.
- *Incorporate position-independent letter units.* The observed cross-position priming for non-word strings indicates that letter units exist which can encode a letter in any position. This is consistent with the behavior of letter-position dyslexics, who get the identities of letters correct, but not their positions. The model should explain how a positional information is dynamically bound to such units.
- *Explain basic positional patterns.* Under central presentation, letter perceptibility generally decreases from the beginning of the string to the end of the string, but rises for the final letter. The model should explain why this pattern differs from the acuity pattern, and why the final-letter advantage is not evident for degraded word recognition.
- *Explain relative-position and transposition priming.* The highest prelexical representation should be consistent with the fact that word-level priming only occurs when the target's letter order is preserved for the most part in the prime. This priming does not depend on absolute string position.
- *Explain evidence for serial processing.* Tasks in which two strings are briefly sequentially presented show that the beginning of the first string and the end of the second string are processed. This shows a serial read out of the visual image.

- *Explain visual field differences.* Positional patterns vary with VF, reading direction, and hemispheric dominance. In particular, there is an initial-letter primacy in the LVF, and a final-letter primacy in the RVF.

## 5.2 Review of Modeling Basics

A neuron (or group of neurons) is modeled as a *node* which receives activation on incoming connections from other nodes, and sends activation on its outgoing connections to other nodes. Activation is integer-valued or real-valued. Nodes are usually grouped into layers. Each node in a layer has the same pattern of connections to nodes within that layer (lateral connections), and nodes in other layers.

Activations at the lowest layer of the model (input layer) are set by the modeler. Otherwise, the activation is a (often non-linear) function of the input reaching the node on its incoming connections. Associated with each connection is a weight, which models the efficacy of the synaptic transmission between the two nodes. The amount of input arriving along a connection is usually taken as the product of the activation of the sending node and the connection weight. Thus a lower connection weight allows less transfer of activation. The input to a node is the sum of these individual inputs. Weights can be set by the modeler, or developed via a learning algorithm. A node only carries out local computations, limited to functions of its internal state and the incoming activations. Such functions operate on numbers, not abstract symbols. See Figure 5.1.

The connection weights into a node can be considered a vector; the activations of sending nodes can also be considered a vector. The input to a node is then the dot-product of the weight vector and the activation vector. For activation vectors

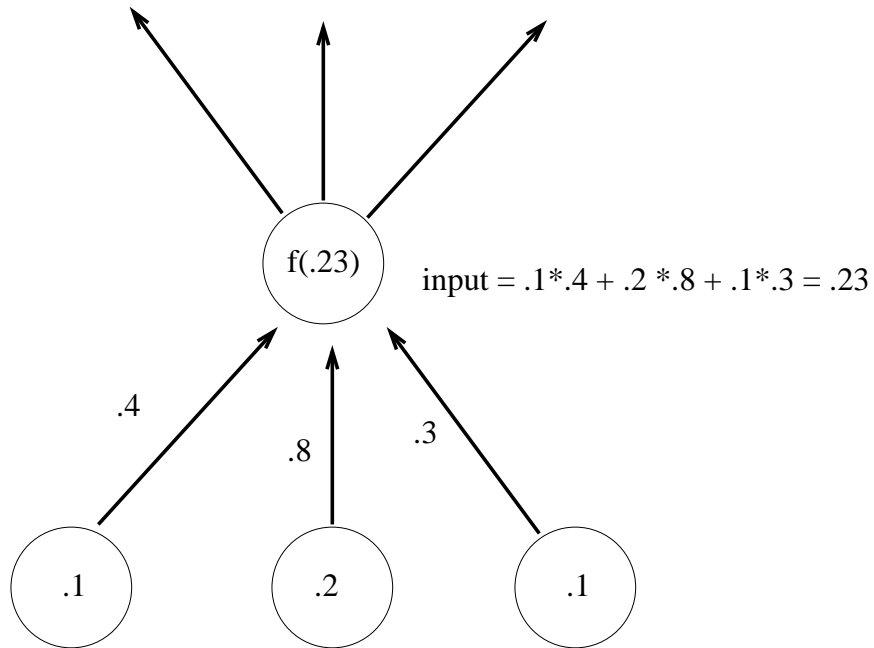


Figure 5.1: Basic components of an implemented model. Each node has an activation value (shown in the center of the node). At the lowest level of the model, activation values are clamped to particular values. Each connection has an associated weight. The input to a node is the dot-product of the activation vector and the weight vector. The activation of a node is a function of this input. A node sends its activation along outgoing connections.

of fixed Euclidean length, the dot-product is maximized when the activation vector is parallel to the weight vector (i.e., the angle between the two vectors is 0). Thus activation level generally reflects how closely the incoming activation vector is aligned with the node's weight vector.

Activation level can model different aspects of neural function. One possibility is for the activation value to represent membrane potential. In this cause, outgoing connections carry information about individual neural spikes (or sets of spikes if the node represents a set of neurons with the same dynamics). This requires simulation at millisecond time scale. The alternative is that the activation value represents firing rate, or the total number of spikes over some time period. This is a more abstract level of modeling.

There are three basic types of learning algorithms used to modify connection weights: supervised, reinforcement, and unsupervised learning. In all cases, the input units are clamped, and activation flows through network to the output layer. In supervised learning, the activations on the output layer are compared to the target activations desired by the modeler (for that particular input). The differences between the actual and target activations (the errors) are used to modify connection weights so that the errors decrease in magnitude. The most well-known algorithm of this type is back-propagation. In this way, the network learns to perform a particular task (i.e. an input-output mapping).

In reinforcement learning, the modeler provides feedback that is limited to a reward signal; this is essentially trial-and-error learning. Thus there is some randomness associated with the activation functions. When an output pattern receives a positive reward, connection weights are modified to increase the probability of repeating that output pattern. When a negative reward is generated, connection weights are changed to decrease the probability of repeating that out-

put pattern, and to increase the probability of creating a different output pattern (which could potentially generate a good response).

In unsupervised learning, there is no feedback from the modeler. Rather, the goal is for the network to self-organize to reflect regularities in the input. One example of this approach is Hebbian learning, where connection weights are strengthened between two nodes that are both activated at the same time (or at successive time steps).

## **5.3 Models of LPE**

### **5.3.1 Interactive Activation Model**

The most well-known model of visual word recognition, the Interactive Activation (IA) model [McC81], used a position-specific encoding. That is, there was a separate node representing each letter in each position. Letter nodes connected forward to nodes representing individual words, and these word nodes connected back to letter nodes. Connection weights were binary, with a weight of 1 if the letter was in that position in the word, and 0 otherwise. The model was based on four-letter words, so differing lengths were not an issue. The primary goal of this model was to illustrate the effects of top-down activation from the word level back to the letter level. The position-specific encoding was likely implemented as an expediency, rather than as an instantiation of a theoretical model of LPE. As we have seen, such a position-specific encoding is not compatible with behavioral studies.



### 5.3.2 Print-to-Sound Models Trained by Back-Propagation.

In a model of learning to read aloud [Sei89], orthographic input was represented by letter trigrams. (An example of a trigram encoding is given in section 2.1). Orthographic units were fully interconnected with a hidden layer, which was fully interconnected with the output layer representing phonemes. The connection weights were learned via the back propagation algorithm. The goal of the modeling was to show that a single network could learn to pronounce both regular and exception words. (An exception word is a word that does not follow the usual rules of pronunciation, such as *pint*.) However, this model produced poor generalization on reading pseudowords (pronounceable letter strings that are not words). This was likely due the choice of input representation; representations of the same letter in different contexts bore no relationship to each other, making generalization difficult. This is known as the *dispersion problem*. Changing the input encoding to a slot-based one which encoded letters and graphemes in onset, vowel and coda positions allowed better performance on pseudoword reading [Plsa93]. Again, the focus of these models was not LPE, and the choice of orthographic encoding was merely a means to an end.

Nevertheless, it is instructive to evaluate trigrams as a potential prelexical encoding. They do not rely on absolute-position encoding, consistent with experimental evidence. However, they do not offer sufficient flexibility to account for the relative-position results. Consider the prime 1346 and the target word 123456. The only trigrams it shares with the target are *\*\*1* and *6\*\**, so there is no basis for an influence of the order of the interior letters.

The difficulties in achieving good pseudoword performance using trigrams points to an important issue. LPE not only serves a direct, orthographic route

to lexical semantics, it also provides input to a process which learns to map orthography to phonology. These results demonstrate that context units, such as trigrams, make that task difficult due to the dispersion problem. This suggests that an encoding based on position-independent letter units should exist at some level of processing, in order to provide dispersion-free input to the phonological system. This is consistent with the above experimental evidence for such representational units [Per95, Fri01]. However, such an encoding cannot readily explain the relative-position priming data [Hum90, Per99, Per03], suggesting that position-independent letter units may not directly contact the lexical level.

### 5.3.3 BLIRNET

Unlike the preceding models, BLIRNET [Moz91] specifically focused on the issue of how a retinotopic representation of a letter string could be transformed into a location-invariant LPE. At layer 1, the retina was represented as a 6 x 36 array of nodes, with each location comprised of 5 nodes which represented different features (corresponding to four different line orientations, and a node encoding whether the line terminated in that location). Each letter was represented by these features over a 3 x 3 region. Above this layer were 5 more layers in which the size of the array progressively decreased, but the number of features at each location increased. For example, at layer 2, the array was 3 x 12, with 45 different types of feature nodes at each location. Each node in layer 2 received inputs in a topographic manner from a 4 x 6 region of layer 1. By layer 6, the array was reduced to 1 x 1, with 720 different features; thus, this layer did not encode any information about stimulus location.

The features detected by these higher layers were hard-coded in the connection weights. Weights were equal to each other on all connections joining a given

feature type at layer  $n - 1$  to a given feature type at layer  $n$ . This provided a degree of location invariance. The connection weight between two feature types was randomly chosen (under some distributional constraints). Thus the detected “features” did not correspond to psychologically motivated units.

The resulting pattern in layer 6 was mapped to a trigram representation via supervised learning. Trigrams also included units of the form A\_BC, meaning that a single letter occurred between A and BC. Because the total number of possible trigrams is huge, the 540 most common were chosen. Thus there was a trigram layer of 540 units above layer 6, with initially random connection weights. The network was trained by encoding various words at various locations in layer 1. For each trial, the resulting trigram activation pattern was compared to the desired trigram activation pattern for that word, and the weights were modified to bring the actual activation pattern closer to the desired pattern.

At the end of training, the trigram layer produced a noisy, somewhat location-invariant representation of the letters encoded on layer 1. This representation was cleaned up via lateral excitation between consistent trigrams (e.g., BON and B\_ND) and lateral inhibition between inconsistent trigrams (e.g., BON and OBN), while unrelated trigrams (e.g., BON and DER) did not affect each other. The trigram layer was connected to a lexical layer with top-down connections back to the trigram layer. This feedback also helped to clean up the trigram representations. This allowed translation-invariant recognition of words encoded in layer 1. Thus, at layer 6, there was sufficient information to reliably extract trigram identities when additional consistency and lexical information were added.

While this model addresses the difficult problem of location-invariance, there are some difficulties with the proposed solution. The only way to represent letter position in this model is via a contextual encoding, since all spatial informa-

tion is factored out. Thus a model that achieves location invariance in this way fundamentally rules out a position-independent, letter-based representation. It could learn to detect individual letters, but could not represent their positions independent of their context in a general way. The model did include some trigrams corresponding to positional letter detectors, of the forms  $**A$ ,  $**_A$ ,  $A_**$ , and  $A**$  (encoding first, second, next-to-last, and last letters). However, this representation cannot be extended to the interior positions. This is in conflict with the above arguments for the existence of a letter-based representation and for position-independent letter units. Furthermore, this location invariance is achieved via an unrealistic jump in receptive-field size; the representation at the first layer corresponds to features, while the representation at the second layer corresponds to adjacent letter pairs.

Nevertheless, it is instructive to evaluate the proposed trigram encoding. It includes “\_” and therefore offers a more flexible encoding than one that only includes contiguous trigrams. Under this scheme, 1346 shares trigrams  $*1_3$ ,  $1_34$ ,  $34_6$ , and  $4_6*$  with the target 123456, while 1436 does not share these trigrams. This is consistent with relative-position priming results. However, the representation is still not flexible enough. Consider the stimuli 12dd56 and 124356 for the target 123456. The two stimuli share an equal number of trigrams with the target, which is inconsistent with transposition-priming results.

### 5.3.4 A Split Fovea Model Trained by Back-Propagation

The goal of this model is to explain visual field differences based on a split cortical representation of the fovea [Mon04]. The input was in the form of location-specific letter units, with four locations in each “visual field”, corresponding to foveal vision. The left four units connected directly to one bank of hidden units

representing the RH, while the right four units connected directly to a different bank representing the LH. The two banks of hidden units were fully interconnected with each other. The hidden units connected to an output layer representing a lexical encoding, which could be phonological or semantic depending on the simulation. The network was trained on four-letter words. The size of the input layer allowed a four-letter word to occur in five different locations. The network was trained via back-propagation by representing each word at all five possible locations on the input layer. Thus the network learned to produce a location-invariant representation at the output layer.

Measurements of performance based on mean squared error showed similar patterns to some human phenomena, such as the location of the Optimal Viewing Position, positional effects of grapheme-phoneme irregularities, and VF differences in semantic processing. These results are explained as arising from an interaction between positional letter statistics and the distribution of positions with a “visual field”. In English, there is more variation in letter identities at the beginning of words than at the ends of words. However, when contiguous letter pairs are considered, there is more variation at the ends of words than at the beginnings. Under the model’s input conditions, initial letters fell more frequently in the “LVF” and final letters in the “RVF”. For example, of the five input locations, only one yields the first letter in the “RVF”. Thus the “RH” becomes tuned to the statistics of beginnings of words, while the “LH” becomes tuned to the ends of words. Because of the differing information granularities within words, coarseness of representation varies with “hemisphere”. This leads to the positional and VF effects in the model.

This model has the advantage that it deals with the issue of a split representation of the fovea. However, I suggest that the underlying assumptions are

unrealistic. The model's results arise because the hidden-layer representation of a string varies with its location on the input layer. However, this means that there is no location-invariant prelexical representation. For example, there is no abstract encoding that the letters *cart* spells the word CART. Rather, the relationship between the letters *cart* and the word CART has to be relearned for every possible stimulus location.<sup>1</sup> This is inefficient and contrary to all other models and theories of LPE, which assume that an abstract representation of letter order contacts the lexicon. This is also inconsistent with imaging evidence that processing becomes left-lateralized at a prelexical level [Coh00, Deh04].

Furthermore, there are also problems with the assumption underlying the positional frequencies. The relationship between letter position and visual fields arose from the use of symmetric visual fields in the model. However, to achieve normal speed in text reading, 4 LVF letters and 12 RVF letters must be visible [Ray75]. Thus, the visual fields are effectively highly asymmetric in reading. Short words falling in the right parafovea are frequently processed without being directly fixated, about 50% of the time for four-letter words [Ray76]. Thus it is not actually the case that the initial letters of a word are considerably more likely to fall in the LVF than the RVF.

### 5.3.5 SOLAR

The SOLAR model was developed to illustrate how a word recognition module could self-organize to recognize strings of varying lengths [Dav99]. It is based on

---

<sup>1</sup>This is only strictly true when the output is not predictable from the input, such as for a semantic encoding. The model could potentially give the correct phonological response for a regular word at untrained location, based on generalization from other words presented at that location.

the SONNET model [Nig93]. The highest prelexical encoding employed position-independent letter units, where position was represented by an activation gradient. That is, a letter unit could represent that letter in any position, and its activation level encoded position. Activation decreased from left to right. For example, to represent the input CART, letter node C would have the highest activation, A the next highest, R the next and T the lowest. (Multiple instances of a letter were encoded by different instances of a letter node). This activation pattern was taken to arise from a serial read out of the visual image, i.e., letter node C is activated, then A, then R, then T. Earlier letters accumulate higher activation values because they fire longer. Therefore, the final letter of a string is taken to have a set minimum activation level, and activation increases as position decreases. The serial read out was taken to correspond to a covert attentional scan of the visual image.

After learning, connection weights into a word node became equal to the letter activation pattern resulting from that word. However, a simple activation function based on the dot-product of the weight and activation vectors causes difficulties. For example, the input *car* would activate CART more than CAR, since the activations of C, A, and R are higher for *cart* than for *car*, and these differences are learned on the connection weights. One approach to solving this problem is to normalize the length of the activation vector to 1. This is accomplished by squaring each activation, summing these squares, and dividing each activation by the square root of this sum. Thus the activations for C, A, and R for *cart* become smaller than for *car* because the activations for CART are divided by a larger quantity (because it includes the square of the activation for T). These differences are reflected on the respective learned connection weights. Therefore the input *car* activates CAR more than CART. However, this approach

becomes less and less effective for longer words, because the weights on large positions become quite small, and they do not have much influence. Therefore this solution is not robust in the presence of noise because differences in activation values at the word level can be quite small. For example, taking the ratio between positional activations to be 1.2, an input of the form 1234 activates a word node encoding 12345 to 0.96 (versus 1.0 for word node 1234). Another problem is that this encoding of position is not robust, especially toward the ends of words where activations and weights are low. For example, the sequence 12354 would activate word node 12345 to 0.99.

To solve these problems, another component,  $I$ , contributed to the bottom-up input in the SONNET and SOLAR models.  $I$  operates on normalized activation and weight vectors, as follows. For word node  $W$ , the ratio of each letter unit's activation and its connection weight was taken. This ratio was capped at a maximum of 1.0 for each letter. Each ratio was multiplied by a constant  $K$ , and the result was added to  $2 - K$ .  $I_W$  was set to the product of these values, giving a maximum possible value of  $2^L$ , where  $L$  is the number of letters having non-zero connection weights into  $W$ .  $I_W$  was multiplied with the dot-product of the activation and weight vectors to comprise the bottom-up input to word node  $W$ .

Thus this new component compared the weight and activation vectors directly to each other, producing a penalty at letters where the activation value was less than the connection weight. For example, taking  $K = 1.5$ , input of the form 1234 gives  $I = 16$  for word node 1234, and  $I = 8$  for word node 12345, since the ratio of 5's activation to its weight is 0, giving a value of 0.5 for that connection. Thus the shorter word has an advantage. Input of the form 12345 gives  $I = 32$  for word node 12345, while 12354 gives  $I = 28$ , which is 87.5% of the maximal value possible value of  $I$ , as opposed to 99% of the maximal possible value for



the dot-product. Thus misorderings are amplified.

This model has the advantages that it is consistent with the evidence for serial read out and position-independent letter detectors. However, there are difficulties concerning the way that letter nodes activate the lexical level. While the computation of  $I$  does indeed solve the above problems, it is not biologically plausible. It is unclear how an activation level and a weight could be directly compared. A weight reflects properties of synaptic transmission. It modulates the efficiency of the interaction between nodes. This value cannot be extracted from synapses and used in other calculations. Thus,  $I$  is a computational convenience and does not illuminate how ordering information is actually compared in the brain. Moreover, it does not give the correct results in some cases. Consider the inputs 1346 and 1436 for the word node 123456. For all positions in both of these stimuli, the letter activation is higher than the weight; thus all the ratios are maxed out at 1.0, giving  $I = 16$  for both stimuli. Therefore, this measure is insensitive to ordering for a stimulus that is shorter than the target, contrary to experimental evidence.

Another problem is the proposed activation pattern across the letters. The final letter has the lowest activation level. This is inconsistent with the broad range of evidence for a final letter advantage. While it is argued that increased performance could arise at the last letter due to a recency effect from the serial activation of letters, this implies that activation level and performance are independent of each other. It is generally assumed that performance reflects activation level. That is, a recency effect occurs because the activation of the final letter remains higher than previous letters due to less decay. This is inconsistent with the assumption that the final letter has the lowest activation level.

### 5.3.6 LEX

In this model position-independent letter units connected to word units, which included a phonological encoding of the word [Kwa99a]. Letter order was represented serially. The first letter fires and activates all words having that first letter. The second letter fires and further activates only those words matching on that letter in that position, and the more highly activated words inhibit those words that do not match. This process continues for each letter until there is only one active word remaining.

After each letter fires, the model generates a phonological output which is a combination of the pronunciations of all active words. The goal of the model was to demonstrate that this type of processing could account for naming and lexical-decision phenomena in the absence of learned grapheme-phoneme mapping rules.

For the present purpose, the serial encoding of letter order is of the most interest. However, no details are given as to how the serial input is matched to the stored lexical encoding. That is, how does the  $n$ th letter only further activate those words matching that letter in the  $n$ th position, without explicitly encoding letter positions within a word node? It seems unlikely that this match was implemented in a biologically plausible manner (i.e. based on numerical computations on weighted connections). However, even if it were, these activation dynamics are problematic, because they are equivalent to a position-specific encoding. That is, although the letter units are position-independent and dynamically represent position by firing order, they activate words in a position-specific way. This is incompatible with relative-position priming.

## 5.4 Summary

I conclude this section by reviewing how the requirements of an LPE model are satisfied (or not) by the above models.

*Neurobiological Plausibility:* The LEX model did not specify the activation function for the word level. The SOLAR model used an unrealistic one in which activations and weights were directly compared. Other models were plausible in that they relied on standard activation functions.

*Split fovea, and Visual Field Differences:* The only model that addressed these issues was the Split fovea model. However, the split representation was not integrated into an location-invariant LPE, and the VF differences arose from unrealistic assumptions.

*Retinotopic to Location-Invariant Encoding:* Only the Split Fovea and BLIR-NET models addressed this problem. Neither model incorporated position-independent letter units, which is in conflict with the next requirement.

*Position-Independent Letter Units and Serial Encoding:* The SOLAR and LEX models included such letter units under a serial readout of the visual image. In the SOLAR model, position was dynamically represented by activation level, and activation level was driven by serial processing. However, the word-level activation function was neurobiologically implausible. In the LEX model, order was represented temporally. However, it is unclear exactly how this encoding activated words, and the proposed activation dynamics were position-specific. Thus neither model really solves the problem of how the letter-based, temporal encoding could be decoded at the word level.

*Relative and Transposition Priming:* The SOLAR model came the closest to fulfilling these requirements. However, this achievement is based on an implau-

sible word-activation function, which doesn't give the proper relative-position results for primes and targets of differing lengths. As for contextual units, the trigrams used in the present models (print-to-sound1 and BLIRNET) are not sufficiently flexible. The IA, print-to-sound2, and LEX models activated words in a position-specific way. It is unlikely that the location-specific encodings developed by the Split Fovea model could replicate these phenomena.

*Serial Processing:* The LEX and SOLAR models include a serial readout of letters.

*Positional Patterns:* SOLAR is the only model producing varying letter activation levels, which arise from serial activation of letter nodes. The proposed activation gradient is consistent with behavioral evidence for non-final positions, but not with the final-letter advantage. Furthermore, the activation gradient does not explain the interaction of perceptibility patterns with visual field under unilateral presentation.

In the following chapter, I present the SERIOL model, which satisfies all of these requirements.

## Chapter 6

### The SERIOL Model of LPE

My theoretical model of LPE is dubbed the SERIOL model (Sequential Encoding Regulated by Inputs to Oscillations within Letter units) [Whi01a]. The model is best motivated in a top-down manner. I first give an overview, starting at the word level and working down. I then specify the model in more detail, which is best done in a bottom-up manner.

#### 6.1 Overview

##### 6.1.1 Highest Prelexical Orthographic Representation

The relative-position and transposition priming results [Hum90, Per99, Per03, Gra04a] place strong constraints on the nature of the highest prelexical representation [Gra04b]. Contextual units are a natural type of unit to represent order in the non-position-specific manner that seems to be required. As we have seen, trigram units do not offer sufficient flexibility. Adding a wild-card character (“\_” in BLIRNET [Moz91]) brings the representation closer to explaining the priming results. Maximum flexibility is achieved by using bigrams instead of trigrams, and allowing letters to occur between the two letters of the bigram. Following my original specification of such bigram units [Whi99, Whi01a], Grainger later

also endorsed such units, dubbing them *open bigrams* [Gra04b, Sch04].

Bigram activation level is a decreasing function of the distance between two letters. Thus, bigrams triggered by contiguous letters are more highly activated than those representing separated letters. This leads naturally to a maximal allowable separation. Priming data suggest 2 is the maximum [Sch04]. A new assumption is that the external letters are anchored by edge bigrams<sup>1</sup>. For example, the stimulus *chart* activates bigrams \*C, CH, HA, AR, RT, T\* and CA, HR, AT and CR, HT, where the first group of bigrams has the highest activation level, the next group has a lower activation level, and the last pair of bigrams has the lowest activation level. Bigrams contact the word level via weighted connections, where the weights are proportional to the bigram activation pattern for each word.

### 6.1.2 Nature of Pre-Bigram representation

So then how are bigrams activated? Consistent with evidence for position-independent letter units [Per95], I assume that such units comprise the next lowest level. This requires that position be dynamically represented. Two possibilities are that position is represented by an activation pattern, or by firing order. As discussed above in section 5.3.5, a monotonically decreasing activation gradient is inconsistent with the final-letter advantage. Therefore, in line with evidence for left-to-right string processing [Har75, Nic76], letter order is taken to be represented serially. Thus in our example, C fires, then A, then R, then T. A bigram is activated when its constituent letters fire in the correct order, and bigram activation level falls off as the interval between letters increases.

---

<sup>1</sup>The original specifications of the model [Whi99, Whi01a] did not include edge bigrams.

Because bigram units intervene between the letter and word levels, this solves the problem of how a temporal letter encoding can activate the word level. (This problem is discussed in section 5.4.) However, recall that the encoding of letter order also subserves a phonological route to the lexicon, as discussed in section 5.3.2. While open bigrams are suitable for lexical access along an orthographic route, they are not suitable for phonological access because they do not provide phonologically meaningful units, thereby introducing the dispersion problem. Therefore, I assume that processing branches after the sequential encoding. Along the orthographic route, letters activate open-bigrams, and then words. Along the phonological route, letters activate phonemes and then words (perhaps via some intermediate syllable-based encoding). The point is that bigrams are not activated along the phonological route. The serial nature of the letter-based encoding maps well onto the serial nature of phonology.

### **6.1.3 Induction of Serial Encoding**

How is this temporal firing pattern induced at the letter level? Hopfield [Hop95], and Lisman and Idiart [Lis95] have proposed related mechanisms for precisely controlling timing of firing. This is accomplished via a node which undergoes sub-threshold oscillations of excitability. For convenience, I designate the trough of the cycle to be the “start” of the cycle. Input level then determines how early in the cycle such a node is able to cross threshold and fire. (See Figure 6.1.) Near the beginning of the cycle, excitability is low, so only a node receiving a high level of input can cross threshold and fire. Excitability increases over time, allowing nodes receiving less and less input to progressively fire. Thus serial firing at the letter level can be accomplished via letter nodes which oscillate in synchrony and take input in the form of an activation gradient. In our example,

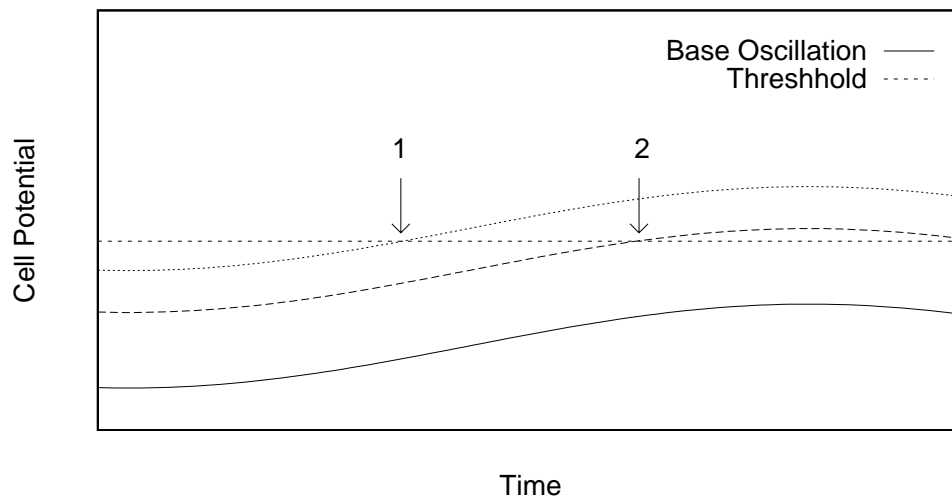


Figure 6.1: Interaction of input level and timing of firing for a cell with undergoing a sub-threshold oscillation of excitability. When a relatively high level of input (top curving line) is added to the base oscillation, the cell crosses threshold at time 1 (action potential not illustrated). If less input were received, the cell would cross threshold later in the cycle, such as at time 2.

C would get the most input, A the next, R the next, and T the least. So C can fire the earliest, A next, R next, and finally T. Note that these letter nodes are not tied to location or position. The same letter node can represent a letter occurring at any position, based on its timing of firing.

Thus there must be an activation gradient across the next lower level of the model, to provide input to the letter level. Because this gradient decreases from left to right, these lower-level units must be tuned to retinal location. I have assumed that the input to the letter level comes from feature units [Whi01a]. However, the assumption of feature units is not crucial. Input to position-independent letter units could just as well come from location-specific *letter* units. The important point is that an activation gradient across units tuned to retinal location interacts with synchronously oscillating letter nodes which are location-



and position-independent; a retinotopic representation is converted into a serial representation. The resulting serial representation is a location-invariant encoding. Thus location invariance is achieved by mapping space onto time. This location-invariant encoding is presumed to occur in the LH. For convenience, I assume that this *locational gradient* occurs across feature units.

The induction of serial firing also results in varying activations at the letter level. Letters that receive more input also fire faster and achieve higher activations. Therefore positional activations at the letter level are similar to those at the feature level, with the exception of the final letter. The final letter is not inhibited by the firing of a subsequent letter; thus it can fire longer than non-final letters. Although the final letter receives a lower level of input than the other letters, it can reach a higher activation level (where activation is based on total number of spikes). This is consistent with the final-letter advantage. See Figure 6.2 for a schematic of the letter through word layers.

#### **6.1.4 Creation of the Locational Gradient**

How is the activation gradient induced at the feature level? Recall that at the lowest level of the model (dubbed the edge layer) there is a different activation pattern, one based on acuity. For a fixated word, the acuity pattern across the letters in the RVF is the same as required for the locational gradient (i.e., decreasing from left to right). Thus the acuity gradient can serve as the locational gradient for those letters. However, in the LVF, the acuity gradient increases from left to right; its slope is in the opposite direction as required for the locational gradient. Therefore, when the edge level activates the feature level, the acuity gradient must be inverted in the LVF/RH, while it can be maintained for the RVF/LH. Details of this processing are presented below. For now, it is sufficient

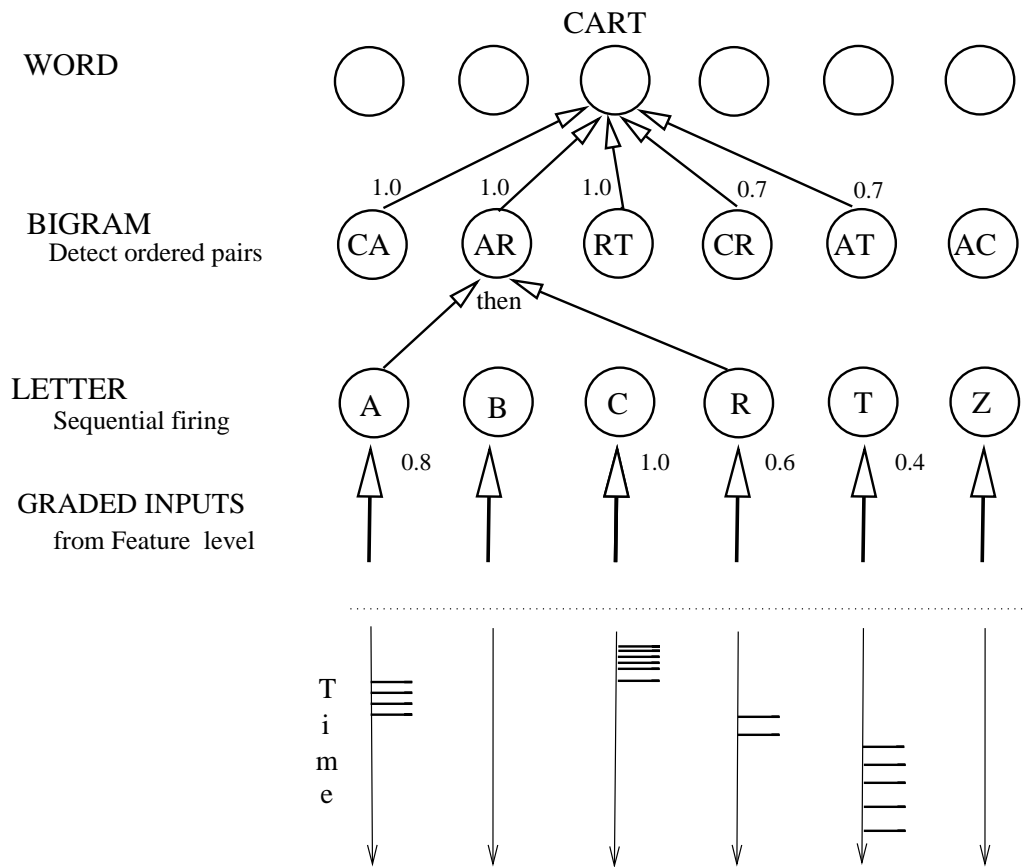


Figure 6.2: Architecture of the letter, bigram, and word levels of the SERIAL model, with example of encoding the word CART. At the letter level, simultaneous graded inputs are converted into serial firing, as indicated by the timing of firing displayed under the letter nodes. Bigram nodes recognize temporally ordered pairs of letters (connections shown for a single bigram). Bigram activations (shown above the nodes) decrease with increasing temporal separation of the constituent letters. Activation of word nodes is based on the conventional dot-product model.

to note that such hemisphere-specific processing could potentially be a source of VF differences.

### 6.1.5 Summary

There are five levels of representation: edge, feature, letter, bigram, and word. The acuity gradient at the edge level is converted via hemisphere-specific processing into a monotonically decreasing locational gradient at the feature level. This gradient interacts with oscillatory letter nodes, yielding serial firing and creating a location-invariant representation. A positional activation pattern also results at the letter level. The letter level feeds into separate orthographic and phonological routes. Along the orthographic route, open-bigram nodes respond to letter pairs that fire in a particular order. Bigram activation depends on the time lag between the firing of the constituent letters. The bigrams contact the lexical level via weighted connections.

I conclude this overview by briefly specifying how this model satisfies the requirements for a LPE model.

*Split fovea and VF differences:* A split fovea is assumed at edge level. Formation of the locational gradient integrates the two halves of the string. Due to the acuity gradient, this requires hemisphere-specific processing, potentially accounting for VF differences.

*Position-independent letter units and serial processing:* Position is dynamically represented by firing order. Activation of words via a bigram layer provides a mechanism for decoding the temporal representation.

*Retinotopic to Location-Invariant encoding:* This is achieved via the interaction of the locational gradient and oscillatory letter nodes.

*Relative-position and transposition priming:* These phenomena are explained

by the open-bigram units.

*Positional patterns:* The locational gradient overrides the acuity pattern. This gradient creates varying activations at the letter level. Lack of inhibition of the final letter by a subsequent letter creates a final-letter advantage.

*Neurobiological plausibility:* Most interactions occur along standard weighted connections. As for the proposed temporal encoding, Lisman and Idiart discuss empirical support for the underlying assumptions [Lis95]. In line with the proposed precision of spike timing, recent studies have shown that single spikes encode significant amounts of information [Rie97], and that spike timing is reproducible at a millisecond time scale [Ber97, Vic96]. In line with the proposed oscillatory cells, slice preparations have shown sub-threshold, theta-band oscillations in cortical pyramidal cells [Fel01, Buc04]. Furthermore, a role for theta-band oscillations has been implicated in visual word recognition [Kli01]. As for the bigram nodes, others have proposed neural mechanisms of how temporally ordered pairs could be recognized, via transition of receptor conformations [Deh87], or activation decay under specific connectivity patterns [Pul03].

## 6.2 SERIOL model

Having given an overview of the model, it is now presented in more detail. As discussed in the Introduction, this a theoretical framework. The model is specified by describing the representation and the activation pattern at each layer, and the transformations between layers.

In the following, the term *activation* denotes the total amount of neural activity induced by a letter (within a given processing layer) over some fixed time period. Thus, activation increases with the number of cells firing, their firing

rate, and the duration of firing (if firing duration is less than the time period being considered).

### 6.2.1 Edge Layer to Feature Layer

At the edge level, the activation pattern results from the acuity gradient. That is, the total amount of neural activity representing a letter decreases as distance from fixation increases. At the feature level, this pattern must be converted into the locational gradient, wherein activation decreases from left to right. Obviously, a high level of activation for the leftmost letter cannot be achieved by increasing the number of cells representing that letter. Rather, the locational gradient is created via modification of firing rates. It is assumed that the following transformations are learned during reading acquisition, most likely in response to a top-down attentional gradient.

Recall that the acuity gradient can serve as the locational gradient in the RVF/LH, but not the LVF/RH. In the LVF/RH, the acuity gradient is inverted as the feature level is activated, via a combination of excitation and lateral inhibition. This process is displayed in Figure 6.3. It is proposed that letter features in the LVF/RH become more highly activated by edge-level inputs than those in the RVF/LH. This allows the first letter to reach a high level of activation. This could occur either via higher bottom-up connection weights from the edge level, or by stronger self-excitatory connections. Within the RH feature level, there is strong left-to-right lateral inhibition. That is, a feature node inhibits nodes to its right. As a result, letter features corresponding to the first letter receive no lateral inhibition, and inhibition increases as letter position increases. Thus, the features comprising the first letter attain the highest activation level (due to strong excitation and lack of lateral inhibition), and activation decreases

toward fixation (due to sharply increasing lateral inhibition, from more and more letters).

In the RVF/LH, the acuity gradient serves as the locational gradient. Overall excitation is weaker than to the LVF/RH. Left-to-right inhibition is not necessary, although some weak such inhibition may steepen the slope of the gradient.

The two hemispheric gradients are “spliced” together via functional cross-hemispheric inhibition. The RH features inhibit the LH features, bringing the activation of the RH features lower than the activation of the least activated LH features. As a result, an activation gradient that is strictly decreasing from left to right is created. This cross-hemispheric inhibition explains the LVF advantage for letter perceptibility in strings that straddle both visual fields [Ste03, Naz04a].

Next I consider the nature of this proposed cross-hemispheric inhibition. One possibility is that RH features directly inhibit LH features across the corpus callosum. Another possibility is that the RH feature-layer representation activates a corresponding feature-level representation in the LH, and that the inhibition occurs within the LH. It is a matter of debate whether callosal connections are primarily excitatory or inhibitory (see [Reg01] for a discussion). Computational models have shown that inhibitory cross-hemispheric connections are required to produce strong hemispheric lateralization, while predominately excitatory connections are necessary to model the reduced neural activity observed contralateral to a cortical lesion [Lev00, Reg01]. A single model demonstrated that inhibition at the sub-cortical level and excitation at the cortical level could account for both phenomena, suggesting that callosal connections may be predominately excitatory [Reg01]. However, even if callosal connections are predominately excitatory, the existence of inhibitory connections is not ruled out. Indeed, in the cat, stimulation of transcallosal neurons resulted in both excitatory and inhibitory post-synaptic

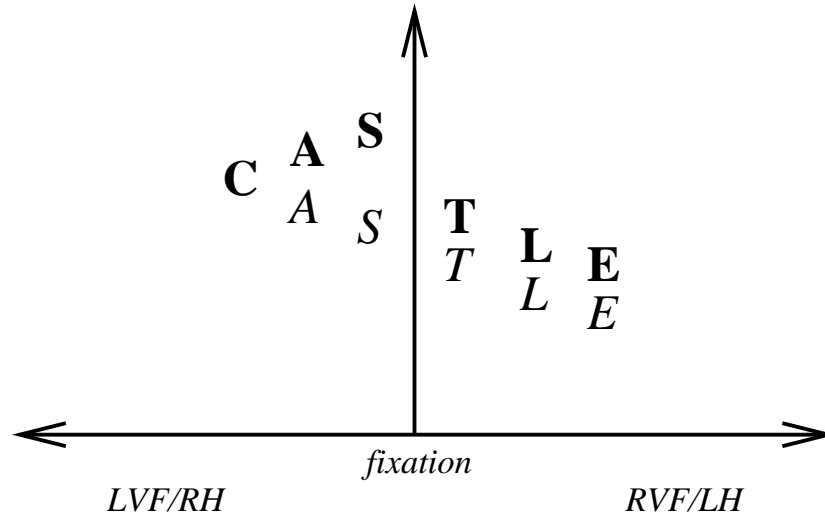


Figure 6.3: Formation of the locational gradient at the feature layer, for the centrally fixated stimulus CASTLE. The horizontal axis represents retinal location, while the vertical axis represents activation level. The bold-face letters represent bottom-up input levels, which are higher in the RH than the LH. In each hemisphere, activation decreases as eccentricity increases, due to the acuity gradient. The italicized letters represent the effect left-to-right inhibition within the RH, and RH-to-LH inhibition. In the RH, C inhibits A, and C and A inhibition S, creating a decreasing gradient. The RH inhibits each letter in the LH by the same amount, bringing the activation of T lower than that of S. As a result, activation monotonically decreases from left to right.

potentials in contralateral receptive cells [Cis03]. It might be possible to selectively strengthen such inhibitory connections, allowing unidirectional, transcallosal inhibition.

I do assume that callosal transfer of the RH information to the LH occurs prior to the letter level, which is taken to correspond to the LH's VWFA. So the RH features may transcallosally inhibit the LH features and excite the LH letter representations. Alternatively, RH features may activate feature-level "copies" within the LH, and such inhibition and excitation would then occur entirely within the LH. Thus, I leave as open questions the nature of callosal transfer and the substrate of the proposed cross-hemispheric inhibition.

### 6.2.2 Feature Layer to Letter Layer

The locational gradient of the feature level induces a temporal firing pattern across letter nodes wherein position is represented by the precise timing of firing relative to other letter nodes. All letter nodes are assumed to undergo synchronous, periodic oscillations of excitability. Following Lisman and Idiart [Lis95], this oscillation is taken to fall in the theta range (5 - 8 Hz; cycle length = 125 to 200 ms). Due to the locational gradient, letter nodes fire serially. An activated letter node inhibits other letter nodes. As a letter node continues to fire, its firing rate slows, reducing lateral inhibition to the other nodes. This allows a new letter node to start firing. When an active letter node receives lateral inhibition, it then becomes strongly inhibited, so that it will not re-fire for the remainder of the oscillatory cycle.<sup>2</sup> Thus the graded input levels and lateral inhibition create

---

<sup>2</sup>This raises the question of how repeated letters are handled. I assume that there are multiple copies of each letter node, and a different node becomes activated for each instance.



serial firing at the letter level.

This process also creates varying activations at the letter level. I assume that a higher input level leads to faster firing.<sup>3</sup> The activation of a letter node depends on both its firing rate and duration. Firing duration is determined by when the next letter starts to fire, which is determined by the input level to that node. Thus the activation of a letter depends both on its own input level, and the input level to the next letter. Assuming a fairly constant firing duration across letters, this gives a decreasing activation gradient at the letter level. The firing duration of each letter is taken to be on the order of 10 - 20 ms. However, the final letter is not inhibited by a subsequent letter. It can continue fire until the end (down-phase) of the oscillatory cycle.<sup>4</sup> Therefore, the final letter could potentially fire longer than the other letters, and reach a higher level of activation than the internal letters even though it receives less input.

### 6.2.3 Letter Layer to Bigram Layer

A bigram node XY becomes activated when letter node X fires, and then letter node Y fires within a certain time period. Thus letter node X primes or gates node XY, allowing it to fire when input from letter node Y is received. If the node is not initially primed by input from letter X, it cannot fire. A bigram

---

<sup>3</sup>This is consistent with experimental results in which a hippocampal CA1 neuron was driven by oscillatory current injected into the cell body, coupled with stimulation to the dendrites. Increasing the amplitude of the dendritic current caused the cell to fire earlier with respect to the somatic oscillatory cycle, and to fire more quickly, generating more action potentials [Mag01].

<sup>4</sup>This assumes that a single word is being processed, as in experimental studies. Under natural reading conditions, multiple short words could be represented in a single oscillatory cycle.

node responds with a burst of firing, and then is quiet. The number of spikes in this burst decreases as the time increases between the firing of X and Y. Thus, the activation of XY indexes the separation of letters X and Y in the string.

In previous articles on the SERIOL model, I have assumed that bigram activations were influenced by letter activations [Whi99, Whi01a, Whi04a, Whi04c]. However, this assumption is inconsistent with emerging evidence on the weak positional effects of priming at the word level [Gra04a]. Therefore, I now take bigram activation levels to be affected only by separation of the constituent letters.

Following the evidence for a special role for external letters, the string is anchored to these endpoints via edge bigrams.<sup>5</sup> That is, bigram \*X is activated when letter X is preceded by a space, and bigram Y\* is activated when letter Y is followed by a space. In contrast to other bigrams, an edge bigram cannot become partially activated (i.e., by the second or next-to-last letter). Thus I assume a special mechanism for the activation of edge bigrams, which operates somewhat differently than for bigrams detecting a pair of letters. The details of this edge detection are left for future work.

Because letters are activated sequentially, bigram activations occur sequentially. For example, the input *cart* first activates bigram node \*C (when letter node C fires), then CA (when A fires), then AR and CR (when R fires), then RT, AT, and CT (when T fires), and then T\*.

---

<sup>5</sup>This is a new assumption. The importance of the external letters was formerly captured via high activations of bigrams containing those letters. However, now that bigram activation levels do not reflect letter activation levels, edge bigrams are now assumed.

## 6.2.4 Bigram Layer to Letter Layer

Bigram nodes connect to word nodes via weighted connections. The weight on a bigram-word connection is proportional to the activation level of that bigram when that word is presented as input. (As would result from Hebbian learning.) As is usual in neural network models, the weight vector is normalized, so that bigrams making up shorter words have higher connection weights than bigrams making up longer words. For example, this allows the string *tee* to activate the word node TEE more than TEETHE.<sup>6</sup> The input to a word node is the dot product of the weight vector and input vector. The input vector changes over time, because bigram activations occur serially, as indicated above. The activation of a word node at time  $t$  is a function of its input at time  $t$  and its activation at time  $t - 1$ . Lateral inhibition within the word layer also operates over time.

## 6.3 Summary

In the following, I summarize the important assumptions in the model.

- Edge Layer
  - Retinotopic
  - Activation levels based on acuity gradient.
  - Representation of fovea split across hemispheres.
- Feature Layer (for a left-to-right language)

---

<sup>6</sup>Normalization is another new assumption. Information concerning the length of the string was formerly carried on the activations of bigrams which represented the final letter.

- Retinotopic, representation still split across hemispheres
- Locational Gradient - Activation decreases from left to right.
- Locational Gradient formed by hemisphere-specific processing:
  - \* Stronger excitation to RH than LH.
  - \* Left-to-right lateral inhibition with a hemisphere, much stronger in RH.
  - \* RH inhibits LH.
- Letter Layer
  - Location- and position-independent letter nodes, located in LH.
  - Letter nodes undergo sub-threshold oscillations in synchrony.
  - Lateral inhibition between letter nodes.
  - Interaction of oscillations, lateral inhibition, and locational-gradient input gives serial firing.
  - Letter node activation depends on:
    - \* Firing rate - determined by input level.
    - \* Firing duration - determined by when next letter starts to fire, which is determined by the input level to that letter.
- Bigram Layer
  - Bigram XY activated when letter X fires and then letter Y fires.
  - Activation of bigram XY decreases with the amount of time between the firing of letter X and letter Y.
  - Edge bigrams also activated.

- Word Layer
  - Receives weighted connections from bigram layer.
  - Weight vectors are normalized to give advantage to shorter words.
  - Lateral inhibition operates as bigrams sequentially activate word nodes.

## **Chapter 7**

### **Account and Simulations of LPE Behavioral Results**

Having specified the SERIOL model, and motivated the different processing layers, I next discuss in more detail how the model accounts for the behavioral results, with the use of implemented models in some cases. The topics are presented in roughly the same order as the review of the experimental results.

#### **7.1 Word Level**

##### **7.1.1 Bigrams**

I start with a simulation of the bigram and word layers, based on a database of over 3,500 monosyllabic words. The most fundamental requirement is that the bigram representation of letter string should activate the corresponding word node more highly than any other word node. Thus one goal of the simulation is to show that the bigram representation does indeed allow correct word recognition. In addition to demonstrating the viability of the bigrams, another goal is to reconcile some conflicting results concerning positional effects at the word level. Recent priming data on long words has demonstrated that facilitation is rather insensitive to position of matched letters in the target. That is, when a prime is 4 or 5 letters, and a target is 7 or 9 letters, there is little difference in facilitation

between primes matching on the first letters versus the final letters [Gra04a]. Yet aphasic [Whi99] and perceptual data [Hum90, Mon98] indicate an advantage for the initial letters over the final letters. In a previous implementation of the bigram and word levels, I simulated the aphasic data using bigram activations that were sensitive to letter activations (and therefore to string position) [Whi99]. However, this assumption that letter position influences bigram activations is inconsistent with the lack of positional effects in priming. Another problem is that the original simulation required an additional assumption - that input to the letter level was reduced in aphasics, thereby pushing the firing of the final letter near the end of the oscillatory cycle, yielding a low activation level for the final letter (as opposed to the usual final-letter advantage.) This was necessary to simulate the finding that the final letter is the least likely to be preserved in an erroneous response.

Therefore, I sought to implement an improved bigram-to-word simulation that demonstrates both a weak positional priming effect, and the strong positional error pattern in the aphasic data (ideally without requiring additional assumptions about activation patterns at the letter level). In the original simulation, the temporal aspect of bigram and word activations was not considered, nor was lateral inhibition within the word layer. Rather, a bigram vector activated the word layer in a single time step in a purely bottom up manner. However, a more realistic simulation which includes these factors may allow the above goals to be met. It may be the case that the aphasic error pattern arises from a temporal activation pattern, rather than a positional one. That is, bigrams that are matched early in the word-activation process could have an advantage over those that are matched later (due to ongoing lateral inhibition within the word layer), even though bigram activations do not vary with position. Based on these ideas, I implemented the following simulation, which met the three goals: (1) correct recognition of

all words in the database; (2) replication of aphasic error pattern under noise; (3) lack of positional effects in target-node activations. I first give a brief overview of the simulation.

The input layer was comprised of bigram nodes, and the output layer consisted of word nodes representing all words in a database of 3650 single-syllable English words. The input layer connected directly to the output layer. Bigram-to-word weights were set according to the principles in section 6.2.4. Bigram activations were clamped sequentially, as discussed in section 6.2.3. Lateral inhibition within the word layer occurred after each set of bigram activations. Lateral inhibition was included to show that the temporal development of word-level activations could account for the aphasic error pattern. It was not used to simulate settling (reaction) time. Thus the word node having the highest activation following presentation of the final bigram was simply selected as the response. Aphasic performance was simulated by adding noise to the word level. Priming was simulated by noting target-node activation under partial input.

Next the simulation is specified in more detail. The functions implementing normalization and lateral inhibition were chosen on the basis of convenience and computational efficiency, rather than biological plausibility. In the following,  $C$  denotes a parameter. Let  $B^{xy}$  denote a bigram node representing the letter  $x$  followed by the letter  $y$ . Its activation,  $A$ , for a string  $S$  is a function of the number of letters separating  $x$  and  $y$ , denoted  $Sep$ .  $A(B^{xy}, S) = 1.0$  for  $Sep = 0$ ,  $C_{S1}$  for  $Sep = 1$ ,  $C_{S2}$  for  $Sep = 2$ , and 0 otherwise. Let  $Wd^S$  represent a word node encoding string  $S$ . The weight from a bigram node to a word node is given by:

$$W(B^{xy}, Wd^S) = \frac{C_{nrm}}{Len(S) + C_{nrm}} * A(B^{xy}, S)$$

where  $Len(S)$  gives the length of the string. This scaling of the bigram's activa-



tion value provides normalization by decreasing the weights for longer words, via division by  $Len(S)$ . The constant  $C_{norm}$  modulates this normalization; the higher its value, the less the effect. (If a bigram receives two different activation levels for a word, the larger of  $A(B^{xy}, S)$  is taken.)

A string  $S$  is presented over  $Len(S) + 1$  time steps. At each time step  $t$ , the bigrams are clamped to the values that would arise from the activation of the letter in position  $t$ . Word-level activations are then updated in two stages. (1) For each word node, the incoming activation is simply added to the current activation. The incoming activation is given by the dot product of the bigram vector and the word node’s weight vector. (2) The effects of lateral inhibition are simulated by updating each word node’s activation as follows:

$$A(Wd^S, t) = C_{inh} * \frac{A(Wd^S, t)}{MaxA(t)} + (1.0 - C_{inh}) * A(Wd^S, t)$$

where  $MaxA(t)$  is the activation of the word node having the highest activation. The constant  $C_{inh}$  (which takes values from 0 to 1.0) determines the overall contribution of inhibition. When  $C_{inh}$  is 0.0, the activation remains unchanged; when  $C_{inh}$  is 1.0, the activation is weighted by the ratio of the activation to the maximum activation. (Thus, the lower the activation value is with respect to the maximum activation, the more the activation is reduced, thereby simulating the effect of lateral inhibition.)

The parameters were hand-tuned to meet the above three goals. These goals are often at cross purposes. Goal (1) requires normalization of the weight vector. Yet if shorter words have too much of an advantage, they excessively inhibit longer words, under the inhibition required for goal (2). Goal (2) requires strong positional effects, while goal (3) requires weak positional effects.

A range of parameter values near the following values yielded reasonable re-

sults; the results for these particular values are presented.

$$C_{S1} = 0.8 \quad C_{S2} = 0.2 \quad C_{nrm} = 50 \quad C_{inh} = 0.5$$

All words in the database were recognized correctly, under the requirement that difference between the activation of the target word and the next highest word be at least 0.2. The most challenging task was to distinguish between TEE, THEE, TEETH, and TEETHE.

Priming was simulated by including the strings CDFGHKLMN, LMNPQRS, and STVWX in the database, and calculating their activation when a partial match was used as input. For example, to simulate a prime corresponding to the final five letters of a nine-letter word (56789), the activation of the CDFGHKLMN node is calculated for the input HKLMN. Table 7.1 gives the results. For seven- and nine-letter targets, there was a very weak advantage (of about 0.15) for initial versus final primes, which is numerically consistent with the experimental results [Gra04a]. Primes which did not experimentally produce facilitation all yielded simulated activation levels ( $< 3.9$ ) that were lower than activations of all primes that did produce facilitation ( $> 4.2$ ). There was a strong correlation between amount of facilitation and simulated score, as shown in Figure 7.1.

The large difference in the values of  $C_{S1}$  and  $C_{S2}$  was required to allow 13459 to give a considerably lower score than 6789 (for a 9-letter target), in accordance with the finding that only the latter produced priming. However,  $C_{S2}$  had to remain non-zero in order to be consistent with the finding that 125436 primes a six-letter target, while 12d4d6 does not [Per04]. That is, if  $C_{S2}$  is 0, bigrams 25 and 36 have weights of 0, erroneously giving no difference between these two types of primes.

A lesion was simulated by adding normally distributed noise to each word

Prime	Act	Fac (ms)
1234	4.71	36*
2345	4.73	32*
1245	4.50	31*
12345	6.01	45*
34567	5.84	37*
13457	5.58	29*
1234	4.56	36*
4567	4.41	32*
1357	3.71	12
15437	2.69	0
73451	2.30	7
12345	5.75	30*
56789	5.62	26*
1234	4.35	23*
6789	4.22	19*
14569	3.86	12
1469	2.20	8

Table 7.1: Simulated and experimental results for priming conditions from [Gra04a]. *Act* denotes the activation of the target node in the simulation for the given prime. *Fac* denotes the the facilitation for that prime in the experimental results (difference between reaction times for the control condition (dddd or ddddd) and the prime condition), where \* denotes facilitation is statistically significant. The top group is five-letter targets; the middle group is seven-letter targets and the bottom group is nine-letter targets.

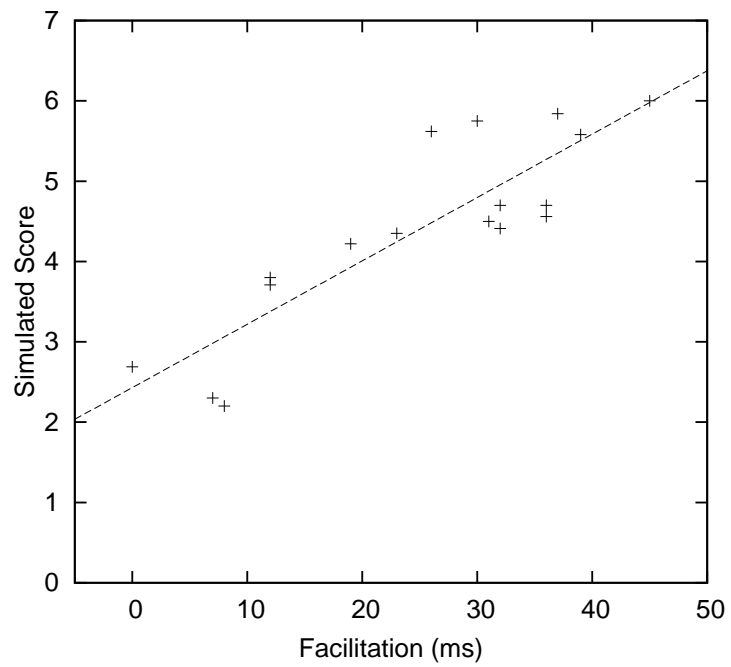


Figure 7.1: Comparison of simulated score and amount of facilitation using data from Table 7.1 ( $r=.87$ ;  $p < .0001$ ).

node at each time step (prior to the inhibition). Noise with mean 0.3 and standard deviation 0.35 yielded good results, shown in figure 7.2. As is evident, the probability of retaining a letter decreased with its position. This is not merely an artifact of the scoring method (in which a retained letter had to be in the correct absolute position), as scoring from right to left did not yield this pattern. Furthermore, this decreasing pattern was not present when the simulation was run without lateral inhibition. (See Figure 7.3). Thus, under inhibition, words that are highly activated early come to dominate. Therefore, final letters have less influence than the initial letters even though their bigrams are activated to same level.

The results of the lesioned simulation also showed other similarities to the experimental data. Aphasic subjects tended to preserve word length in their erroneous responses. Average response length to targets of lengths 3-6 were 4.0, 4.2, 4.9, and 5.9, respectively [Whi99]. The simulated data also showed sensitivity to target length, giving 4.2, 4.8, 5.1, and 5.8. Retention level at a given position tended to increase with target length for both the aphasics and the simulation. For example, for position 3, experimental retention rates were 40%, 55%, 65%, and 55% for target lengths 3-6, respectively. The simulated data exaggerated this effect, giving 36%, 48%, 81%, and 92%.

Thus the simulation accomplished the stated goals. There was a weak positional effect for priming, but a strong positional effect in the presence of noise. In the priming simulation, the target node's activation was primarily influenced by the number and separation of the prime's bigrams, while the temporal nature of the inhibition only had a small effect. In the lesion simulation, potential erroneous responses that were not highly activated initially became inhibited and remained at a disadvantage. Therefore, retention level was highest for early string

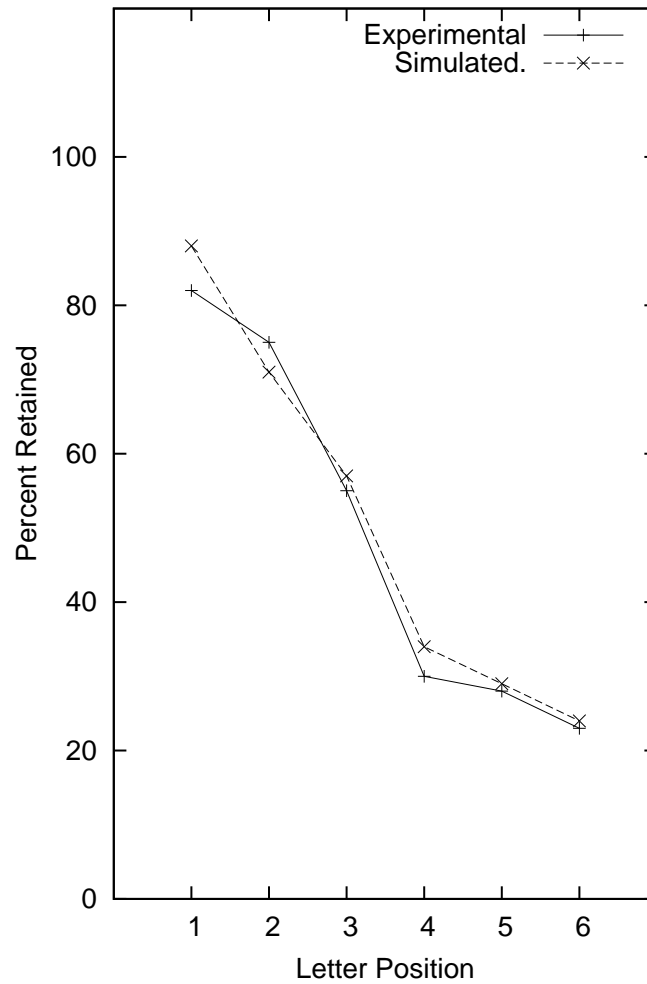


Figure 7.2: Experimental [Whi99] and simulated results for the aphasic error pattern. The *percent retained* refers to the percentage of erroneous trials in which the letter in the  $i$ th position in the target occurred in the  $i$ th position the response ( $n = 201$  for experiment;  $n = 363$  for simulation). Data are collapsed over target lengths of three to six. (In the both the experimental data and the simulation, there was also a decreasing pattern within each target length.)

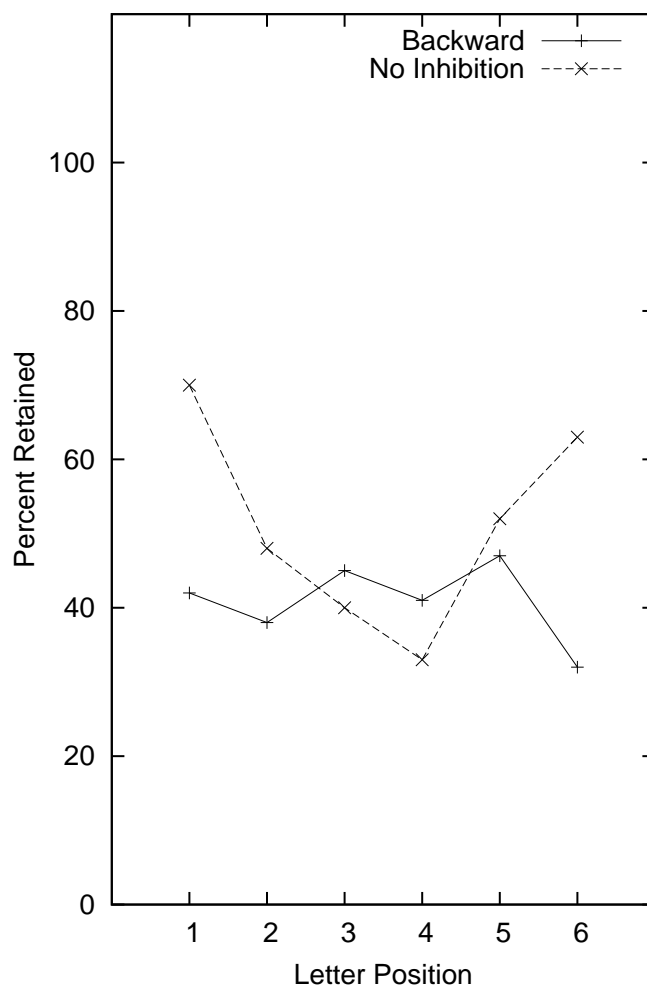


Figure 7.3: Simulation results under backward scoring, and no inhibition. In backward scoring, the target and response are aligned at the final letter, and scored from right to left. In this case, position 1 corresponds to the final letter, 2 corresponds to the next-to-last letter, etc. The backward results are from the same simulation run as Figure 7.2. For the no-inhibition condition, a new simulation was run with  $C_{inh} = 0$ , and scored in the forward manner. Because backward scoring yielded a relatively flat pattern, and no inhibition yielded a V-shaped pattern, this shows that the decreasing pattern in Figure 7.2 was not merely an artifact of the scoring method.

positions and decreased across the string, giving a strong positional effect.

The principles implemented in this simulation are also consistent with other priming and perceptual data. The left-to-right activation of bigrams accounts for the initial-letter advantage when only a single letter is matched in the prime [Hum90], and the perceptual error pattern, in which letter retention decreases across the string (like the aphasic error pattern) [Hum90, Mon04]. The anchoring of the external letters (via edge bigrams) accounts for their positional specificity, wherein priming does not occur when an external letter is moved to an internal position [Hum90, Per03]. For simplicity in the simulation, internal bigrams were weighted as highly as edge bigrams. However, edge bigrams may actually be weighted higher, which would account for the finding that matching the external letters produces more facilitation than matching any other two letters of a four-letter target [Hum90]. When three out of four letters are matched in a prime, there is no positional specificity [Hum90], consistent with the weak positional effects in the priming simulations. Next I discuss a word-level effect that is explained at the letter level.

### **7.1.2 Letters**

Sequential activation at the letter level explains the observed interaction between word length and rotation angle in the lexical-decision experiment in which the stimuli were rotated [Kor85] (as discussed in section 8.1). Recall that length had no effect for small angles, while each additional letter delayed RTs by about 200 ms for large angles. For intermediate angles, RTs were neither flat nor linear. These data are redisplayed in Figure 7.4. The authors concluded that it was not possible to explain this data under a single unitary principle. However, the SERIOL model does allow such an explanation.



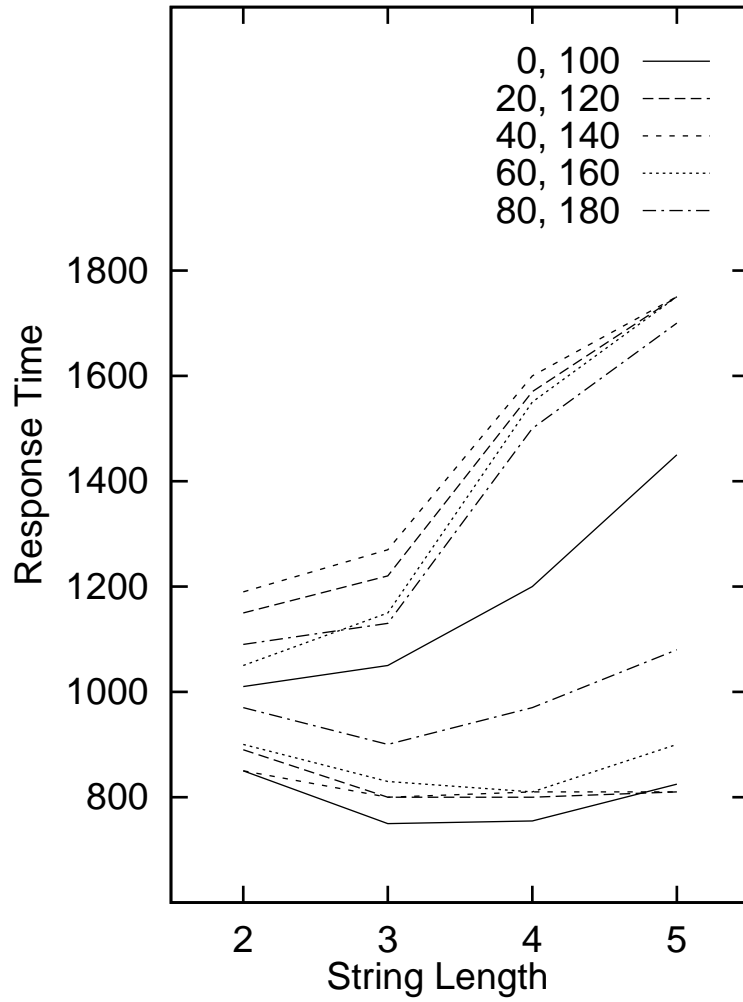


Figure 7.4: Experimental reaction times (in milliseconds) for the rotated-string lexical-decision task. Each line represents one angle of rotation, where the lower lines correspond to  $0^\circ$  through  $80^\circ$ , and the upper lines correspond to  $100^\circ$  to  $180^\circ$ .

As discussed in section 8.1, the presence or absence of a length effect does not necessarily diagnose whether lexical access occurs serially or in parallel. There could be no length effect under serial access if earlier firing of the final letter for shorter words is offset by longer settling time at the word level. However, length effects may arise under conditions of degraded presentation, when input levels to letter nodes are reduced such that it takes multiple oscillatory cycles to represent a sequence of letters that is normally represented in a single cycle. I suggest that such a phenomenon underlies the RT results from the rotated word experiment. This analysis implies that such length effects should depend on the time scale of the oscillatory cycle. Recall that for the largest rotation angles, each additional letter increased RTs by approximately 200 ms, which is on the order of the length of the proposed oscillatory cycle. Thus I propose that a unitary principle can explain this data - namely, that letter position is encoded temporally via an oscillatory carrier wave. When the input is degraded (by rotating the letter string), the underlying temporal nature of the encoding is exposed. The feasibility of explaining this data under the SERIOL model is next demonstrated via a simulation, which was first presented in [Whi02].

## Simulation

I assume that subjects performed the lexical decision task by mentally rotating the string to the canonical horizontal orientation, and then processing the string as usual. This assumption is consistent with the fact that RTs for two-letter words were smoothly increasing with rotation angle. It is also assumed that the act of mental rotation decreases the amount of input reaching the letter nodes, and that this degradation increases with the amount of rotation. These assumptions, in conjunction with the SERIOL model, provide a natural explanation for the

experimental data. Up to a certain amount of rotation, there is still sufficient input to activate all the letters within a single oscillatory cycle (i.e., up to  $60^\circ$ ). After that point, there is sufficient input to activate all of the letters in shorter words, while longer strings require an additional cycle (i.e., for  $80^\circ$  and  $100^\circ$ ). This accounts for the intermediate region where RTs are neither constant nor linear. With further degradation, only two-letter words can be represented in a single cycle; each additional letter requires an additional cycle (i.e.,  $120^\circ$  to  $180^\circ$ ).

It is assumed that once the mental image of a letter has activated a letter node, that image is inhibited. This allows a determination of whether all letters have been processed. However, bigram activation depends on the ordered firing of letter nodes within a single oscillatory cycle. If severely degraded input causes each letter node to fire on a separate cycle, how then could the bigram nodes become activated? It is assumed that letters which have already fired can re-fire again on successive cycles. However, this re-firing can't be triggered by bottom-up input, since the mental image is inhibited once it activates a letter node. How then could a previously activated letter node re-fire?

It has been proposed that an after-depolarization (ADP), which has been observed in cortical pyramidal cells following spiking, can maintain short-term memory across oscillatory cycles [Lis95]. The ADP is a slow, steady increase in excitability, peaking at approximately 200 ms post-spike. The temporal gradient of the ADP can maintain the firing order of elements across oscillatory cycles, in the absence of bottom-up input, as demonstrated in a simulation [Lis95]. Thus, this mechanism could maintain the firing order of letter nodes that have been previously activated.

I have implemented a simulation of the RT for the rotated word experiment based on these ideas. The interaction between the underlying oscillatory cycle,

external input levels, lateral inhibition, and the ADP was modeled in order to arrive at a firing time for the final letter of the string. This firing time, combined with other quantities, gives the modeled RT.

Next the details of the implemented model are presented. Instantiating the theoretical framework in a simulation entails the specification of quite a few parameters. Most of these parameters are related to the neuronal dynamics (ADP, oscillations, and inhibition) and are set to physiologically plausible values, similar to those used in [Lis95]. In fitting the computational model to the experimental data, the primary focus of optimization was the input function. This function was hand tuned.

## Reaction-Time Equation

The modeled RT,  $R$ , is given by:

$$R(\theta, l) = C_{BR} + H(\theta) + W(\theta, l)$$

where  $\theta$  denotes the angle of rotation (given in degrees) and  $l$  denotes the string length.

$C_{BR}$  denotes a base RT, set to 730 ms.  $H$  denotes the time required to mentally rotate the string; it is a linearly increasing function of  $\theta$ . Fitting to the RTs for two-letter words gives:

$$H(\theta) = 1.5\theta$$

$W$  denotes the time required to activate all the letter nodes corresponding to the string; that is,  $W$  is the first time at which the final letter node fires. The functions which determine  $W$  are the instantiation of the SERIOL framework. These functions specify the activation of the letter nodes.

## Letter-node Equations

Following Lisman and Idiart [Lis95], letter nodes are modeled as units that undergo a sub-threshold oscillatory drive, exhibit an increase in excitability after firing (ADP), and send inhibitory inputs to each other. The membrane potential,  $V$ , of a letter node is given by:

$$V(\theta, i, t, c) = O(t) + A(i, t) - I(t) + E(\theta, i, c)$$

where  $i$  denotes the letter node representing the  $i$ th letter of the word,  $t$  denotes time (ranging from 0 to length of the oscillatory cycle), and  $c$  denotes the number of completed oscillatory cycles.  $O$  gives the oscillatory drive,  $A$  gives the ADP,  $I$  gives the inhibitory input, and  $E$  gives the excitatory external input (originating from the feature level).

A node fires when  $V$  exceeds a threshold,  $C_{TH}$ , which is specified relative to resting potential, and set to 10mV. Firing causes the node's ADP component to be reset and inhibition to be sent to the other nodes.  $E$  is permanently set to 0 the first time that a node fires.

The oscillatory function  $O$  has a cycle length of 200 ms, and linearly increases from -5mV to 5mV during the first half of the cycle, and decreases back to -5mV during the second half.

The ADP and inhibition are modeled by functions of the form:

$$F(t; M, T) = M * (t/T)^{1.5} * \exp(1 - t/T)$$

which increases to a maximal value (controlled by parameter  $M$ ) and then decreases (on a time scale controlled by parameter  $T$ ). The ADP is given by:

$$A(i, t) = F(t_i; M_A, T_A)$$

where  $t_i$  denotes the amount of time since node  $i$  last fired. ( $A$  is 0 if the node has not yet fired.) The inhibition is the sum of the inhibitory inputs from all letter nodes, given by:

$$I(i, t) = \sum_{j=1}^l F(t_j; M_I, T_I).$$

These parameters were hand tuned (in conjunction with  $E$ ) to give the desired firing pattern. The following values were used:  $T_A = 200$  ms,  $M_A = 11$  mV,  $T_I = 3$  ms,  $M_I = 3$  mV.

The external input  $E$  is a decreasing function of position  $i$ ; this corresponds to the locational gradient at the feature level. In the following,  $E$  is specified for a node that has not yet fired; if node  $i$  has already fired,  $E(\theta, i, c) = 0$ . First we consider the initial oscillatory cycle, for an unrotated string. The following function was used:

$$E(0^\circ, i, 0) = 10.6 - 0.5i$$

Mental rotation degrades the external input, so  $E$  decreases as  $\theta$  increases:

$$E(\theta + 20^\circ, i, 0) = E(\theta, i, 0) - 0.65 \sin(\theta + 20^\circ)$$

External input builds up over time, with  $E$  increasing after each oscillatory cycle:

$$E(\theta, i, c + 1) = E(\theta, i, c) + 0.2$$

## Results

A simulation for each combination of  $l$  and  $\theta$  was run, starting at time  $t = 0$  and using a time step of 1 ms. At each time step, each letter node's potential was calculated using the equation for  $V$ . For all rotation angles and string lengths, all active letters of the string fired in the correct sequence on each cycle. The value of  $W(\theta, l)$  was set to  $t^{final} + 200c^{final}$  where  $t^{final}$  and  $c^{final}$  are the

first  $t, c$  at which  $V(\theta, l, t, c) > C_{th}$ . For example, for  $\theta = 0^\circ$  and  $l = 4$ , nodes 1, 2, 3, and 4 fired at  $t = 49, 63, 74,$  and  $84$ , respectively, during the first cycle, giving  $W(0^\circ, 4) = 84$ . For  $\theta = 180^\circ$  and  $l = 4$ , nodes 1 and 2 fired at  $t = 86$  and  $100$  in the first cycle. In the second cycle, nodes 1, 2, and 3 fired at  $t = 52, 65,$  and  $94$ . In the third cycle, nodes 1, 2, 3, and 4 fired at  $t = 43, 55, 66,$  and  $97$ , giving  $W(180^\circ, 4) = 97 + 200 * 2 = 497$ . Each node refired earlier in successive cycles due to the ADP. This earlier firing, in conjunction with increasing external input, allowed more letters to fire on each cycle. The slowly increasing ramp of the ADP, in conjunction with lateral inhibition, maintained the proper firing sequence across cycles.

The RT was then calculated using the equation for  $R$ . The results are given in Figure 7.5. The simulation reproduced the experimental pattern of relatively flat RTs for small angles, and rapidly increasing RTs for large angles, with a mixture of the two patterns for intermediate angles. In the experimental data, there was also a pervasive disadvantage for two-letter words, which is not captured by the simulation.<sup>1</sup>

While this simulation may seem complex, it is merely an instantiation of the previously specified dynamics for induction of the serial firing pattern (with the addition of the ADP), coupled with the assumption that bottom-up input decreases with rotation angle and increases over time. It illustrates the simple idea that the interaction between string length and rotation angle arose because multiple oscillatory cycles were progressively required to represent all of the letters

---

<sup>1</sup>The simulated results can be made to look more like the experimental results by simply adding 100 ms to all two-letter reaction times. It is unclear what the source of this disadvantage is. It may be related to the fact that vowels are normally explicitly expressed in Hebrew, leading to ambiguity for very short strings.

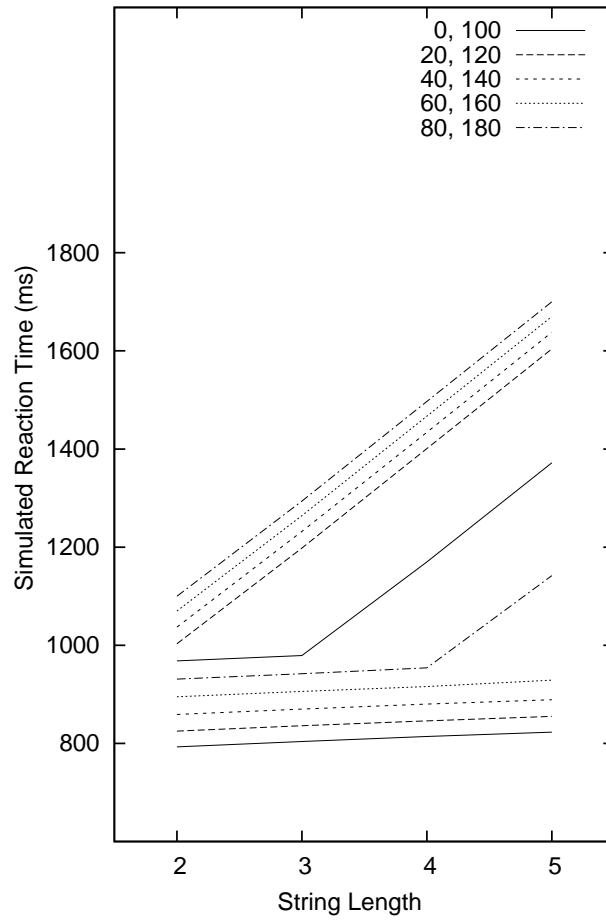


Figure 7.5: Simulated reaction times for the rotated-string, lexical-decision task. Notation is the same as Figure 7.4.



of the input string. Applying the SERIOL model to this experimental data yields a natural explanation of these data. It accounts for the finding that there is an intermediate region of rotation angles where processing seems neither fully parallel nor fully serial, which is difficult to explain otherwise. It also predicts the finding that the increase in RT per letter for large rotation angles is on the order of 200 ms (i.e., an oscillatory cycle in the theta range).

## 7.2 Letter Perceptibility Patterns

As discussed in section 6.2.2, the induction of the serial encoding leads to differing activations at the letter level. These activation patterns depend on the interaction of the locational gradient and the oscillatory cycle. Such dynamics explain observed patterns of letter perceptibility, which vary with string position and visual field, as follows.

For a centrally fixated string, the initial-letter advantage and final-letter advantage arise for different reasons. The initial letter has an advantage because it receives the highest level of bottom-up input, allowing it to fire the fastest. It receives the most input because it is not inhibited from the left at the feature level. The final letter has an advantage because it is not inhibited by a subsequent letter during the induction of serial firing. That is, it is not inhibited from the right at the letter level. Thus, like others, I also attribute the advantage for the external letters to a lack of lateral inhibition. But this arises because of string-specific processing, and not from a lack of masking at a very low level (as is generally assumed). This proposal is consistent with the finding that there is no external symbol advantage for strings of symbols that are not letters or numbers [Mas82]. For such centrally fixated symbol strings, the external symbol is

the least well perceived, as would be expected on the basis of acuity. Strings of letters and numbers show an external symbol advantage because of string-specific processing.

This predicts that it should be possible to differentially affect the initial- and final-letter advantages. The initial-letter advantage should disappear if the amount of bottom-up input to the initial letter is not significantly higher than to the other letters. The final-letter advantage should disappear if the firing of the final letter is pushed late into the oscillatory cycle. As we shall see, this is exactly what happens for brief, lateralized presentation of short strings. First however, a more in depth consideration of activation patterns at the feature level is required.

Recall that locational gradient formation requires different processing across the hemispheres. In the RVF/LH, the acuity gradient serves as the locational gradient. In the LVF/RH, the acuity gradient is inverted via strong bottom-up excitation and left-to-right lateral inhibition. Because the locational gradient is formed by different mechanisms in each hemisphere, the shape of the resulting gradient may vary with hemisphere, especially at large eccentricities. Recall that acuity falls off fastest near fixation, and falls off more slowly as eccentricity increases. That is, the slope of the acuity gradient is steepest near fixation, and becomes shallower as eccentricity increases. Since the RVF/LH locational gradient is based on the acuity gradient, this implies that the RVF/LH locational gradient becomes more shallow as eccentricity increases. (See right half of Figure 7.6.)

In the LVF/RH, formation of the locational gradient depends on left-to-right lateral inhibition. This processing is optimized to create the locational gradient for a small number of letters near fixation. For long strings at large eccentricities, inhibition may be too strong at early string positions (due to their relatively

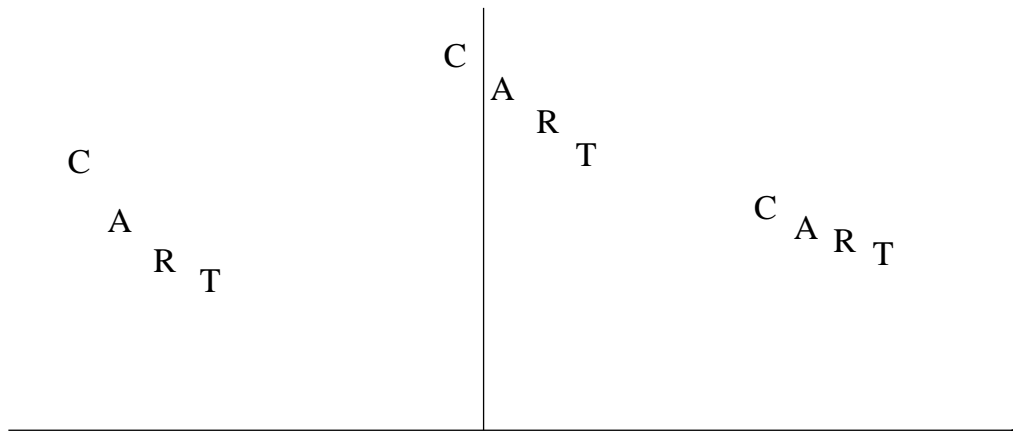


Figure 7.6: Schematic of locational gradients for the stimulus CART at three different presentation locations. The vertical axis represents activation, while the horizontal axis represents retinal location. For central presentation, the gradient is smoothly and rapidly decreasing. For RVF presentation, the gradient is shallower because the acuity gradient is shallower. For LVF presentation, the initial letter strongly inhibits nearby letters, but the gradient flattens out as acuity increases.

low level of activation), but may become too weak at later string positions (due to their increasing acuity). (See left half of Figure 7.6). Thus the prediction is that the locational gradient should vary with visual field. Assuming that letter perceptibility directly indexes letter activation levels, which depend on feature activation levels, this suggests that letter perceptibility patterns should vary with visual field. As discussed in section 4.2, they do indeed. Recall that letter perceptibility uniformly drops off with increasing string position in the RVF/LH. In contrast, in the LVF/RH, perceptibility drops off sharply across early string positions, and then flattens out for later string positions [Wol74]. This data is re-presented in the top panels of Figure 7.7. The proposed hemisphere-specific shapes of the locational gradient explain this data<sup>2</sup>, as shown by the results of a mathematical model (bottom panels of Figure 7.7), described next.

### 7.2.1 Mathematical Model

The data are modeled by calculating a feature-level activation, which is converted into a letter-level activation, which determines an accuracy score. At the feature level, the stronger excitation and left-to-right inhibition in the RH are modeled, as well as the cross-hemispheric inhibition from the LH to RH. The general form of the equations are presented first, then the specific instantiations of those equations are specified.

---

<sup>2</sup>I'd like to note that the theory of locational gradient formation was not formulated to explain this data. Rather, the theory was constructed to explain how a monotonically decreasing gradient could be formed, starting with the assumption of the acuity pattern. Once the theory was formulated, it predicted these activation patterns. Only then did I actually seek out relevant experimental data. The fact that existing data showed the predicted pattern convinced me that I was on the right track.

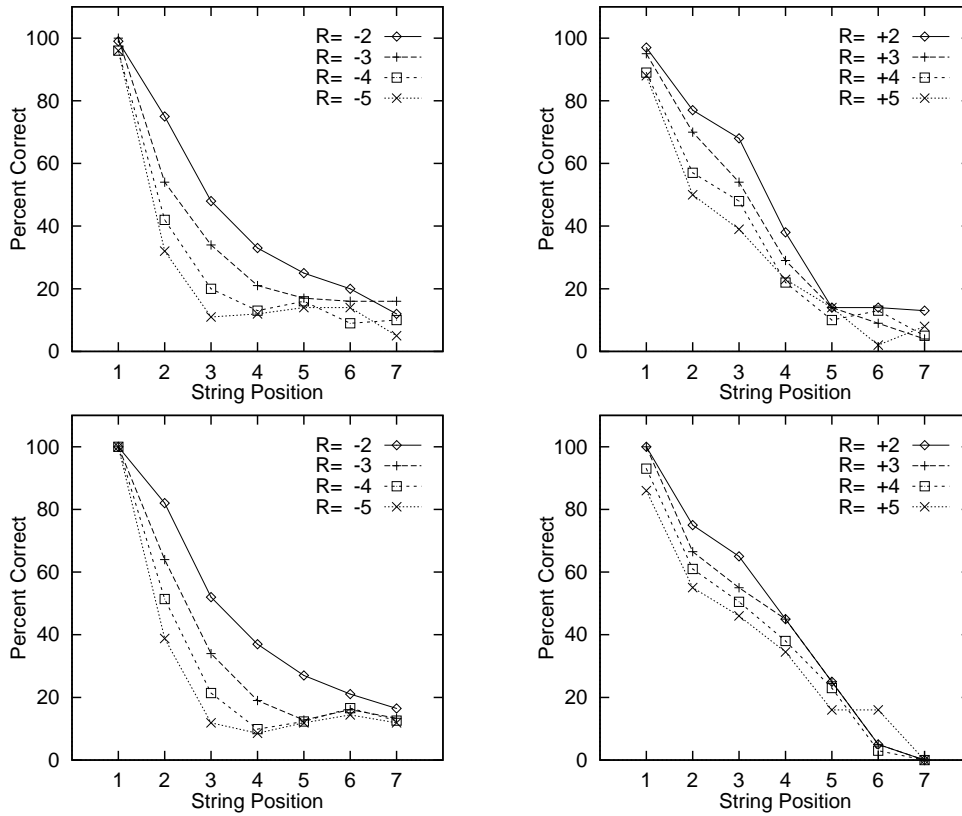


Figure 7.7: Experimental (top) and modeled (bottom) results of [Wol74], with LVF presentation on the left and RVF on the right. Each graph shows the effect of string position on perceptibility at a given retinal location (specified in  $R$  units of letter width).

All feature nodes comprising a single letter are assumed to reach a similar level of activation, which is determined by the retinal location,  $R$ , and string position,  $P$ , of that letter. For simplicity, fixation ( $R = 0$ ) is assigned to a single hemisphere, namely the RVF/LH. The activation of a feature node is denoted  $F$ . I first consider feature activations in the absence of hemispheric interaction, denoted  $F_h$ .  $F_h$  is determined by the combination of bottom-up excitatory input,  $E$ , and lateral inhibitory input,  $I$ , and is restricted to a maximal value,  $c_M$ . That is,

$$F_h(R, P) = \min(c_M, E(R) - I(R, P)).$$

In the following specification of  $F_h$ , “letter” will refer to a letter’s feature nodes.

## Bottom-Up Excitation

Excitatory input is a function of acuity, denoted  $C$ , and visual field (which is determined by  $R$ ):

$$E(R) = \begin{cases} C(R) & \text{if } R \geq 0 \\ c_E * C(R) & \text{if } R < 0 \end{cases}$$

where  $c_E \gg 1$ , reflecting the assumption of stronger excitatory input to the LVF/RH.  $E$  decreases as  $|R|$  increases, reflecting the acuity gradient.

## Lateral Inhibition

Lateral inhibitory input is the sum of inhibitory inputs from letters to the left of  $R$ . This quantity increases with the number of such letters, their activation levels, and the strength of the inhibitory connections. Rather than directly modeling the feedback processes underlying such lateral inhibition, the amount of inhibitory input is approximated as the activation of the leftmost letter’s features

weighted by a function of the number of letters sending inhibition.

The *leftmost letter* refers to the letter which lies farthest to the left within the same visual field as  $R$ . Its retinal location is denoted  $R_l$ . To determine the inhibitory input, the leftmost letter's activation is multiplied by a weighting function,  $W$ .  $W$  increases with the number of letters lying between  $R_l$  and  $R$ .  $W$  also depends on hemisphere; inhibitory connections are stronger in the LVF/RH than in the RVF/LH (as is necessary to invert the acuity gradient). Thus, we have

$$I(R, P) = F_h(R_l, P_l) * W(|R_l - R|, R)$$

where the first term on the right hand side gives the activation of the leftmost letter, and  $W$  is a non-decreasing function of  $|R - R_l|$ , which is larger for  $R < 0$  than for  $R > 0$ . If  $R = R_l$ ,  $W = 0$  (because the features of the leftmost letter do not receive inhibition).

## Cross-hemispheric Lateral Inhibition

The individual hemispheric gradients are “spliced together” via inhibition of the RVF/LH's letters by an amount proportional to the number of letters in the LVF/RH. That is,

$$F(R, P) = \begin{cases} F_h(R, P) - c_F * (P - R - 1) & \text{if } R \geq 0 \\ F_h(R, P) & \text{if } R < 0 \end{cases}$$

where  $c_F$  is a positive constant. This yields a decreasing gradient such that  $F(R, P) > F(R + 1, P + 1)$ .

## Specification of Feature-Level Parameters and Functions

To instantiate the feature level, values for the constants  $c_M$ ,  $c_E$ , and  $c_F$ , and definitions of the functions  $C$  and  $W$  must be supplied. The following allowed a good fit to the data:

$$c_M = 1.0 \quad c_E = 1.8 \quad c_F = 0.2$$

The acuity function is defined recursively:

$$C(|R| + 1) = C(|R|) - C_{dif}(|R| + 1)$$

$$C(0) = 1.1$$

$$C_{dif}(r) = \begin{cases} 0.1 & \text{if } 1 \leq r \leq 3 \\ 0.07 & \text{if } 3 < r \leq 6 \\ 0.05 & \text{if } 6 < r \leq 9 \\ 0.04 & \text{if } 9 < r. \end{cases}$$

$C_{dif}$  decreases as  $r$  increases, reflecting the decrease in the slope of the acuity gradient with increasing eccentricity. The definition of the inhibitory weighting function,  $W$ , is best displayed in tabular form:

$ R - R_i $	$R \geq 0$	$R < 0$
0	0.00	0.00
1	0.15	0.80
2	0.25	1.10
3	0.30	1.25
4	0.50	1.35
5	0.50	1.45
6	0.50	1.65



## The Letter Level

For simplicity, feature level activations were directly converted to letter level activations (rather than modeling the oscillatory cycle), as follows:

$$L(R, P) = \begin{cases} F(R, P) + 0.2 & \text{if } P = 1 \\ F(R, P) & \text{if } P > 1 \end{cases}$$

Thus, letter level activations are equivalent to feature level activations, except at the first position. This was necessary to provide a good fit the data, and corresponds to a non-linearity in the interaction of the oscillatory function and the locational gradient near the trough of the cycle for high levels of input. The letter activation was converted to the modeled accuracy by multiplying by 100 and bounding the result between 0 and 100.

The results are given in Figure 7.7. In the LVF/RH, increasing string position from 1 to 2 or from 2 to 3 has a strong effect because of the high level of inhibition. However, as string position continues to increase, there is less and less effect because the leftmost letter becomes less and less activated. Thus the perceptibility function flattens out. This effect is most pronounced at larger eccentricities where feature-level activations are lower. In the RVF/LH, increasing string position leads to an increase in the activation of the leftmost letter or to increased cross-hemispheric inhibition. Coupled with the weak inhibition, this leads to a slow, steady decrease in perceptibility as string position increases.

### 7.2.2 Short Strings

Next I consider perceptibility patterns for short strings (three or four letters) at large eccentricities, as discussed in section 4.2.3. In the following, *primacy* will signify that a letter is perceived better than all other letters, whereas *advantage*

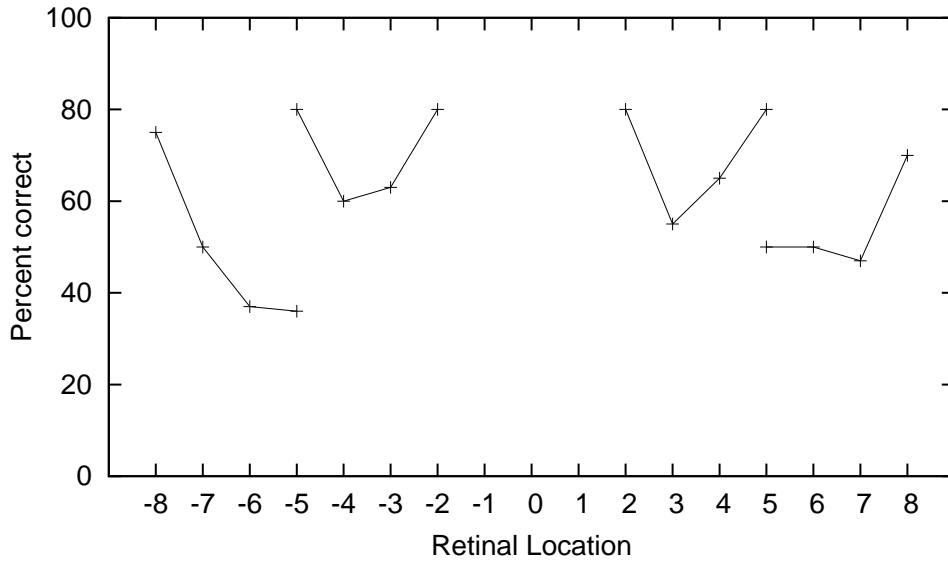


Figure 7.8: Experimental results from [Est76] for a four-letter string embedded in \$’s, occurring at two different retinal locations in each visual field. Exposure duration was 2400 ms. (Subjects were trained to maintain central fixation, and their gaze was monitored.)

will mean that an external letter is perceived better than the internal letters. Recall that in the LVF/RH, there is an initial-letter primacy, with little or no advantage for the final letter. In the RVF/LH, there is little or no advantage for the initial letter, and there is a final-letter primacy. Thus, in each visual field, the letter farthest from fixation is the best perceived, and the advantage for the other external letter is reduced. In particular, we will consider the results of [Est76], re-presented in Figure 7.8.

The proposed hemisphere-specific locational gradients, coupled with their interaction with the oscillatory cycle explain these patterns. In the LVF/RH, at the feature level, the initial letter is strongly excited, and strongly inhibits letters to the left. This leads to an initial-letter primacy, while the firing of the final

letter is pushed late into the oscillatory cycle, providing little advantage. In the RVF/LH, overall bottom-up excitation is weaker. Therefore, the activation of the initial letter's features is not boosted to a high level. Furthermore, there is weak left-to-right inhibition, while the acuity/location gradient is quite shallow. Therefore the activation of the second letter's features is quite close to that of the first letter. As a result, at the letter level, the firing of the first letter is rapidly cut off by the second letter. Each successive letter quickly inhibits the preceding letter (due to the shallow locational gradient), allowing the final letter to start firing early in the oscillatory cycle. Therefore the final letter can fire longer than the other letters, creating a final-letter primacy. The proposed activation patterns are displayed in Figure 7.9. This explains the perceptibility patterns for locations -8:-5 and 5:8.

This account also explains the initial/final difference at a single retinal location (at -5 and 5 in Figure 7.8). In the LVF/RH, the left-to-right inhibition creates a disadvantage for a final letter. In the RVF/LH, the shallow locational gradient creates a disadvantage for an initial letter because its firing is rapidly inhibited by the second letter.

In contrast to asymmetric patterns at the larger eccentricity, the perceptibility function is U-shaped for both -5:-2 and 2:5. Due to higher acuity, bottom-up input is higher overall. In the LVF/RH, this allows the final letter to start firing earlier in the cycle, creating a final-letter advantage. Along with the usual initial-letter advantage, this gives the U-shaped pattern. In the RVF/LH, the acuity/location gradient is steeper than for the larger eccentricity, so the difference in input to the first and second letters is larger, creating an initial-letter advantage and giving an overall U-shape.

Next we consider the implications of this account for the effects of exposure

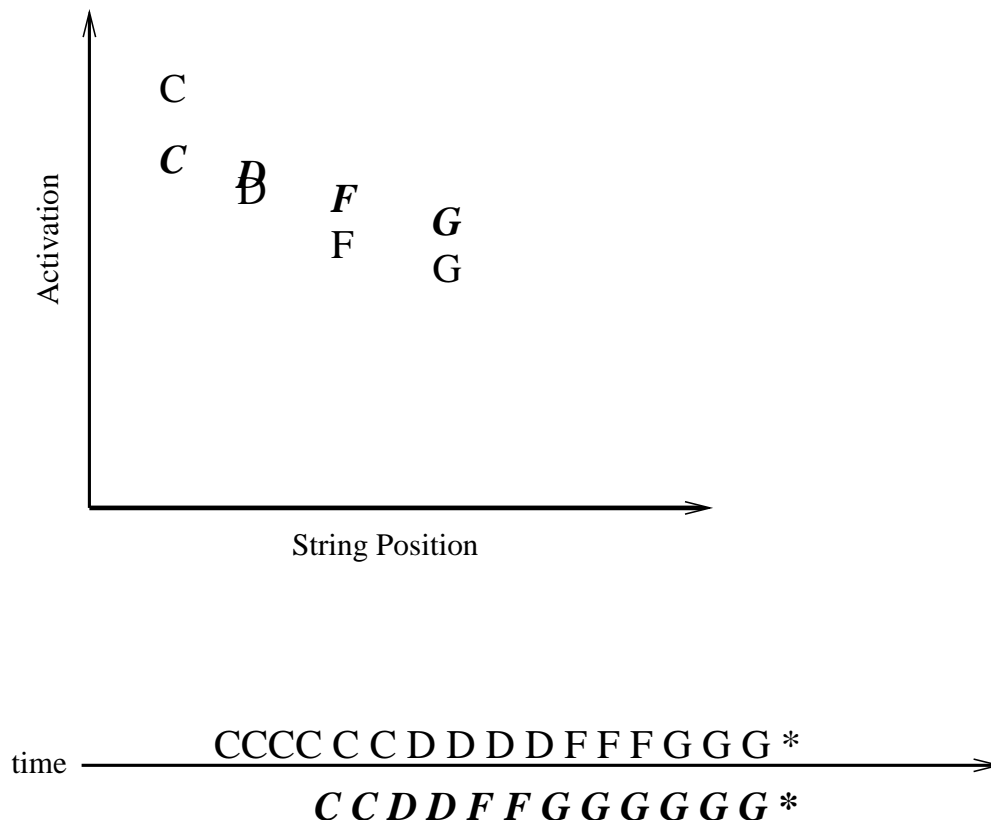


Figure 7.9: Locational gradient and resulting firing pattern for LVF/RH presentation (normal font) and RVF/LH presentation (bold italics). Top: Comparison of locational gradient for string CDFG under RVF/LH presentation and LVF/RH presentation. Bottom: Cartoon of resulting firing pattern at the letter level. The point in the oscillatory cycle at which the down phase prevents further firing is marked \*. In the LVF/RH, the first letter fires faster and longer than the other letters, because it receives a much higher level of input. The variations in the amount of bottom-up input create decreasing activation across the string. The final letter starts firing late in the cycle, and is soon cut off by the end of the oscillatory cycle, giving no final-letter advantage. In the RVF/LH, each letter rapidly cuts off firing of the previous letter, allowing the final letter to fire a long time. As a result, activation is flat across the string and rises for the final letter. These firing patterns account for the perceptibility patterns at the larger eccentricities in Figure 7.8.

duration for presentation at large eccentricities. Under the assumption that a longer exposure duration increases the overall level of bottom-up input, the above analysis suggests that the LVF initial-letter primacy and the RVF final-letter primacy should be differentially by variations in exposures. In the RVF, we would *not* expect to see a final-letter primacy at very brief exposures, because the very low level of input pushes the firing of the final letter late into the oscillatory cycle. As exposure duration increases, the firing of all the letters is shifted earlier and earlier into the cycle, allowing the final letter to fire longer and longer. In contrast, the activation of a non-final letter shouldn't change much, because its firing is still quickly cut off by the subsequent letter. Thus, a final-letter primacy should emerge as exposure duration increases.

However, in the LVF, the initial-letter primacy *should* be present at very brief durations, because strong left-to-right inhibition at the feature level does not depend on temporality, so it is always present. As exposure duration increases, the initial-letter should be the primary beneficiary, because, at the feature level, the increased bottom-up input to the non-initial letters is canceled by increased lateral inhibition from the first letter.

To summarize, in the RVF, the final-letter primacy should *not* be present at very brief exposures. Increasing exposure duration should primarily benefit the final letter, creating a final-letter primacy. In the LVF, the initial-letter primacy *should* be present a very brief exposures. Increasing exposure duration should primarily benefit the initial letter, increasing its primacy.

A search of the literature revealed that a relevant experiment had already been performed, in which retinal location and exposure duration were varied in a trigram identification task [Leg01]. However, the published data were not presented in a way that would allow evaluation of the above predictions. So I

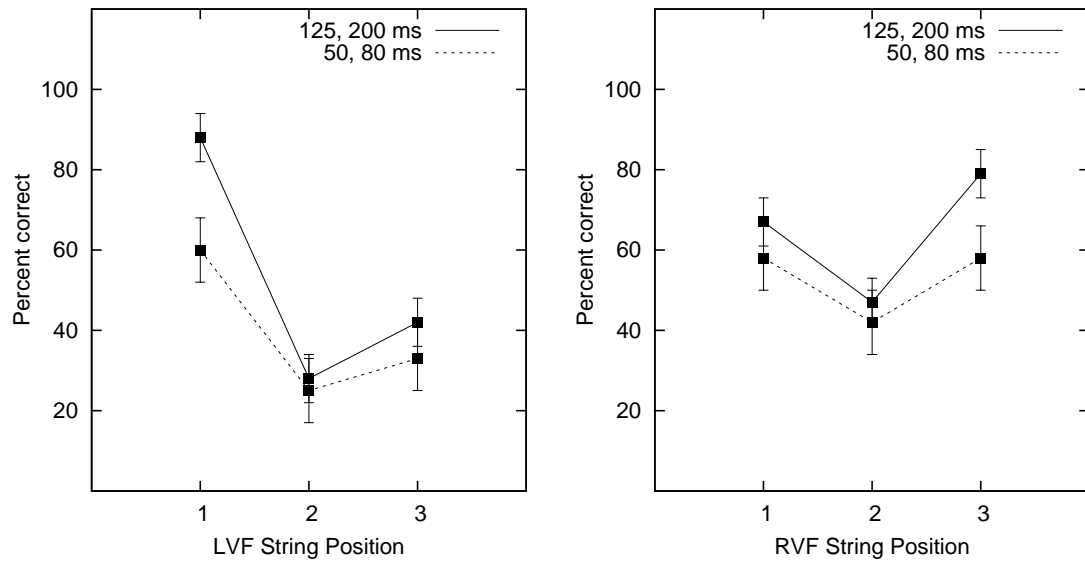


Figure 7.10: Results from Experiment 2 of [Leg01] for the two largest eccentricities, grouped by exposure duration, with 95% confidence intervals.

requested the raw data from the authors, who kindly provided it. The data were analyzed for the largest two eccentricities (-12::-10 and -11::-9 versus 9::11 and 10::12) for very brief exposures (50 ms and 80 ms) versus longer exposures (125 ms and 200 ms). This analysis did indeed reveal the predicted patterns, as shown in Figure 7.10.

This account also explains the error patterns observed for unilaterally presented vertical strings in a left-to-right language ([Hel95, Hle97], discussed in 4.2.3), under the assumption that the string is first mentally projected to the horizontal and then the locational gradient is formed as usual. The preponderance of the LVF pattern under bilateral presentation may reflect the cross-hemispheric inhibition necessary for locational gradient formation. For right-to-left languages, acuity gradient inversion would occur in the RVF/LH and cross-hemispheric in-

hibition would apply from the LH to the RH. This explains the observed reversal of the error patterns for Hebrew [Evi99]. For a vertical language, locational gradient formation would not occur along the horizontal axis, so there should be no left/right asymmetry. This explains the observed error patterns for vertical Japanese kana [Hell99].

### 7.3 Summary and Discussion

We have seen that bigrams allow correct word identification, and that the proposed mechanism of bigram activation (i.e., formation of the locational gradient; interaction of the locational gradient with the oscillatory cycle) has allowed a cohesive explanation of aphasic error patterns, priming data, perceptibility patterns, and the reaction-time pattern for rotated strings.

The central (and most controversial) proposal of the SERIOL model is the temporal encoding of letter order. While the above accounts do not directly prove a serial encoding, these phenomena are otherwise difficult to explain. The fact that the letters farthest from fixation are the best perceived is in complete contradiction to their acuity, and a possible underlying mechanism has heretofore never been proposed. However, this counterintuitive pattern arises naturally from the serial-encoding mechanisms (based on the principles of feature-level left-to-right inhibition and word-level right-to-left inhibition). Furthermore, the temporal development of this pattern (as exposure duration is increased) is exactly as predicted. The reaction-time pattern for rotated strings also defied explanation prior to the SERIOL model. Again, the model allows a natural explanation of this data (based on the number of oscillatory cycles required to represent the string). I note that the model was not designed to explain these phenomena.

Rather, these explanations fell out of the already-formulated model.

Normals shown very briefly presented strings and aphasics both show a strong positional error pattern, with retention rate falling off with position. However, there is little or no effect of position when primes possess more than two of the target's letters. The temporal nature of word activation explains the strong positional effect in the error patterns. In the presence of noise, small differences that occur early are exaggerated by lateral inhibition. Therefore, the initial letters have more influence than the final letters on the relative levels of word activations, even in the absence of position-specific activation patterns at the bigram level. In contrast, priming reveals activation of the target word in particular. In this case, the temporal development of competition is not very important; the size of the effect is dominated by the number of bigrams shared by the prime and target. Thus the serial encoding allows an explanation of the contrast between the error data and the priming data.

The model *was* originally designed to explain aphasic error pattern. However, the explanation of this pattern has evolved, while the model has remained basically the same. Originally, the positional error pattern was explained by varying letter activations (induced by the serial firing) that were passed on to the bigram level. However, this error pattern is now explained by directly considering the temporal aspect of bigram- and word-level activations. This required only a minor revision to the model - bigram activations are no longer sensitive to the positions of the constituent letters.

Thus, a temporal encoding allows these disparate phenomena to be explained, and provides an answer to the question of how a position-independent letter node could be dynamically bound to a string position. Furthermore, experiments in which two strings were briefly sequentially presented have provided direct evi-



dence of a serial read out [Har75, Nic76].

The serial encoding depends on the formation of the locational gradient, which requires hemisphere-specific processing, leading to hemisphere-specific activation patterns. Such activation patterns could also potentially explain the visual-field asymmetries at the lexical level, as discussed in the following two chapters.

## Chapter 8

### Asymmetry of the Length Effect

We have seen how the SERIOL model explains VF-specific letter perceptibility patterns. Word-level VF asymmetries have also been observed. Such asymmetries have generally been taken to reflect hemisphere-specific modes of lexical access. However, such an assumption conflicts with brain imaging evidence for a left-lateralized lexical access [Coh00, Deh04] (and with the SERIOL model, which assumes a single mode of lexical access). Yet, how could asymmetries at the lexical level arise under a single mode of lexical access? This chapter and the next provides an answer to this question.

One of the most studied asymmetries involves the effect of string length. I will concentrate on this asymmetry in this chapter. I first review the relevant experimental data. I then present the SERIOL account of the asymmetry, and an experiment testing this account.

#### 8.1 Experimental Data

It has long been recognized that string length has differing effects across the visual fields [Mel57, Bou73]. Ellis and Young performed an extensive series of experiments elucidating this interesting phenomenon. For lexical decision on

words of three to six letters, presentation to the RVF yields no length effect, while presentation to the LVF causes RTs to increase by approximately 20 ms for each additional letter [You85, Ell88]. This pattern is present even if the location of the initial letter is held fixed as string length is increased [You85], indicating that the asymmetry is not related to the acuity of the initial letter. For short stimuli, there is a small RVF advantage, and this advantage increases with word length, because RVF RTs do not increase, while LVF RTs do. These results have been taken as evidence for dual modes of lexical access, where the LH uses an efficient, parallel method of access, while the RH uses a less efficient, non-parallel mode of access [You85, Ell88].

However, there are difficulties with this proposal. As discussed in section 3.3, imaging evidence indicates that processing becomes left-lateralized at a prelexical level. Thus there can be only one mode of lexical access because lexical access is always routed through the LH. Furthermore, a split representation of the fovea means that the left half a centrally fixated word is projected to the RH, and the right half to the LH. Dual modes of lexical access would then lead to the unlikely scenario that each half of a word is accessed by a separate mechanism.

Indeed, the processing of fixated words is influenced by the number of letters in the LVF, but not the RVF, as would be expected under a split fovea [Bry96, Lav01a]. This is related to the phenomenon of the Optimal Viewing Position (OVP) [Ore84]. Word recognition is optimal when fixation falls between the first and middle letters of a word. The cost of moving away from this OVP varies with direction. When fixation falls at the first letter, there is a small decrement in performance that varies little with the length of the word (i.e. the number of letters falling in the RVF). However, when fixation falls at the final letter, there is a larger decrement in performance that increases with word length (i.e.,

the number of letters in the LVF). Brysbaert and colleagues [Bry96] investigated this phenomenon by varying the location of words of different lengths. Locations ranged from entirely within the LVF, to various fixation points within the word, to entirely within the RVF. There was a smooth pattern of performance as location was systematically changed. This indicates that the OVP and the LVF length effect arise from the same underlying factors.

What factors are involved has been a subject of some debate. One possibility is that the initial letters of a word are more informative, so there is an advantage for fixating near them [Ore84]. While it was demonstrated that informational content does affect the OVP, this cannot be the whole story. The advantage for fixating near the end of words in which the final letters are the most informative is much reduced compared to the advantage to fixating near the beginning of words in which the initial letters are the most informative [Ore84].

Therefore, the OVP probably also depends on hemispheric specificity. The more efficient processing in the RVF may arise from more direct access to the dominant hemisphere [Bry94, Bry96]. Alternatively, it could be due to effects of reading direction. In a left-to-right language, the to-be-processed text occurs in the RVF. Thus the LH may become specialized through perceptual learning to process upcoming text, giving a RVF advantage [Naz03, Naz04a].

One way of evaluating these possibilities is to investigate right-to-left languages. A study of Arabic showed that the OVP falls in the center of the word; it is not shifted to the left as in left-to-right languages [Far96]. Hemi-field studies of the length effect in Hebrew have shown a length effect in both VFs [Lav01b, Lav02c, Naz04b]. Thus right-to-left languages show a different pattern than left-to-right languages, although not an exact reversal. These results suggest that hemispheric specialization provides a constant RVF/LH advantage,

while reading direction provides a RVF/LH advantage for left-to-right languages versus a LVF/RH advantage for right-to-left languages. The sum of these two factors provides a strong RVF/LH advantage for left-to-right languages, and more balanced performance across the VFs for right-to-left languages.

Another way of investigating these issues is to vary hemispheric specialization. Brysbaert [Bry94] identified a group of subjects who did not display the usual LH dominance for language. LH-dominant and non-LH-dominant readers read words of varying lengths, where fixation could fall either on the first or last letter of the word. For the LH-dominant readers, there was a strong RVF (initial letter) advantage, and a strong LVF (final letter) length effect. For the non-LH-dominant readers, the RVF advantage and the LVF length effect were reduced. This indicates that the cost of callosal transfer contributes to the RVF advantage. However, reading direction probably also plays a role, as the results did not completely reverse for the non-LH-dominant readers.

In summary, in left-to-right languages there is a strong asymmetry in the length effect. It is likely that reading direction and hemispheric dominance both play a part in this asymmetry. It is unlikely that differing modes of lexical access contribute to this asymmetry, because we have seen that length effects are not indicative of serial versus parallel processing [New04], the asymmetry is present for fixated words [Ore84, Bry96], and brain imaging indicates that lexical access is left-lateralized independently of presentation location [Coh00, Deh04]. Next I discuss how the SERIOL model explains the asymmetry of the length effect.

## 8.2 SERIOL Account of the Length Effect

Recall that there is an asymmetry of activation patterns across the feature level. For RVF/LH presentation, the locational gradient is smoothly decreasing. For LVF/RH presentation, the second and third letters are strongly inhibited, while letters closer to fixation may not be inhibited enough. Thus the locational gradient is initially steeply decreasing, and then flattens out. For longer words, the locational gradient may not even be monotonically decreasing. (See bold-faced characters in Figure 8.1.) A smoothly decreasing locational gradient is necessary for the optimal encoding of letter order. The LVF/RH locational gradient becomes more and more non-smooth (non-optimal) as string length increases. This increasingly degraded LVF/RH activation pattern then provides an increasingly degraded representation of letter order. A degraded representation of letter order would increase settling time at the lexical level, as activation would be less focused on the target word. Thus an LVF/RH length effect may emerge because letter-position encoding becomes less and less accurate, and settling time increases more and more.

This analysis describes the hypothesized contribution of reading direction to the length effect. In a right-to-left language, acuity gradient inversion would occur in the opposite hemisphere, and thus there would be a non-optimal gradient for the RVF/LH. What then is the contribution of hemispheric dominance? It may be the case that a non-optimal locational gradient in a left-to-right language is further degraded by callosal transfer, increasing its effect. For right-to-left languages, the effect of a non-optimal RVF/LH locational gradient may be reduced. This issue is further discussed in section 8.4. In the following, we will concentrate on left-to-right languages.

This above analysis implies that the length effect should disappear if a smoothly decreasing locational gradient could be created in the LVF. It should be possible to create a smooth gradient via an increase of bottom-up input to the second and third letters (to compensate for lateral inhibition from the first letter). Increasing bottom-up input to those letters should also decrease the activations of the features of the fourth and fifth letters, due to increased left-to-right lateral inhibition. Additionally, for words of more than five letters, a reduction of bottom-up input is probably required at the final letters in order to compensate for their increasing acuity (i.e., to bring their activation levels low enough to make a smooth gradient). (See italic characters in Figure 8.1.)

These adjustments could be accomplished under experimental conditions by increasing contrast at the second and third positions, and reducing contrast at the sixth and higher positions. This leads to the prediction that such a manipulation should cancel the length effect in the LVF/RH via facilitation for the longer strings. That is, for four- to six-letter strings, mean RTs to five- and six-letter strings under this contrast manipulation should be as fast as the mean RT to four-letter strings under normal presentation. Conversely, application of the same pattern in the RVF/LH should create a length effect due to disruption of a previously smooth locational gradient. We tested these predictions in the following experiment [Whi04c].

### **8.3 Length Investigation**

This experiment was designed by me, but run by my colleague Michal Lavidor at the University of Hull, U.K. I specified the overall contrast pattern, while she developed the particular presentation conditions (i.e. background color and letter

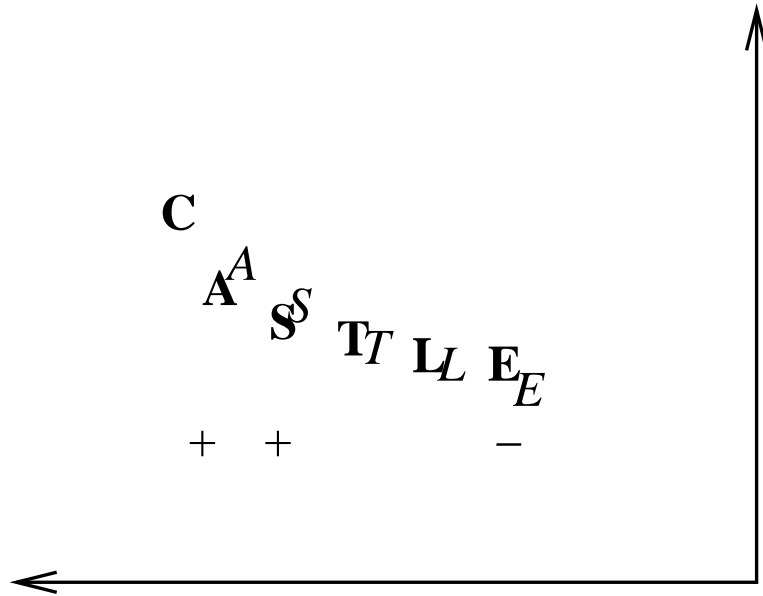


Figure 8.1: Example of proposed LVF/RH locational gradient for normal presentation (bold face) and under contrast manipulation (italics, shifted to the right for clarity) for a six-letter word. Horizontal axis represents retinal location, while vertical axis represents activation level at the feature layer. For normal presentation, the locational gradient is not smooth, becoming quite flat near fixation. Increasing the contrast of the second and third letters raises their activation levels, and decreases the activation levels of the fourth and fifth letters due to increased left-to-right inhibition. Decreasing the contrast of the sixth letter decreases its activation level. As a result, the locational gradient is more smoothly decreasing.



colors).

## Participants

Twenty-three right-handed, native English speakers served as subjects for a lexical decision experiment (mean age 19.7). Ten were males, and 13 were females. All gave their informed consent to participate in the study.

## Stimuli

Ninety-six English content words and 96 nonwords were used, with equal numbers of 4-, 5-, and 6-letter words (32 of each). These three word sets were matched for written word frequency, orthographic neighborhood size, and imageability. Ninety-six nonwords were generated from another word pool by changing one letter, such that the nonwords were legal and pronounceable. Nonwords were also made of 4, 5 and 6 letters in equal proportion. All stimuli were presented in 14-point Helvetica lower-case font on a dark gray background of 3 cd/m<sup>2</sup>. Letters were displayed at three contrast levels: high contrast ( $c=0.64$ ) white letters, medium contrast light-gray letters (high contrast reduced by 40%), and low contrast darker-gray letters (high contrast reduced by 60%). In the control condition, letters at all positions were presented in medium contrast. In the adjust condition, for all string lengths, the first and fourth positions were presented in medium contrast, and the second and third positions in high contrast. For 5- and 6-letter targets, the fifth position was presented in medium contrast. For 6-letter targets, the sixth position was presented in low contrast. Thus, relative to the control condition, the second and third letters were brightened, and the sixth letter (if present) was darkened, while the other positions were presented

at the same contrast level.

## **Design**

Each subject was assigned to one of the 2 versions of the experiment. The different versions rotated the word sets across the presentation conditions (control and adjust conditions in a Latin square design). Each session began with 70 practice trials to introduce the task. Every target stimulus was presented twice, once in each visual field, giving 384 trials for each subject. Stimuli were presented in a random order with the restriction that no more than 3 successive words or nonwords, or 3 successive LVF or RVF trials occurred together. The within-subject factors were lexical status (word or nonword), length (4, 5, or 6 letters), visual field (RVF, LVF), and presentation condition (control or adjust).

## **Procedure**

Each trial began with + appearing in the center of the screen for 400 ms, which then disappeared when the target string was presented. Targets were briefly presented for 180 ms at a displacement of 2.5° from the fixation point to the center of the string. The subject's task was to decide, as quickly and as accurately as possible, whether the stimulus was a legal English word or a nonword. Participants were informed that central fixation was important, and a chinrest together with a head strap were used to ensure stable head position at a distance of 50 cm from screen center. Participants' eye movements were monitored by an infra-red eye tracker, and were recorded for the first 700 ms of each trial.

## Results

Trials in which gaze did not remain stable on the fixation cross were discarded (3% of word trials; 5.1% of nonword trials). RTs of less than 200 ms and more than 1100 ms were also discarded either as anticipatory or excessively lengthy (discarded trials occurred infrequently, less than 3% of the total). Mean RTs and error rates are given in Tables 8.1 and 8.2. Repeated measures ANOVAs on RTs (separated for words and nonwords) revealed that visual field had a significant effect ( $F(1,22)=47.3$ ,  $p<0.00001$  for words, ns for nonwords), with RVF words (mean RT = 502 ms) responded to faster than LVF words (mean RT = 545 ms). String length was significant (words:  $F(2,44)=19.8$ ,  $p<0.001$ ; nonwords:  $F(2,44)=3.53$ ,  $p<0.05$ ), with longer latencies to longer strings. The main effect of presentation condition was not significant.

Presentation condition and visual field interacted ( $F(1,22)=22.7$ ,  $p<0.001$  for words;  $F(1,22)=8.0$ ,  $p<0.05$  for nonwords). This interaction was analyzed using a simple main effects analysis. For LVF stimuli, the adjust condition was faster than the control condition ( $F(1,22)=6.76$ ,  $p<0.05$ ); For RVF stimuli, the opposite pattern was found ( $F(1,22)=5.33$ ,  $p<0.05$ ). No interaction was found for presentation condition and length, nor for visual field and length. The interaction between presentation condition, visual field, and word length was significant for word stimuli ( $F(2,44)=16.84$ ,  $p<0.001$ ; ns for nonwords). The triple interaction was analyzed using a simple main effects analysis. For LVF words, a length effect occurred only under the control condition ( $F(2,44)=7.91$ ,  $p<0.01$ ). For RVF words, a length effect occurred only under the adjust condition ( $F(2,44)=8.14$ ,  $p<0.01$ ). This pattern is clearly shown in Figure 8.2.

The pattern for nonwords was similar, but the three-way interaction did not

	LVF con.	LVF Adj.	RVF con.	RVF adj.
<i>Four</i>				
Mean RT	527	536	487	474
S.D.	71	69	68	66
% error	15	18	12	7
<i>Five</i>				
Mean RT	563	518	477	536
S.D.	70	67	70	72
% error	13	10	12	7
<i>Six</i>				
Mean RT	594	535	490	548
S.D.	80	71	77	71
% error	14	14	10	11

Table 8.1: Results for word targets.

	LVF Con.	LVF Adj.	RVF Con.	RVF Adj.
<i>Four</i>				
Mean RT	561	560	573	587
S.D.	88	82	82	89
% error	11	11	11	21
<i>Five</i>				
Mean RT	613	582	572	617
S.D.	100	87	83	79
% error	13	11	10	14
<i>Six</i>				
Mean RT	653	561	596	630
S.D.	88	93	84	89
% error	9	10	18	14

Table 8.2: Results for non-word targets.

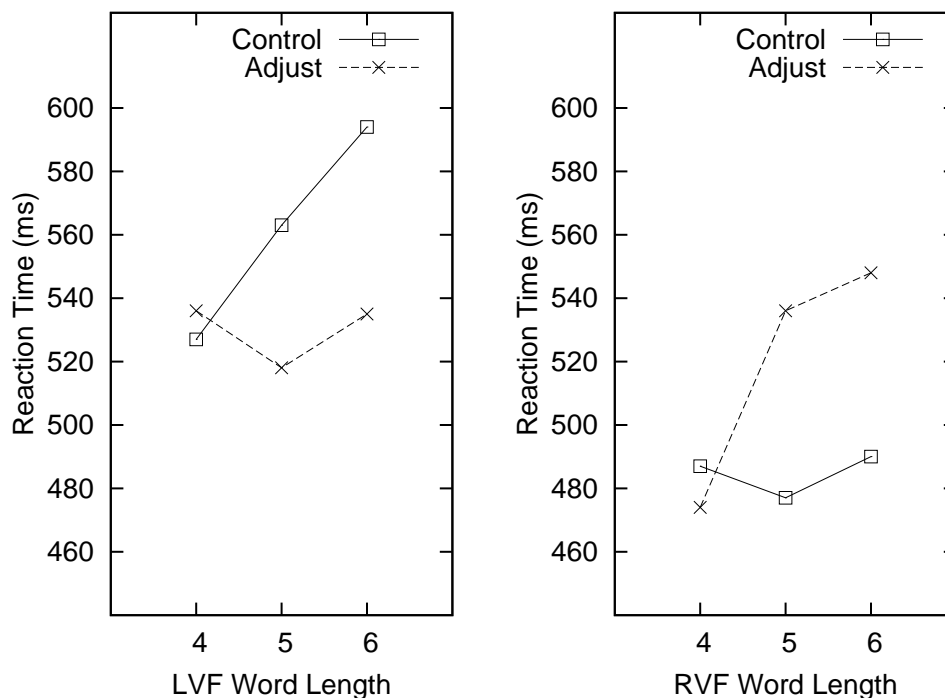


Figure 8.2: Results for word targets.

reach significance. Average error rate was 12%, and no significant effects of visual field, length, or presentation condition were found.

## 8.4 Discussion

As predicted, the LVF/RH length effect was eliminated under the adjust condition. It cannot be argued that the effect was still present, though masked. Five- and six-letter words under the adjust condition were processed as quickly as four-letter words under the control condition, demonstrating that the length effect was completely neutralized. This conclusively demonstrates that a length effect is not inherent feature of RH processing, for if it were, it would not be possible to eliminate it via a visual manipulation. Therefore, the LVF length effect does not arise from an RH-specific mode of lexical access, disproving the

dual-modes theory [Ell88].

Since we were able to abolish the length effect via an activation-pattern correction, this indicates that the LVF activation pattern is a contributing factor to the length effect. The appropriate contrast manipulations to neutralize the length effect were precisely predicted from the theory of locational-gradient formation, providing strong support for this aspect of the SERIOL model. We suggest that locational gradient formation provides a mechanistic account of the perceptual learning espoused by Nazir [Naz03, Naz04a].

Also in line with our predictions, a length effect was created in the RVF/LH. While it may not be surprising that increased RTs were associated with the degradation of the sixth letter in the RVF adjust condition (since it was far from fixation), we note that most of this increase was present for five-letter strings. For these strings, the only change from the control condition was positional contrast enhancement at the second and third letters. Yet, this enhancement was inhibitory in the RVF. It is unlikely that the inhibition arose solely because this enhancement reduced the visibility of nearby letters, because this manipulation had no effect on error rates or on RT to four-letter words, although the possibility that the low-acuity fourth letter was affected only when it was not the last letter cannot be ruled out. Nevertheless, the RVF adjust-condition results are consistent with our predictions. The adjust condition had no effect on four-letter words, relative to their respective control conditions. However, it might be expected that RVF RT should increase due to a degraded locational gradient, and LVF RT should decrease due to an improved locational gradient. So why did the contrast manipulation have no effect on four-letter words? It may be the case that settling time is relatively insensitive to small differences in activation patterns for shorter words, due to the large number of competitors.

Further investigations into the length effect will involve languages read from right to left, such as Hebrew. For such languages, the locational gradient should decrease from right to left. Thus, the consistency of the acuity gradient with respect to the locational gradient is reversed. That is, the acuity gradient matches the locational gradient in the LVF/RH, not the RVF/LH. This suggests that the length effect should reverse. However, experimental studies have given conflicting results. One has shown a length effect for both visual fields [Lav01b]. One has shown the predicted reversal [Naz04b], while another has shown the same pattern as left-to-right languages [Lav02c]. Overall, these results suggest that the robust asymmetry observed for left-to-right languages is not present for Hebrew, where a length effect seems to occur in both visual fields. Based on these findings, I proposed that callosal transfer to the dominant hemisphere also contributes to the length effect by preferentially degrading more lowly activated letter features [Whi04a]. In the case of a left-to-right language, this further reduces the feature-level activations of the second and third letters. In the case of a right-to-left language, this reduces feature-level activations of the final letters, thereby delaying their firing at the letter layer, creating a length effect. Thus, it should be possible to cancel the Hebrew LVF/RH length effect by using a different experimental manipulation than in a left-to-right language - namely, by increasing bottom-up input in proportion to distance from fixation. In contrast, the same type of manipulation as in English should cancel the Hebrew RVF/LH length effect.



## Chapter 9

### Asymmetry of the N effect

The effect of another lexical property, orthographic neighborhood size (N), also interacts with VF. N is the number words which can be formed by replacing one letter of the target word [Col77]. For example, CARE has a large neighborhood - BARE, DARE, CORE, CURE, CAME, CAGE, CART, CARD, etc. First I present experimental data on the N effect. Then I discuss the SERIOL account of the N effect, and present two experiments testing this account.

#### 9.1 The N effect

Under central presentation in a lexical-decision task, low-frequency words with large neighborhoods are responded to more quickly than those with small neighborhoods [And89, And97]. It is surprising that the N effect manifests as facilitation, because lateral inhibition within the word level is commonly assumed. Therefore, increased similarity to other words should increase inhibition to the target and slow down RTs, rather than speed them. Thus, there must be some facilitatory effect that arises in spite of lateral inhibition.

Several explanations for the locus of this unexpected facilitation have been proposed. It could arise at the letter level, as in the Interactive Activation model

[McC81]. In this scenario, excitation from the word level feeds back to the letter level, and then forward to the word level. Thus increased similarity to words increases the amount excitatory feedback to the letter level, which then allows the target word to reach response criterion more quickly. Alternatively, the facilitation could arise solely within the word level. In their multiple read-out model [Gra96], Grainger and Jacobs have proposed that increased activation across the word-level speeds a task that does not require the unique identification of a single word, such as lexical decision.

Another possible locus is the phonological level, either through general feedback to the target word or specifically via word bodies [Zie98]. The word-body hypothesis was tested in a series of lexical-decision experiments [Zie98]. In one set of words, N was held constant while the number of words matching the target's body (body neighbors, denoted BN) was varied. (A body neighbor does not have to be of the same length as the target.) In another set of words, BN was held constant while N was varied. In the BN manipulation, high BN was facilitatory (as compared to low BN). However, in the N manipulation, high N had no effect. Thus facilitation depended on a large number of body neighbors, not N-metric neighbors. Since BN and N are usually highly correlated, these results suggest that the standard N effect arises from body neighbors.

The same manipulations were also performed for non-words. For such targets, high N has an inhibitory effect, as increased similarity to real words makes it more difficult to reject a target. However, the BN manipulation did not produce an inhibitory effect for high BN targets. In contrast, the N manipulation did produce the standard inhibitory N effect. These results cast doubt on the phonological interpretation of the facilitatory effect of BN on word targets, because increased phonological similarity of non-words to real words (for high-BN targets) should

have slowed RTs.

Investigation into the N effect has recently been extended to lateralized presentation. These experiments demonstrated that the N effect is present for LVF, but not RVF, presentation [Lav02a, Lav02b]. Thus, for the N effect, central presentation patterns with the LVF, not the RVF. Therefore, it cannot be the case that the LVF/RVF difference occurs simply because LVF stimuli are less efficiently processed than RVF stimuli, because the N effect occurs for central presentation, where stimuli are the most efficiently processed. (This pattern has been shown within a single set of stimuli [Lav02b].)

## **9.2 The SERIOL Account of the N effect**

This asymmetry makes it unlikely that the facilitatory N effect is due to phonological influences or to total word-level activation, as it is unclear why those factors would vary with visual field. A more likely candidate is word-to-letter feedback, as we have already discussed that letter-level activation patterns vary with VF. Although the SERIOL model focuses on the bottom-up processing stream, I do not mean to rule out top-down activation from the word level back to lower levels. The oscillatory cycle driving the letter level is taken to fall in the theta band (5 - 8 Hz) [Lis95, Kli96, Kli01]. Thus, an individual cycle would take 125 to 200 ms, allowing more than one cycle to occur during lexical decision. Input to the letter level is necessarily bottom-up during the first oscillatory cycle. On subsequent cycles, input to letter nodes could arise from both bottom-up and top-down sources. It is assumed that top-down input from the word to the letter level is also in the form of a gradient, where the first letter receives the most input, the second letter the next most, etc. Such a gradient would be instrumental in

serial output of letters when spelling.

I propose that the hemispheric asymmetry of the N effect arises from the formation of the locational gradient, coupled with the processing which converts the locational gradient into a serial firing pattern. Due to these dynamics, top-down input to the letter level (from high N) has a facilitatory effect for LVF/RH presentation, but not for RVF/LH presentation.

First I focus on the dynamics of the conversion of the spatial gradient to serial firing at the letter level. The point at which a non-initial letter node can start to fire is limited both by lateral inhibition from the prior letter, and by its own level of excitatory input. When the firing rate of the currently active letter node exceeds a certain level, no other letter node can fire, due to the constant lateral inhibition. At some point, the current letter's firing rate and the resulting lateral inhibition will decrease to a level which would allow the next letter to fire. If the next letter currently receives enough excitatory input to cross threshold at this point, it can fire. In this case, lateral inhibition from the active letter was the limiting factor on when the next letter could start to fire. However, if the next letter does not receive enough excitatory input to fire immediately, its activation is delayed until its excitability increases enough (via the oscillatory cycle) for it to cross firing threshold. In this case, the limiting factor was the amount of excitatory input.

In the following, I will focus on four-letter stimuli, as most N experiments are performed on stimuli of that length. Recall that in the feature level of the LVF/RH, the second and third letters receive strong lateral inhibition, whereas the second and third letters in the RVF/LH do not. For central presentation, the second letter receives strong inhibition in the LVF/RH, and the third letter receives strong cross-hemispheric inhibition. Based on this difference in activation

patterns and the above dynamics, I propose that the lower level of bottom-up input (to the letter level) to the second and third letters is the primary locus of the N effect, as follows. For LVF and central presentation, the activations of the second and third letters are limited by their level of excitatory input. Therefore, a slight increase in excitation (due to feedback from the word level from high N) allows those letter nodes to cross threshold and fire sooner. In contrast, for RVF presentation, those letter nodes receive a relatively higher level of bottom-up input. Their firing is limited by lateral inhibition, rather than excitatory input. Thus, the second and third letter nodes already fire as early as possible, and a slight increase in excitatory input has no effect. So top-down excitation allows the internal letter nodes to fire earlier for LVF/RH and central presentation, but not RVF/LH presentation.

When the second and third letter nodes fire earlier, the corresponding bigrams are activated earlier. This then allows activation to begin to be focused on the target word node earlier, reducing lateral inhibition from other word nodes. For example, consider the stimulus *bore*. When \*B fires, two-letter words starting with B are the most highly activated (due to the higher connection weights for shorter words). These word nodes inhibit less highly activated word nodes (including the target BORE). When BO fires, three-letter words starting with BO are the most highly activated, and inhibit the other word nodes. When OR and BR fire, four-letter word nodes starting with BOR are the most highly activated. Finally, the target BORE is no longer inhibited by other more highly activated word nodes (although if BOR were itself a word, there would still be a more active word node). Thus the sooner that the activation becomes focused on the target, the less lateral inhibition there is from other word nodes. This decreased lateral inhibition over the course of the oscillatory cycle will allow the target word node

to reach response criterion sooner, decreasing RT.<sup>1</sup> Thus top-down excitation from high N decreases RTs for LVF and central presentation. For RVF presentation, the second and third letter nodes already fire as early as possible, so there is no N effect.

For LVF presentation, another factor may also be at work. Recall that the activation level of non-contiguous bigrams is determined by the amount of time between the firing of the first constituent letter and the firing of the second constituent letter. This time lag is determined by the relative levels of feature-layer inputs to those letter nodes, which are determined by the locational gradient. Bigram-to-word connection weights are based on the bigram activation pattern resulting from a smoothly decreasing locational gradient. When the locational gradient is not smoothly decreasing (as in the LVF/RH), a somewhat different bigram activation pattern results. Thus, there is a mismatch between the bigram activation vector and the learned weight vector, making activation less focused on the target word. Top-down input from high N may compensate for the lack of smoothness of the locational gradient, bringing the bigram activation vector nearer the learned weight vector. This could also contribute to the N effect for LVF presentation.

---

<sup>1</sup>This account is revised from the original account, which focused on increased activation levels for the second and third letters, which were passed onto the bigram and word levels. Given the new assumption that bigram activations do not reflect letter activations, this account has been modified to focus on timing of firing. However, the underlying idea remains the same. There is an N effect for central and LVF/RH presentation because the locational gradient is steeper than for RVF/LH presentation.

### 9.3 Predictions

In this experiment, we concentrated on the asymmetry of the N effect under lateralized presentation. Because the proposal is that differences in bottom-up activation patterns underlie this asymmetry, changes to these activation patterns should modulate the N effect. If the LVF/RH activation pattern could be created in the RVF/LH, the N effect should appear in the RVF/LH. Conversely, if the RVF/LH activation pattern could be created in the LVF/RH, the N effect should disappear in the LVF/RH. It should be possible to adjust activation patterns by manipulating contrast levels at specific string positions. The RVF/LH's feature-level activation pattern could be replicated in the LVF/RH by slightly dimming the external letters. Dimming the first letter should decrease lateral inhibition from that letter, mimicking the weaker left-to-right inhibition in the LH. Dimming the final letter should compensate for the increasing acuity, creating a more smoothly decreasing gradient. As a result, the locational gradient should be smoother and shallower, mimicking the usual activation pattern in the RVF/LH. (See Figure 9.1.) This should negate the N effect. Conversely, the LVF/RH's activation pattern could be mimicked in the RVF/LH by slightly dimming the internal letters. This should induce the N effect in the RVF.

To test these predictions, we performed a lateralized lexical-decision experiment of low-N versus high-N words, with two different patterns of dimmed input, in addition to the control (undimmed) condition [Whi04b]. All stimuli were four-letter words. In the inner-dimmed condition, the contrast of the second and third letters was reduced. In the outer-dimmed condition, the contrast of the first and fourth letters was reduced. The analysis of the N effect allows precise predictions concerning the expected effects of these manipulations.

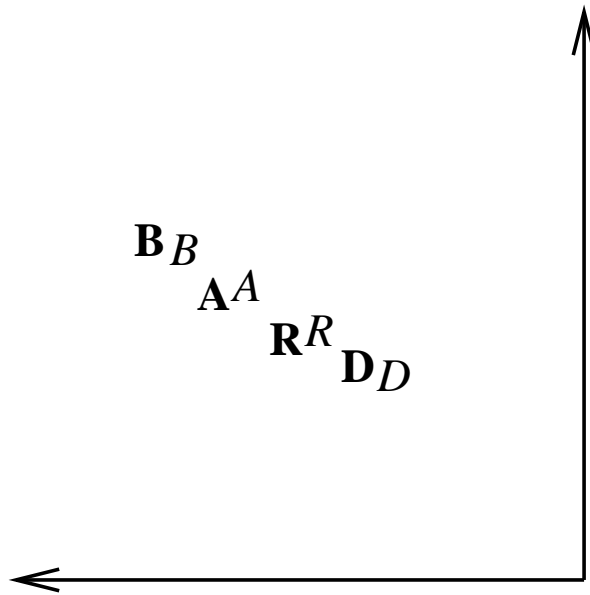


Figure 9.1: Outer dimming in the LVF/RH. The normal locational gradient is shown in bold-face. The results of outer dimming are shown in italics (shifted to the right for clarity.) Reducing the contrast of the first letter reduces its activation level, and decreases inhibition to the second and third letters, increasing their activation levels. As a result, the locational gradient is shallower across the first three letters. Reducing the contrast of the fourth letter reduces its activation level. As a result, the locational gradient is smoother across the last three letters.



Let  $R$  be the RT for the control / RVF / low-N condition,  $L$  be the additional time cost of presentation to the non-dominant hemisphere, and  $Z$  be the cost of low input to the second and third letters. Thus, the expected RTs for the other control conditions are:

$$\text{control/RVF/high-N} = R \quad \text{control/LVF/high-N} = R + L \quad \text{control/LVF/low-N} = R + L + Z$$

First outer dimming is considered. There is little direct cost for reducing input to the external letters, because their activations remain relatively high. Therefore, in the RVF, outer dimming should have little effect, giving:

$$\text{outer/RVF/high-N} = R \quad \text{outer/RVF/low-N} = R$$

In the LVF, outer dimming should compensate for the normal cost of low input to the second and third letters. Therefore:

$$\text{outer/LVF/high-N} = R + L \quad \text{outer/LVF/low-N} = R + L$$

Note the counterintuitive prediction that such stimulus degradation should produce *facilitation* for low-N (relative to the undimmed control). As a result, there should be no N effect.

Next inner dimming is considered. In the RVF, this should induce a cost at the internal letters for low-N. However, top-down activation from high-N should compensate for this decreased bottom-up input. Thus:

$$\text{inner/RVF/high-N} = R \quad \text{inner/RVF/low-N} = R + L$$

Therefore, an N effect should be created. In the LVF, inner dimming should not change the overall bottom-up activation pattern, although it could potentially increase the cost of low activation at the internal letters. We would expect the size of the N effect to stay the same or get larger.

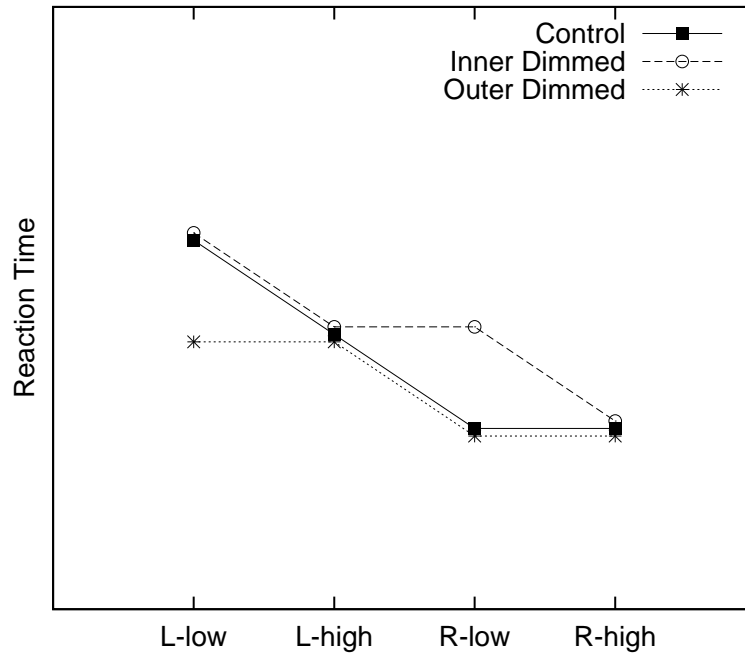


Figure 9.2: Predicted pattern for Experiment 2.

In summary, the predictions are that outer dimming should decrease RTs for the LVF / low-N condition (giving no N effect), and that inner dimming should increase RTs for the RVF / high-N condition (giving an N effect). Inner dimming might also increase RTs for the LVF conditions. Other manipulations should have little effect. See Figure 9.2 for a graphical presentation of these predictions, under the simplest assumptions - that inner dimming incurs no additional cost in the LVF, and that  $L$  and  $Z$  are of the same magnitude. The latter assumption is consistent with the results of [Lav02a], in which both were on the order of 30 ms.

## 9.4 N-effect Investigation 1

This experiment was designed by me, but run by my colleague Michal Lavidor at the University of Hull, U.K. I specified the overall contrast patterns, while she developed the particular presentation conditions (i.e. background color and letter colors).

### Participants

Nineteen native English speakers participated in the experiment. All had normal or corrected-to-normal vision and were aged 18-26 (mean age 19.4, s.d. 1.6). Each participant received either a course credit or £2. All participants were right-handed and scored at least 80 on the Edinburgh test. Nine were males, 10 females.

### Design and materials

*Stimuli.* The word stimuli were 78 4-letter, English content words (nouns and verbs). Half of the words had fewer than 10 orthographic neighbors (mean no. of neighbors 6.2). These words formed the low-N group. The remaining words all had more than 12 neighbors (mean 17.0). These formed the high-N group. The low-N and high-N groups were matched on written frequency, imageability, and age of acquisition. Each group was divided into 3 sets, to allow rotation through the 3 different presentation conditions (control, inner-dimmed, or outer-dimmed). These 6 sets were also matched for written word frequency, imageability, and age of acquisition. The stimuli are given in Table 9.1.

Since the model we tested focuses on words, the non-words were created such that they would amplify the N effect for words (based on [Sik02]). The non-

Low 1	Low 2	Low 3	High 1	High 2	High 3
beau	arch	babe	bush	beam	bite
cube	aunt	coal	cage	bolt	boot
earl	chop	crab	cone	deer	cake
germ	disc	gasp	dent	duck	cart
gulf	duel	grip	dusk	dump	dock
heap	fork	jerk	hank	gang	hail
howl	lamb	lens	herd	gore	hint
newt	menu	liar	hind	hose	hush
oath	omen	oven	hump	lime	joke
palm	plug	raid	mall	maze	leak
shed	prey	riot	mule	pump	mist
soap	roar	sand	nail	rent	port
swim	suds	sigh	rust	rope	rake

Table 9.1: Stimuli for N-effect investigations.

words were generated from a different pool of 4-letter words by altering one or two letters, usually replacing the vowels with consonants (however bigrams were always orthographically legal). There was no special effort to match N size of the non-words as they served mainly as the context for the words; however to keep chance performance at 50% level we presented the non-words at the same illumination conditions as the real words.

All stimuli were presented in 14-point Helvetica lower-case font, appearing as high contrast ( $c=0.72$ ) white letters on a gray background of 4 cd/m<sup>2</sup>. In the inner-dimmed condition, light-gray patches were projected on the 2nd and 3rd letters of the presented target, so the contrast between the letter and the background color was decreased by 33%; thus these letters were dimmer than the rest of the word. Similarly, two light-gray patches dimmed the 1st and 4th letters in the outer-dimmed condition. In the control condition, no letters were dimmed. The stimuli were presented for 180 ms, at a displacement of 2.5° from the fixation point to the center of the stimulus. The displacement was to the left or to the right of a central focus point (LVF and RVF, respectively).

*Design.* Each subject was assigned to one of the 3 versions of the experiment. The different versions rotated the word sets across the experimental conditions (high- and low-N words in control, inner-dimmed, and outer-dimmed conditions). Each target stimuli was presented once to each visual field. The within-subject factors for words were N size (high, low), visual field (RVF, LVF) and presentation condition (control, inner-dimmed or outer-dimmed). Each combination of the within-subject variables was repeated 13 times.

## Procedure

Stimulus presentation was controlled by an IBM Pentium computer on 17" SVGA display. The participants sat at a viewing distance of 50 cm, with the head positioned in a chin rest. The experiment was designed using Super-Lab version 2.

Each session began with 10 practice trials to introduce the task, followed by 24 additional practice trials of centrally presented letter strings, where the task was to perform lexical decision. Thirty-six additional practice trials presented words and non-words either to the left or to the right of the fixation point. Each trial began with + appearing in the center of the screen for 400 ms. For the first trial, the + remained for 2000 ms, and disappeared when the target word was presented. The + would again reappear to allow projection of the next target word. Targets were briefly presented for 180 ms (either a word or a non-word), to the left or to the right of the focus point. The participant's task was to decide, as quickly and as accurately as possible, whether the stimulus was a legal English word or a non-word. Participants responded by pressing one of two available response keys, labeled 'word' and 'non-word' on a standard 'QWERTY' keyboard. For half of the participants, the response 'word' was made by pressing the 'N' key, and 'non-word' by pressing the 'V' key. For the other half, the response keys were reversed. The participants were randomly assigned to one of the two response options.

## Results

Since the main manipulation of orthographic neighborhood was designed for the word stimuli, the repeated measures analysis with N (high, low), visual field

	L-low	L-high	R-low	R-high
<i>control</i>				
mean RT	620	595	569	566
S.D.	72	70	67	66
% error	19	15	18	18
<i>inner-dim</i>				
mean RT	611	590	590	569
S.D.	72	69	73	69
% error	20	17	18	15
<i>outer-dim</i>				
mean RT	592	598	555	558
S.D.	70	84	80	75
% error	14	20	11	15

Table 9.2: Results for N-effect investigation 1.

(right, left) and presentation condition (control, inner-dimmed or outer-dimmed) as the within-subjects variables were conducted only for words. RTs of less than 150ms and more than 1400ms were discarded either as anticipatory or excessively lengthy (discarded trials occurred infrequently, less than 2% of the total). Mean RTs for correct responses are summarized in Table 9.2, and presented graphically in Figure 9.3.

*Reaction times.* Visual field had a significant effect [ $F(1,18)=7.2$ ,  $p<0.05$ ;  $F(1,24)=6.4$ ,  $p<0.05$ ], with RVF words (mean RT = 567 ms) responded to faster than LVF words (mean RT = 601 ms). Presentation type and neighborhood size interacted [ $F(2,36)=4.18$ ,  $p<0.05$ ;  $F(2)$  not significant]. We examined the simple effects of N for each visual condition separately and found that the N effect was

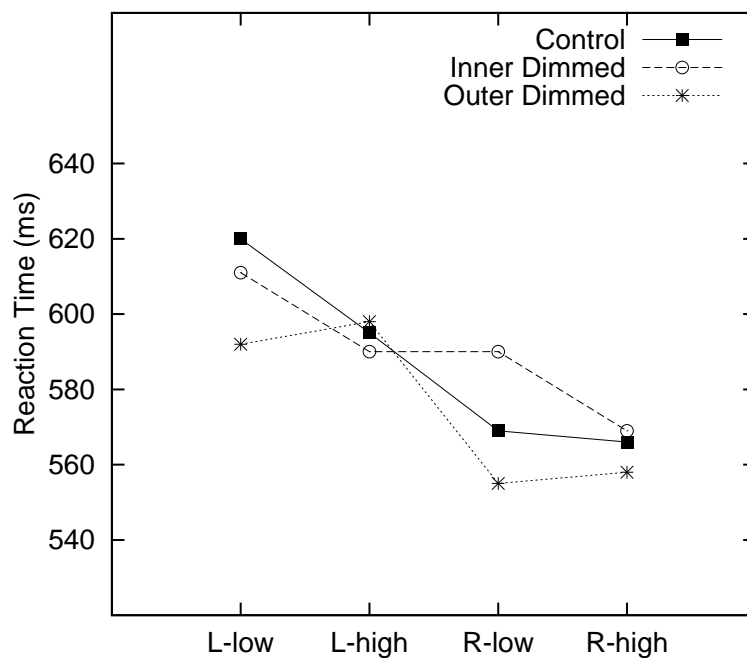


Figure 9.3: Results for N-effect investigation 1.

significant both in the control condition [ $F(1,18)=5.9$ ,  $p<0.05$ ] and the inner-dimmed condition [ $F(1,18)=8.2$ ,  $p<0.05$ ], but not the outer-dimmed condition.

The interaction between presentation type, visual field, and orthographic neighborhood size was also significant [ $F_1(2,36)=6.3$ ,  $p<0.01$ ;  $F_2(2,48)=6.0$ ,  $p<0.01$ ]. Post hoc Bonferroni ( $p<0.05$ ) comparisons yielded that for LVF words, the N effect occurred under both the control and inner-dimmed conditions, but not the outer-dimmed condition. For RVF words, the N effect emerged only under the inner-dimmed condition.

*Error rates.* Average error rate was 16%, and the patterns were similar to the RT data. However, no significant effects of visual field, N size, or presentation condition were found (see mean error rates in Table 9.2).



## Discussion

The hemispheric specificity of the N effect was replicated for the control conditions, with faster RTs to high-N than low-N words in the LVF/RH, but not the RVF/LH. In the LVF/RH, dimming the outer letters negated the N-effect, via facilitation (relative to the control condition) for low-N, but not high-N. In the RVF/LH, outer dimming had no effect. In the RVF/LH, dimming the inner letters created the N effect via inhibition for low-N, but not high-N. In the LVF/RH, inner dimming had no effect. A comparison of Figures 9.2 and 9.3 shows that the experimental results closely match the predicted pattern.

### 9.5 Further Predictions

The previous experiment showed the predicted patterns for lateralized presentation. It should be possible to also negate the N effect for central (CVF) presentation via a contrast manipulation. However, a different manipulation may be required, due to the differing shapes of the locational gradient in the CVF and the LVF. In the LVF, input to the fourth letter may be too high with respect to the third letter (due to incomplete inversion of the acuity gradient). However, this would not be the case for the CVF, where the locational gradient across the third and fourth letters is determined by the steeply decreasing RVF/LH acuity gradient. In the CVF, input to the fourth letter would not be too high, and dimming that letter may not be beneficial. Therefore, we initially ran a pilot study to determine what manipulation would negate the CVF N effect. This study indicated that dimming both outer letters did not remove the effect, while dimming only the first letter did. This is consistent with the proposal that input to the fourth letter is relatively too high for the LVF, but not the CVF.

In the following experiment, we sought to negate the N effect for both LVF and CVF presentation within a single study. In the *dimmed condition*, the outer two letters were dimmed for LVF and RVF presentation, while only the first letter was dimmed for CVF presentation. The respective control conditions remained the same as in the previous experiment. For the dimmed condition, we expected to negate the CVF N effect (by facilitating responses to low-N words), and to replicate the results from the outer-dimmed conditions in the previous experiment.

## 9.6 N-effect Investigation 2

This experiment was designed by me, but run by my colleague Michal Lavidor at the University of Hull, U.K. It used the particular presentation conditions (i.e. background color and letter colors) that she developed for the previous experiment.

### Participants

Twenty-five native English speakers participated in the experiment. All had normal or corrected-to-normal vision and were aged 18-28 (mean age 19.6, s.d. 1.9). Each participant received either a course credit or £2. All participants were right-handed and scored at least 80 on the Edinburgh test. Eleven were males, 14 females.

### Design and Materials

*Stimuli.* The same stimuli as in the previous experiment were used (see Table 9.1).

*Design.* Each subject was assigned to one of the 3 versions of the experiment. The different versions rotated the word sets across the experimental conditions. The within-subject factors for words were N size (high, low), visual field (RVF, LVF or center) and presentation condition (control or dimmed). Each combination of the within-subject variables was repeated 13 times. The dimmed condition included dimming of the two external letters for the RVF and LVF presentations, and dimming of the first letter for the centrally-presented stimuli.

*Procedure.* The procedure was similar to the procedure of the previous experiment.

## Results

Since the main manipulation of orthographic neighborhood was designed for the word stimuli, the repeated measures analysis with N (high, low), visual field (right, left, center) and presentation condition (control or outer-dimmed) as the within-subjects variables were conducted only for words. The results of one participant were not included in the analysis due to low-accuracy performance (below chance level). RTs of less than 150 ms and more than 1400 ms were discarded either as anticipatory or excessively lengthy (discarded trials occurred infrequently, about 2% of the total). Mean RTs for correct responses are presented in Table 9.3 and Figure 9.4.

*Reaction times.* Visual field had a significant effect [ $F(2,46)=10.3$ ,  $p<0.01$ ;  $F(2,24)=8.1$ ,  $p<0.01$ ]. Centrally presented words (mean RT = 478 ms) yielded fastest responses, followed by RVF words (mean RT = 536 ms), then LVF words (mean RT = 567 ms). Post-hoc differences were analyzed employing Bonferroni comparisons ( $p<0.05$ ).

The interaction between presentation type, visual field, and orthographic

	L-few	L-many	C-few	C-many	R-few	R-many
<i>control</i>						
mean RT	582	560	495	472	537	540
S.D.	59	65	57	60	71	66
% error	16	12	10	8	13	11
<i>dimmed</i>						
mean RT	558	570	471	479	536	533
S.D.	63	52	58	55	59	60
% error	14	15	9	10	13	11

Table 9.3: Results for N-effect investigation 2. In the dimmed condition, the outer two letters were dimmed for RVF and LVF presentation, while only the first letter was dimmed for CVF presentation.

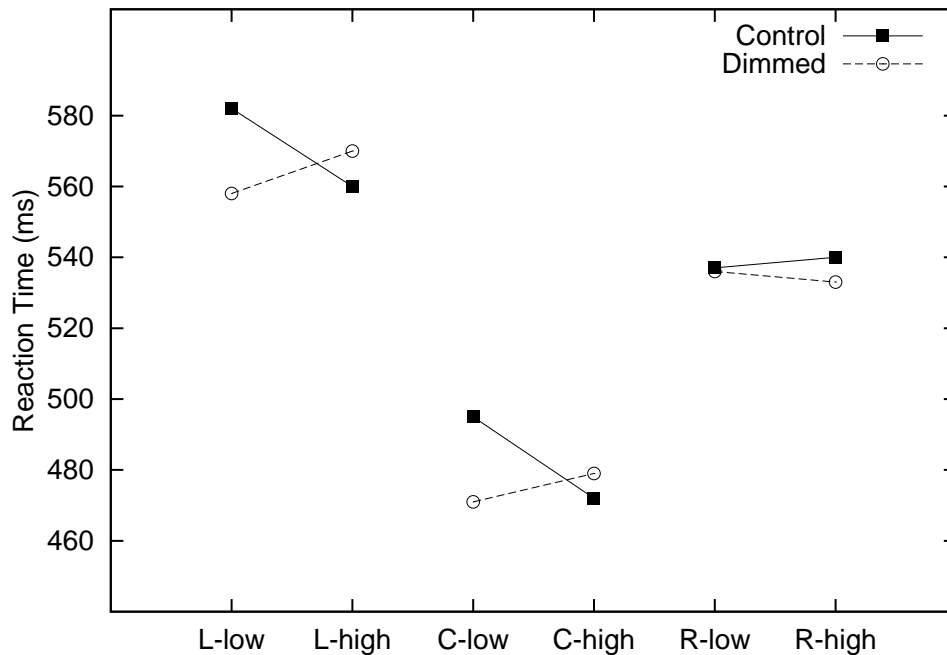


Figure 9.4: Results for N-effect investigation 2.

neighborhood size was also significant [ $F(2,46)=5.8, p<0.01$ ;  $F(2,48)=4.9, p<0.05$ ]. Post hoc Bonferroni ( $p<0.05$ ) comparisons yielded that for LVF and CVF words, the N effect occurred for the control condition, but not the dimmed condition. For RVF words, there was no N effect in either condition.

*Error rates.* Average error rate was 11%, and the patterns were similar to the RT data. However, no significant effects of visual field, N size, or presentation condition were found.

## Discussion

In the control condition, N effects for the CVF and LVF, but not the RVF, were replicated. The dimmed condition for the LVF and RVF (wherein the outer letters were adjusted) replicated the results from the previous experiment - the LVF N effect was negated via facilitation for low-N, while dimming had no effect in the RVF. Crucially, the CVF dimmed condition (wherein only the first letter was adjusted) negated the N effect, via facilitation for low-N, but not high-N. Thus the predicted results were achieved.

## 9.7 Implications

Experiments 2 and 3 showed that it is possible to create or negate the N effect by altering bottom-up activation patterns via contrast manipulations, as predicted by the SERIOL account. Note that a simpler explanation of these results does not suffice. It cannot be the case that dimming the outer letters was facilitatory for LVF / low-N words simply because the internal letters were unmasked at a very low level. In that case, there should have been a similar effect in the RVF, yet none was found. It could not be the case that such RVF

facilitation did not occur simply because the stimuli were less degraded than in the LVF, since we demonstrated a facilitation in the least degraded location, the CVF. Moreover, the creation of an RVF N-effect by dimming the internal letters indicates that the reason that such an effect does not usually occur is that those letters are usually more highly activated. This places the locus of the VF x N-effect interaction squarely at the level of hemisphere-specific, orthographic activation patterns. The SERIOL model explains the source and nature of these patterns.

## **Locus of the N effect in Lexical Decision**

The fact that manipulations of contrast modulated the N effect indicates that its primary locus is the letter level. Other accounts of the N-effect based on word-level activations [Gra96] or phonological representations [Zie98] cannot explain the demonstrated effects of manipulating the visual properties of letters.

Andrews [And97] noted that the N effect appears less strong in French and Spanish. The conclusion that feedback excitation to the letter level is the primary source of the N effect can potentially account for such a linguistic difference. Under the assumption that the reading lexicon also provides the spelling lexicon [Bur02], spelling could be represented by connections from a word node back to the letter nodes. In languages with shallower orthographies than English, such as Spanish and French, it is less necessary to encode spelling via word-to-letter connections, since spelling is predictable from phonology. Therefore, word-to-letter connections may be weaker in such languages. These weaker top-down connections would then account for the reduced influence of N in these languages.

However, others have argued against such a letter-level locus [Bro93, Rey04], based on the absence of an interaction between stimulus quality and word fre-

quency in lexical decision [Sta75, Bro93, Bal95]. That is, when letter contrast is uniformly low, the cost of this degradation does not vary with the frequency of the target word. If there were feedback from the word level to the letter level, this should cause an interaction between stimulus quality and a lexical attribute, such as frequency. The lack of such an interaction has been taken as indicating that processing is staged, rather than interactive. That is, computations are completed at the letter level before being passed on to the word level, as opposed to a continuous interaction between levels.

However, this finding is not inconsistent with the model, or experimental results. Note that the SERIOL model is not a fully interactive; letter activations only occur at specific time intervals. Although I have not fully specified all the timing relationships between levels, the implicit assumption is that there is gating between the feature and letter levels. The induction of the correct firing order at the letter level depends on the proper activation pattern at the feature level. Thus, the feature level must settle into this pattern before it activates the letter level. If the letter nodes were activated while the feature level were still settling, the wrong firing pattern would result. Moreover, feature-level input must be passed to the letter level at the start of an oscillatory cycle. Therefore there has to be some co-ordination between the feature and letter levels, so that feature level activation affects the letter level at the right time. Thus we assume a staged activation. So, the effects of uniformly low stimulus quality may be resolved before the feature level is allowed to activate the letter level, consistent with the lack of interaction between overall stimulus quality and frequency.

However, this does not rule out the possibility of feedback from the word level affecting the letter level at a later point in processing. For example, such feedback might occur during the down-phase of the oscillatory cycle. Under this

scenario, word-level activation would not affect the letter level until the end of the oscillatory cycle. This feedback would then have an effect on letter activations during the next oscillatory cycle. Such feedback would not interact with overall effects of stimulus quality, which have been resolved prior to activation of the letter level. However, this feedback would interact with the resulting activation pattern passed forward from the feature level. That is, overall low stimulus quality may have a large inhibitory effect the first time that the feature level activates the letter level, and this effect may dominate as compared to any later top-down effects.

We have demonstrated an interaction between the N effect and positional manipulations of letter contrast. Thus we have demonstrated an interaction between a lexical attribute and stimulus quality, indicating that feedback from the word to letter level does occur, and is the primary source of the N effect in lexical decision.

## **Orthographic Similarity**

The proposal that the internal letters are the primary source of the N effect implies that the position of difference between a target and its neighbor should matter. A neighbor should be most facilitatory when it matches on the internal letters. This explains the finding that the usual N effect comes from body neighbors [Zie98]. A word node corresponding to body neighbor would not become highly activated, because it likely would not match on the important first letter. Thus, I propose that the N effect occurs as a result of top-down input to letter nodes via the summed excitation of a large number of *moderately* active word nodes. This also explains why non-word targets were not affected by the body-neighbor manipulation [Zie98]. Increasing the number of body neighbors does not



increase the number of highly activated word nodes. Under the assumption that only highly active word nodes slow RTs to non-words, body neighbors would not affect RTs to non-words. In contrast, increasing the number of N-metric neighbors makes a highly activated word node more likely.

The proposal that facilitation results from a moderately active non-target word nodes leaves open the possibility of an inhibitory effect for a *highly* activated non-target word node, as would be expected from lateral inhibition within the word level. This proposal explains observed influences of a single higher frequency neighbor. For five-letter French targets, the existence of a higher frequency neighbor mismatching at the fourth letter had an inhibitory effect in lexical decision, while the existence of one mismatching at the second letter did not [Gra89]. Perea investigated this phenomenon in English using a perceptual identification task for a briefly presented target (67 ms) which was followed by a mask [Per98]. The target was preceded by a 500-ms prime that was a higher frequency neighbor. When the prime mismatched the target on the third or fourth letters, there was an inhibitory effect on target identification (compared to an unrelated prime). In contrast, a prime mismatching on the first, second, or fifth letter had no effect.

The lack of effect for a mismatch at an external letter (first or fifth letter) [Per98] is explained by the edge bigrams. If an edge bigram is not matched, the neighbor does not become highly activated and does not have an inhibitory effect on the target. The effect of internal-letter position [Gra89, Per98] is explained by the sequential activation of bigrams. If a neighbor mismatches on the second letter, it is inhibited early and cannot accrue a high enough activation level to interfere with the target. However, a mismatch occurring later (at the third or fourth letter) has less of an effect, leading to high activation of the neighbor, and

an inhibitory effect on the target. A non-target formed by transposing two letters of the target, such as SALT and SLAT, is also highly activated under the bigram metric, because most of the bigrams are shared [Gra04b]. This accounts for the finding that having such a transposed-letter neighbor can be inhibitory [And96].

In summary, I propose that the facilitatory N effect occurs via moderately active word nodes. Such non-target nodes do not strongly inhibit the target, while their summed top-down input to the letter level provides facilitation. Increased RTs occur when a single neighbor is highly activated, strongly inhibiting the target.

## **Locus of Visual Field Asymmetries**

The fact that the normal visual-field x N-effect interaction was overridden by our manipulations demonstrates that it cannot be a result inherent hemispheric differences at the level of lexical access, because if it were, it would not be affected by such manipulations. Rather, an asymmetric word-level effect can result from differences in processing near the visual level. This casts doubt on the widely held assumption that hemifield effects reflect differential processing at the lexical level.

## **Letter-Position Encoding**

The highly specific, counterintuitive predictions were based on the details of locational gradient formation. The confirmation of these predictions provides strong support for the idea that letter-position encoding employs a spatial activation gradient, the formation of which requires hemisphere-specific processing, giving differing activation patterns across the visual fields. Although these ex-

perimental results do not directly confirm the claim that the locational gradient induces a serial encoding of letter order, the proposed dynamics do explain why top-down feedback has no effect when the internal letters already receive a relatively high level of excitatory bottom-up input.

## 9.8 General Discussion

The SERIOL model has elucidated the source of a phenomenon that has remained mysterious for decades, the asymmetry of the length effect [Mel57, Bou73, Ell88, Naz03, Jor03]. It has also explained the recently discovered asymmetry of the N effect, at the same time revealing the source of facilitation for high N. The model explains why the CVF patterns with the LVF for the N effect, but with the RVF for the length effect. For an N effect to occur, the slope of the locational gradient must be sufficiently steep that top-down input can assist the firing of the internal letters. The locational gradient is steeper in the LVF and CVF than in the RVF across early string positions, explaining the pattern of the N effect. In contrast, the length effect occurs when the locational gradient is not smoothly decreasing. This only occurs in the LVF (as a result of acuity-gradient inversion across a large number of letters), explaining the pattern of the length effect.

This analysis implies that it may also be possible for high N to compensate for a non-smooth, LVF gradient in longer words. Indeed, [Lav02a] showed that there was no length effect for LVF high-N words of three to five letters, while there was a length effect for low-N words. This lack of a length effect for high-N words was the first time that an absence of a length effect was demonstrated for LVF. In contrast, in the present work, we have shown for the first time how to abolish a length effect in a set of words that normally shows such an effect [Whi04c].

These results demonstrate that these hemispheric asymmetries do not entail different modes of lexical access. Rather, hemisphere-specific activation patterns are the cause. Thus, the locus of visual-field effects is lower in the processing stream than is commonly assumed. These results suggest that it is not appropriate to use visual half-field studies to investigate linguistic-level hemispheric specificity. As such experiments are currently widely used, this is an important finding. To further buttress this claim, we are currently applying our contrast-manipulation methodology to a semantic asymmetry related to primes that have two different meanings [Bur88]. Logically, if there is one mode of lexical access, as our results and brain-imaging evidence indicate, semantic asymmetries must also originate prelexically. Therefore, we expect to be able to reverse this asymmetry also. Mechanistically, degraded letter-position encoding in the LVF/RH may create more diffuse lexical activation (than RVF/LH presentation), causing more diffuse semantic activation, leading to an asymmetry in semantic priming. Extension of our results to the semantic level would conclusively demonstrate that VF asymmetries arise at a prelexical level, which would indicate that hemifield experiments should no longer be used to make claims about hemisphere-specific processing at the lexical level and above.

The fact that the SERIOL model has led to these experimental results illustrates the utility of the overall approach. The predictions and experimental designs were generated by reasoning about a theoretical model, not by running a simulation. The theoretical model was formulated by considering a wide range of behavioral data and neurobiological constraints. This allowed a theory of how letter-position is encoded in a mature brain, and has led to novel, counterintuitive predictions that have been experimentally verified, and have elucidated long standing questions in area of visual word recognition.

I believe that this general approach allows one to get at what the brain is doing in a way that is not achievable by training an artificial neural network. It forces consideration of what a brain is actually doing, and how it is doing it. More realistic and complex tasks can be modeled when the work is not limited by implementational issues. Rather, computation is considered at a more abstract level, but is still heavily constrained - by neurobiological plausibility, and behavioral patterns. Thus although the model is specified on the functional level, the specification is still highly specific, much more so than box-and-arrow models. This specificity is in evidence in the range of accurate predictions generated by the SERIOL model. Once neural mechanisms in a mature brain have been established, we are then in a better position to consider how learning occurs, because we know what the end point should be, and what computational mechanisms must be available.

In the following chapter, I consider the implications of the SERIOL model for the more general arena of visual object recognition and for dyslexia. In the subsequent chapters, I then apply the overall approach to the problem of parsing.

## **Chapter 10**

### **SERIOR Speculations**

In section 2.2, I claimed that understanding how the brain handles LPE should shed light on fundamental processing mechanisms. In this chapter, I address this issue. I start with a consideration of which aspects of the SERIOR model are learned, and which are innate. Based on this analysis, I discuss how the presumably innate aspects could apply to object recognition in general. I then consider how disruption to the learned aspects could contribute to dyslexia. This discussion will be sketchy and speculative. To treat these subjects in detail would require several more dissertations!

#### **10.1 Innate versus Learned Aspects of the SERIOR Model**

Starting at the highest level of the model, I now consider how the proposed processing could be learned during reading acquisition. The word level of the model corresponds to the orthographic lexicon. Obviously, people must learn to associate a word's spelling and its meaning. While I have used a localist encoding of the word level in the simulations, this assumption is not central to the theoretical model. I leave the nature of the encoding of the lexicon as an open question.

The lexical level is activated by bigram nodes, which represent the ordering between two letters. Thus, all relationships between the letters in a stimulus are encoded by a set of pairs. The general capacity to represent relationships in this way in the visual system may be innate (such as above/below relationships, as discussed in the following section).

The bigram level is activated by the serial firing of letter nodes. This serial encoding depends on the oscillatory nature of letter nodes. Obviously, the brain does not learn to use oscillations to encode information. Rather oscillatory dynamics must present as an innate encoding mechanism.

Serial firing also depends on a feature-level activation gradient. The left-to-right nature of this locational gradient is obviously learned, as it is based on reading direction. Furthermore, distinguishing objects by horizontal relationships is unnatural. The identity of a natural object does not change as it is rotated around the vertical axis; a lion is still a lion regardless of whether it is facing to the left or to the right. Thus the visual system must learn to distinguish horizontal order for the purpose of processing words, and it must learn to impose a monotonically decreasing activation gradient. However, the general mechanism of creating a location-invariant representation via the conversion of space into time is taken to be innate.

The edge level of the model is based on known properties of the primary visual areas, and these properties are therefore innate. The transformations between the edge and feature level constitute the learned nature of the locational gradient.

Thus these general representational mechanisms are taken to be innate: the pairwise representation of relationships, the existence of oscillatory cells, and the capacity to use these oscillatory cells to convert a spatial representation into a temporal representation via differences in activation levels. In the following

section, I discuss how these capacities could be employed in general object recognition.

Processing that is specific to visual word recognition occurs primarily at the feature level. The visual system must learn to encode letter order via a monotonically decreasing activation gradient across a retinotopic representation. I assume that this learned in response to a top-down attentional gradient. In section 10.3, I present a simple simulation showing the feasibility of such learning. I also discuss how failure to create the locational gradient may be a causal factor in dyslexia.

## 10.2 Object Recognition

There has been an ongoing debate as to whether objects are recognized via interpolation of view-dependent templates[Pog90], or by matching abstract structural representations [Bie87]. However, recent work has indicated that the visual system may use both approaches [Fos02, Hay03]. Both types of recognition entail similar problems of representing the relationship between sub-parts in a location-invariant way, so that a stimulus can be matched against a stored representation. In the view-dependent approach, this would involve two-dimensional relationships between features, while in the compositional approach this would involve three-dimensional relationships between volume primitives, known as *geons* [Bie87].

In an implemented model of the geon approach [Hum92], spatial relationships were encoded using the *above*, *below* and *beside* predicates. Thus ordering along the vertical axis was differentiated, but not along the horizontal axis. This is in line with the above observation that left-right relationships are not invariant for natural objects. In contrast, vertical relationships usually do not vary, because natural objects are not usually upside-down. Thus it is most important to rep-



resent vertical relationships. To represent the structure of the constituent geons, each geon was temporally bound to a one-place relationship. For example, if a cone appeared above a brick, *cone* and *above* would fire simultaneously (encoding that the cone is above something), while *brick* and *below* would fire simultaneously in a different time slot. However, this encoding leads to ambiguity if there are four or more geons above one another. The middle geons are both above and below another geon, so there is ambiguity about their relationships to one another.

The relationships between geons were established by an exhaustive comparison between locations. A coarse coding of ten units was used to encode the vertical spatial location of the center of mass of each geon. For example, using 1 to represent the topmost location, the *above* unit is activated if 1 and 2 are active, or 2 and 3, or 1 and 3, etc. This requires an *and* gate for each pair of locations that satisfies the relationship, and an *or* gate joining all of the *and* gates. While it is feasible to use this approach for a small number of possible locations, the wiring necessary for a more realistic network becomes prohibitively expensive.

How could the proposed representational mechanisms overcome these difficulties? In the SERIOL model, a relational unit (i.e. a bigram) represents a two-place relationship, rather than a one-place relationship. Such units would reduce ambiguity. Thus vertical relationships could be represented by units representing above/below. For example, when a cone appears above a brick, it activates a *cone-above-brick* unit. The number of required units is the square of the number of geons (24) [Bie87], which is not prohibitively large.

In the SERIOL model, the left-right relationship is identified not by exhaustive comparison of locations, but rather by order of firing. The same principle could be used to identify above-below relationships. In the SERIOL model, the sequential

firing is achieved via a monotonically decreasing activation gradient. However, there is no evidence for a monotonically decreasing gradient from the top to the bottom (or bottom to top) of the visual field. Instead, the visual system may use the acuity gradient directly, but differentially, in the upper visual field (UpVF) and the lower visual field (LoVF). If geon1 is above geon2 in the UpVF, geon1 would have a *lower* acuity than geon2. If the UpVF acuity gradient is converted into sequential firing of geons, the geon1-above-geon2 unit should be activated when geon1 fires *after* geon2. In contrast, if geon1 is above geon2 in the LoVF, geon1 will have a *higher* acuity than geon2. If the LoVF acuity gradient is separately converted into a sequential firing pattern, the geon1-above-geon2 unit should be activated if geon1 fires *before* geon2. So the wiring between geon and bi-geon units would vary with visual field. This wiring would be part of the visual system's innate capacity to represent spatial relationships. Relationships could also be hardwired across the visual fields. That is, if geon1 appears in the UpVF and geon2 in the LoVF, the geon1-above-geon2 unit is activated.

In contrast to the UpVF and LoVF, there would be no visual-field-specific wiring for the left and right visual fields, because left-right relationships are not usually invariant. In both visual fields, if geon1 fires after geon2, the geon1-beside-geon2 unit would become activated. In order to read, this mechanism would have to be overridden via a monotonically decreasing gradient which induces first-to-last sequential firing.<sup>1</sup>

---

<sup>1</sup>If there is an innate mechanism for encoding above-below relationships, but not left-right relationships, why then are most scripts read horizontally? The visual field is more extensive along the horizontal axis than along the vertical axis, and acuity decreases more quickly along the vertical axis than the horizontal axis. It may be the case that the increased acuity along the horizontal axis outweighs the cost of special processing. Also, although above-below rela-

Thus, for general object recognition, separate temporal encodings may be induced along the vertical axes in the UpVF and LoVF, and along the horizontal axes in the LVF and RVF. This would lead to the activation of bi-geon units encoding *above*, and bi-geon units encoding *beside*. This location-invariant representation could then be matched against a stored representation based on bi-geon units.<sup>2</sup> A similar encoding mechanism could be based on features, rather than geons, for matching view-dependent templates.

This sketch has suggests that the basic principles of encoding in the SERIOL model could plausibly be extended to the domain of object recognition in general. Of course, many details remain to be worked out.

### 10.3 Feature-Level Processing and Dyslexia

According to the above discussion, the task of learning to encode letter order primarily consists of learning to create the locational gradient. What could drive this learning? I assume that it is attention based. Because print-to-sound translation proceeds from left-to-right (in a left-to-right language, of course), attention is first focused on the first letter, then on second, etc. This may create a top-

---

tionships may be directly computed, this would not be a result of sequential firing across the letters. For example, for the vertical word GLEN fixated in the center, L and E would fire, and then G and N. Thus there is no letter-based invariant representation of order. Recall that the phonological route requires such a representation. Therefore, acuity gradient inversion (in the UpVF) may be necessary for languages read from top to bottom, giving no advantage over horizontal scripts.

<sup>2</sup>*Inside* and/or *in-front-of* relationships would also be required. These principles are less applicable for determining such relationships. Rather, a mechanism to compare spatial extents would necessary.

down attentional gradient across the letters. This top-down gradient may then drive learning on bottom-up connections (between the edge and feature levels) and lateral connections (within the feature level). Over time, the visual system learns to automatically create an activation gradient, without top-down support.

### 10.3.1 Simulation of Learning to Form the Locational Gradient

To test the feasibility of this scenario, I ran a simulation with one layer of feature nodes that were fully interconnected. A self connection was excitatory, whereas connections to other nodes were inhibitory. Bottom-up input was in the form of an acuity gradient, whereas top-down input was in the form of the locational gradient. Following the reception of bottom-up input, the network iterated for three cycles. The resulting activations were compared to locational gradient. If a node's activation was too low, the strength of the self-connection was increased. If the activation was too high, the strengths of the inhibitory connections were increased. Then all weights decayed slightly.

The simulation was performed on set of 10 nodes, where nodes 1::5 represented the LVF/RH and 6::10 represented the RVF/LH. Bottom-up activations  $BU_i$  increased for  $i = 1$  to 5 (from 3.0 to 5.0), and decreased for  $i = 6$  to 10 (from 5.0 to 3.0). The network was trained for stimuli at  $start::10$ , where  $start$  was varied from 1 to 6.  $BU_i$  was set to 0.0 for  $i < start$ . The top-down activation  $TD_i$  was set to 0.0 for  $i < start$ , and to  $5.0 - (i - start) * 0.5$  for  $i \geq start$ . Thus the network had to learn a single set of weights that would generate a gradient of the same shape for all stimulus locations.

Excitatory and inhibitory connection weights were initially set to 0.005 and

-0.005, respectively. On each iteration, a node's activation  $A_i$  was increased by the dot-product of the feature activation vector and the weight vector. After 2 iterations, if  $A_i < TD_i$  (within a tolerance of 0.05), the self-connection weight was increased as follows:

$$w_{ii} = w_{ii} + LR * TD_i$$

where  $LR$  is the learning rate. If  $A_i > TD_i$ :

$$w_{ij} = w_{ij} - LR * A_j \quad \text{for } i \neq j$$

Then all weights were reduced:

$$w_{ij} = D * w_{ij}$$

where  $D < 1$ .

$LR = 0.0006$  and  $D = 0.999$  produced good performance. After 10000 learning cycles, the desired monotonically decreasing gradient was created for all values of *start*. Although connection weights were initially symmetric, this training induced an asymmetry, such that weights on inhibitory connections from  $i$  to  $j$  were more negative for  $i < j$  than  $i > j$ , and were more negative for  $i < 6$  than  $i > 5$ . This is in line with the proposed left-to-right inhibition that is stronger for the LVF/RH. Weights on self-connections were higher for  $i < 6$  than for  $i > 5$ . This is in line with increased excitation for the LVF/RH. Thus simple learning rules yielded connection weights with the proposed characteristics, demonstrating the plausibility of learning to form a locational gradient.

### 10.3.2 Dyslexia

If the visual system fails to learn to create the locational gradient, letter order will not be quickly and automatically represented. A deficit in visual processing

is consistent with evidence showing that normal readers show an early (<150 ms post-stimulus) increased activation in the the left posterior fusiform gyrus in response to letter strings, while dyslexic readers do not [Hle97, Tar99]. This early left-lateralization in normal readers may correspond to the initiation and performance of string-specific processing (i.e. locational gradient formation). Lack of locational-gradient formation in dyslexics is also consistent with a study of the OVP in young readers who were normal or dyslexic [Duc03]. For normal readers, initially fixating on the first letter of a word yielded much better reading performance than fixating on the last letter. This is in line with the usual bias in OVP experiments, which I take to result from the necessity of acuity-gradient inversion in the LVF. If dyslexics do not create a locational gradient, this asymmetry should not be present because acuity gradient inversion would not be performed. Indeed, the dyslexic readers showed a symmetric viewing-position function.

If a rapid sequential representation of letter order cannot be induced, the visual system may compensate by performing an overt scan of the string. That is, instead of creating a sequential encoding in a single fixation, multiple fixations are carried out across the string. Thus a sequential encoding is created, but on a much longer time scale. This proposal explains fixation data for normal versus dyslexic children. Children read words varying in length from 2 to 14 letters while their eye movements were monitored [Mac04]. (The study was in German, so long words were common in the test language.) Normal and dyslexic children showed similar patterns for words of 2-4 letters. As string length increased, the patterns diverged sharply. (The following results are the medians for each group.) For the longest words, normal subjects initially fixated 4.4 letter-widths from the beginning of the word, performed saccades of 3.5 letter-widths, and finally fixated 4.3 letter widths from the end of the word. In contrast, dyslexic children initially

fixated 2 letter-widths from the beginning of the word, performed saccades of 2 letter-widths, and finally fixated 1.8 letter widths from the end of the word. Thus, on a single pass over a long word, dyslexic readers made twice as many saccades as normal subjects. In addition, holding duration per fixation was longer in the dyslexic children.

The dyslexic pattern is consistent with a strategy in which one to three letters are processed per fixation, where a slow, top-down attentional mechanism used to scan letters within a fixation. Because it takes so long to process all the letters of the string, information about the initial letters may be lost by the time that the end of the string is reached. Thus multiple passes across the string may be required to read the string. Indeed, the number of backward saccades increased with word length for the dyslexic, but not the normal, subjects.

Such a deficit in encoding letter order could have ramifications for learning grapheme-to-phoneme correspondences. Perhaps a rapid sequential representation of letter order is necessary for learning an effortless mapping to phonology. That is, it may be necessary to temporally align sequential orthographic and phonological representations in order to learn to effectively translate between the two types of encodings. When a suitable, robust orthographic encoding is not available, this may interfere with such learning. Thus the well-known phonological deficits observed in dyslexics may actually have their source in a visual deficit, at least in some cases.

### **10.3.3 Magnocellular Deficit**

What underlying deficit could prevent formation of the locational gradient? As discussed in section 3.3, recent research has revealed a magnocellular deficit in some dyslexics. The dorsal route of the visual system, which processes motion,

location, and attention, primarily receives inputs from the magnocellular pathway [Mau90]. Therefore, the underlying problem may be attentional, in that the proper top-down attentional gradient is not available to drive learning of the locational gradient. This proposal is consistent with evidence of LVF mini-neglect and RVF over-distractibility in dyslexics [Fac01, Har01], indicating abnormal attentional gradients. Vidyasagar has made a similar proposal, suggesting that dyslexics are unable to sequentially deploy attention across the string in a rapid, top-down manner [Vid01, Vid04]. In contrast, I propose that attentional problems prevent learning of the normal automatic, bottom-up processing that drives a sequential representation of letter order.

The ventral route of the visual system, which processes form and color, receives inputs from both the parvocellular and magnocellular pathways [Fer92]. Little is known about the role the magnocellular pathway along the ventral route. Because magnocells are larger and process information more quickly than parvocells, magnocells may rapidly set up a low spatial-frequency representation of the visual scene, onto which the parvocells fill in detail [Car87, Del00, Van02]. In line with this fast processing, another role of the magnocellular system in locational gradient formation may be to rapidly drive the inhibition that is necessary to invert the acuity gradient and create the locational gradient. If magnocells are functioning too slowly, it may not be possible to set up the locational gradient quickly enough to subserve the bottom-up induction of a serial encoding.

Some dyslexics do not show magnocellular problems. They may fail to develop the locational gradient for other reasons. Perhaps an auditory deficit directly prevents the development of a phonological representation that is based on individual phonemes. In the absence of such a representation, there may be less pressure to develop an orthographic representation that aligns with the phonemic



representation.

Another potential role of the magnocellular system lies at a higher level of processing. Recall that a bigram node is activated by letters that fire in a particular temporal sequence. This response profile is similar to cells which only fire for a stimulus moving in a certain direction. Such directional sensitivity is characteristic of motion-detection cells in V5 [Mau83]. Due to this functional similarity, bigram nodes may be located in V5. Because problems with formation of bigram nodes would result in impaired ability to form and store representations of letter order, this proposal is consistent with evidence showing a correlation between motion-detection ability and both letter-position encoding ability [Cor98], and the ability to distinguish real words from pseudohomophones (e.g. *rain* versus *rane*) [Tal00]. It is also consistent with the phenomenon of letter-position dyslexia in some subjects suffering from occipitoparietal lesions, whose error responses are anagrams of the target word [Fri01]. Such subjects may have an intact serial encoding of letter order, but may lack reliable bigram representations. Similarly, some developmental dyslexics with magnocellular problems may fail to develop bigram nodes, leading to difficulty in developing an orthographic lexicon. An impairment at the bigram level may be directly due to a processing deficit in V5, or may originate earlier in the processing stream, perhaps due to a lack of a rapid sequential letter-based representation, which may be necessary to drive formation of bigram nodes.

#### **10.3.4 Possible Experimental Tests of these Proposals**

While highly speculative, the above analyses do suggest some avenues of experimental investigation. The proposal that dyslexics fail to learn to form a locational gradient could be tested by investigating letter perceptibility patterns for

lateralized presentation of three-letter consonant strings. For normals, the best-perceived letter in each visual field is the letter farthest from fixation, as discussed in section 4.2.3. In the LVF, this is due to the feature-level, left-to-right inhibition necessary to invert the acuity gradient. I would expect a different pattern in dyslexics, with perceptibility more in proportion to acuity. In the RVF, the final letter is the best perceived because it is not inhibited by a subsequent letter at the letter level. If dyslexics rely on a top-down attentional scan, this pattern should not be present. Thus for dyslexics, I would expect positional symmetry across the visual fields (resulting from a top-down scan), somewhat modulated by acuity, giving a V-shaped LVF pattern and an initial-letter primacy in the RVF.

If the predicted pattern is found for dyslexics, this would suggest that it may be possible to treat dyslexia via the external imposition of a locational gradient, in order to jump-start its automatic formation. This could be accomplished by creating a contrast gradient across words presented on a computer screen. That is, the first letter has the highest contrast, the second letter has somewhat lower contrast, the third somewhat lower than the second, etc. Each word should be centrally presented for 200 ms, to force processing within a single fixation. A previous study has shown that treatment utilizing brief presentation (100 to 300 ms) of words, either centrally or randomly lateralized, improved spelling ability in dyslexics, whereas longer central presentation (1500 ms) or presentation to a single visual field did not [Lor04]. This increased spelling ability may reflect a more reliable orthographic lexicon, stemming from more robust letter-position encoding. Perhaps brief presentation by itself forced formation of the locational gradient, because visual scanning was not an option. It would be interesting to see if imposition of a contrast gradient on such stimuli would generate a greater improvement in reading ability than standard stimuli.

The proposal that V5 houses bigram units could be tested via transcranial magnetic stimulation (TMS), which temporally disrupts neural activity in a small area of the cortex. Under TMS to V5, a task that requires encoding of relative letter order should be disrupted. A previous experiment has yielded suggestive results. TMS to V5 impaired ability to read pseudowords [Lis95]. Pseudoword reading requires precise encoding of letter order because top-down, lexical information is not available. Interestingly, the number of transposition errors preferentially increased, consistent with an induced deficit in positional coding ability. There were 75% more transposition errors (21/29) versus 33% more replacements (33/100) and additions (6/18) under TMS as compared to no stimulation. Further investigations could be carried out on Hebrew subjects performing the reading task, to see if letter-position dyslexia [Fri01] can be induced. (As discussed in section 3.3, Hebrew is an ideal language for revealing letter-position dyslexia because vowels are not explicitly represented.) Alternatively, the lexical-decision task used by [Cor98], wherein nonwords were formed by transposing letters of real words, could be employed in any language. As a control, nonwords formed by replacing a letter of a word should also be included. If false positives to anagrammatic non-words were selectively increased under TMS to left V5, this would indicate that letter-position encoding in particular was disrupted.

### **10.3.5 Summary**

The primary locus of learning in the SERIOL model is at the feature level, where the locational gradient is formed. Failure to learn to produce a locational gradient may contribute to dyslexia. Such failure may stem from a magnocellular deficit, or from the lack of a robust phonemic encoding.

The general mechanisms of encoding relationships with a set of pairs, and

of using a spatial activation gradient to induce a temporal, location-invariant encoding could be used by the visual system for object recognition in general. To represent vertical relationships, the visual system would likely directly use the acuity gradients in the upper and lower visual fields. This would require each visual field to interpret the order-of-firing information differently.

This concludes the discussion of LPE. In the remaining chapters, I begin to tackle the problem of how the brain creates the representation of sentence structure.

## Chapter 11

### The Parsing Problem

#### 11.1 Specification of the Problem

A sentence is interpreted to ascertain “who did what to whom”. A verb specifies the “what”. The participants in the actions undertake different *thematic roles*; the “who” is termed the *Agent*, and the “whom” the *Theme*. For example, consider the sentence:

1. The dog that Mary adopted bit Tim.

The main idea of the sentence is (*Agent=dog, action=bit, Theme=Tim*), where the additional information (*Agent=Mary, action=adopted, Theme=dog*) modifies (*Agent = dog*).

The job of the human parser is to take a sequence of words and convert it into such a representation of meaning. This task involves computational and representational problems that are much more difficult than those of letter-position encoding! The most important differences are as follows.

- There must be unlimited productivity. Any word can appear in an instance of the corresponding syntactic or thematic category. In letter position encoding, there are a small number of elements (letters); relationships between

elements can be represented via (bigram) units encoding every possible pair of elements. This type of conjunctive representation is not feasible in parsing due to the large number of words.

- The resulting representation must be *hierarchical*. It must be possible to represent multiple clauses and the relationships between those clauses. For example, a relative clause is embedded within the main clause in (1). In contrast, letter-position encoding only requires a linear representation of the relationships between elements.
- It must be possible to associate particular non-contiguous items in particular ways. In the above example, *dog* must be associated with *bit*, while the intervening material must not affect this association. This problem is especially difficult in the case of a *center-embedding*, as in the above example, where a new clause is started in the middle of a higher level clause. This leads to multiple unattached Agents (*e.g.*, *dog* and *Mary*), and each one must be associated with the proper verb. In contrast, in letter-position encoding, all elements bear the same type of relationship to each other and this problem does not arise.

Thus, the overall problem is “What representations, and transformations on these representations does the brain use to convert a sequence of words into a hierarchical representation of meaning?” This does not include the question of how a word’s meaning is actually represented in the brain. Rather, the focus is on how words could be represented that would allow them to be combined into hierarchical structures. The resulting hierarchical representation of thematic roles is dubbed the *thematic tree*.

Due to the difficulty of the above question, I initially attack this problem by focusing on the neural basis of the underlying representations, while considering the operations on those representations at the algorithmic level. That is, the nature of the thematic tree and of the intermediate representations supporting its construction are considered at the neural level. The parsing algorithm that operates over these representations is considered at the symbolic level, for now.

To satisfy first two requirements discussed above (productive and hierarchical representations), two types of operations must be available. It must be possible to *bind* together an arbitrary word with a thematic role, giving productivity. It also must be possible to *merge* multiple such associations into a single unit so that the entire unit can enter into a binding relationship, thereby allowing hierarchical structure. To understand what third requirement above entails, a discussion of the Chomsky hierarchy of formal languages is in order.

## 11.2 Computational Constraints

Chomsky [Cho59] identified a relationship between the complexity of formal languages and the computational machinery required to accept or reject a string as being a well-formed string of a language. The simplest class of language, a *regular expression*, is recognized by a finite-state machine, which consists of a set of states, and state transitions triggered by input tokens. See Figure 11.1. A finite-state machine can recognize strings of the form  $a^n b^m$ . Grammars of this type correspond to phrases in natural language. For example, a definite noun phrase is recognized by a finite-state machine that expects *the*, followed by any number of adjectives, followed by a noun. A finite-state machine can also recognize strings of the form  $(ab)^n$ . This grammar corresponds to right-branching

clauses, where  $a$ 's are nouns and  $b$ 's are verbs. For example:

2. John knows Sue thinks Bill lied.

In contrast, a finite-state machine cannot recognize strings of the form  $a^n b^n$  (where  $n$  is unbounded), because there is no way to ensure that the number of  $a$ 's and  $b$ 's match up when all the  $a$ 's are processed first. Note that  $a^n b^n$  corresponds to center-embedding, for example (noun (noun verb) verb). Thus, a finite-state machine cannot handle the general case of center-embedded clauses. It is often pointed out that humans cannot either, as more than one center-embedding leads to an uninterpretable sentence, such as:

3. The man that the dog that Mary adopted bit screamed.

However, the human parser can handle certain double center-embeddings, as in the following:

4. The fact that the dog that Mary adopted bit Tim upset her.

Thus humans can indeed parse multiple center-embeddings. Recent research has suggested that the ability to process center-embeddings may be uniquely human [Fit04]. Both humans and tamarins (a type of primate) rapidly learned to recognize sequences of syllables of the form  $(ab)^n$ , where  $a$ 's were in a female voice, and  $b$ 's were in a male voice, for  $n = 2$  or  $3$ . However, only humans learned to recognize sequences of syllables of the form  $a^n b^n$ , for  $n = 2$  or  $3$ . This ability to recognize center-embeddings may reflect a neural adaptation that is specific to language ability [Fit04].

What computational machinery is necessary for recognizing center-embedded structures? Such processing requires the functionality of a *stack*. A stack is characterized by the *push* and *pop* operations. *Push* adds an item to the top



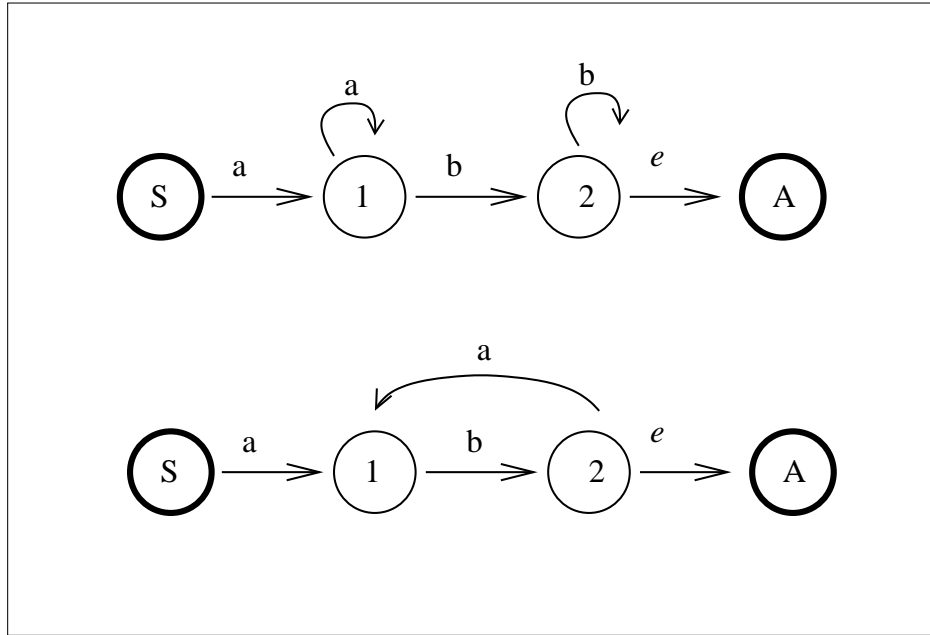


Figure 11.1: Examples of finite state machines (FSMs). Each recognizer consists of a start state, S, and an accept state, A, and intermediate (numbered) states. Transitions occur between states for specific input tokens, where  $e$  represents the end-of-string token. The top FSM accepts strings of the form  $a^n b^m$ , for  $n \geq 1$  and  $m \geq 1$ . For example, the string  $a_1 b_1 b_2 b_3$  would activate the following sequence of states: S,1,2,2,2,A. The bottom FSM accepts strings of the form  $(ab)^n$ , for  $n \geq 1$ . For example, the string  $a_1 b_1 a_2 b_2$  would activate the following sequence of states: S,1,2,1,2,A.

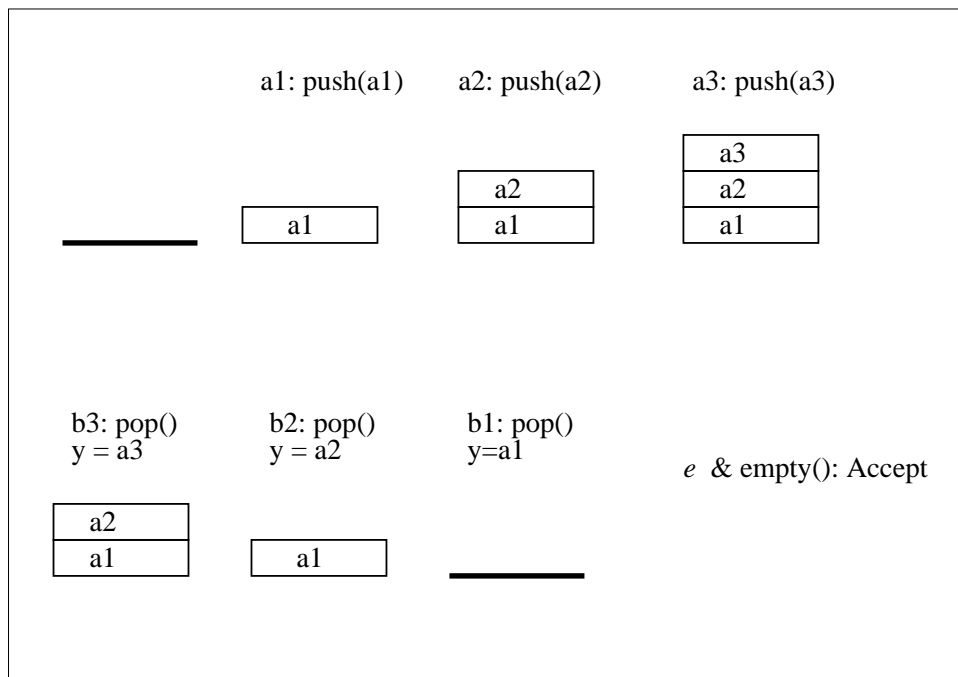


Figure 11.2: Example of using a stack to recognize strings of the form  $a^n b^n$ . A stack  $S$  provides the  $push(S, x)$  operation, which puts  $x$  on the top of the  $S$ , the  $pop(S)$  operation, which removes the top item from  $S$  and returns it, and the  $empty(S)$  operation, which is true only if there are no items on  $S$ . The string  $a^n b^n$  can be recognized using the following algorithm for token  $x$ :

```

if  $x = a$  then  $push(S, x)$ 
else if  $x = b$  and not  $empty(S)$  then  $y = pop(S)$ 
else if  $x = e$  and  $empty(S)$  then Accept
else Reject

```

The operation of this algorithm is illustrated for the string  $a_1 a_2 a_3 b_3 b_2 b_1$ , where the boxed items represent the items on the stack and a line represents an empty stack. In natural language, such a string would correspond to multiple center-embeddings, where a's are subjects and b's are verbs. The recognition algorithm could be augmented to create a representation of the structure of the input by adding appropriate structure-building operations. For example, when an item is popped, a structure could be created that represents the integration of  $y$  (a

of a stack, while *pop* removes the topmost item. Thus, items are popped in the reverse of the order that they were pushed. See Figure 11.2.

Of course, in processing natural language, it is insufficient to merely accept or reject a string of words as being a well-formed sentence. Rather, a representation of meaning must be created as the words are processed. A recognizer can be augmented to construct a such representation. For example, when a pop operation is triggered, the result of the pop could be taken to be the Agent of the current verb (assuming that *a*'s correspond to nouns and *b*'s to verbs in our example).

Natural language contains other structures, called crossed-serial dependencies, that cannot be parsed using a finite-state machine or a stack, as in:

5. John, Bill, and Tom were wearing green, blue, and purple, respectively.

Here, *respectively* indicates that the following associations should be formed: (John, green) (Bill, blue) (Tom, purple). However, stack-based processing would yield (Tom, green) (Bill, blue) (John, purple). In this case, the functionality of a *queue* is required, which is characterized by the *append* and *remove* operations. *Append* adds an item to the end of a queue, while *remove* takes an item from the front of a queue. Thus items are removed in the same order that they are appended. To parse the above sentence, *John*, *Bill*, and *Tom* would be successively appended to a queue. Then *green* would trigger a remove, giving *John*; *blue* would trigger a remove, giving *Bill*, etc.

Thus, in order to process center-embeddings and crossed-serial dependencies, the human parser must be able to perform stack-like and queue-like operations in working memory. Therefore, such operations are an important component of the intermediate representations that allow construction of the thematic tree. In the following two chapters, I consider constraints that narrow the possibilities

for how the thematic tree and the intermediate encodings are represented in the brain.

### 11.3 Neurobiological Constraints

For the problem of letter-position encoding, the architecture of the visual system constrained the lowest level of representation. Due to the high-level nature of the parsing problem, such explicit constraints are not available. Rather, there are more general constraints of neurobiological plausibility as follows.

- There are a finite number of neurons of fixed connectivity. A node representing an association between particular words cannot magically appear.
- Connection weights cannot be quickly altered and then returned to their original values. While there is evidence for rapidly occurring changes in synaptic strength in the hippocampal system, these changes are enduring [Bli73]. (This phenomenon is called *long-term* potentiation.) Due to the large numbers of sentences parsed, it is unlikely that the human parser relies on such semi-permanent changes to connection weights.
- However, it should be possible to store the thematic tree in the hippocampal system, if desired. Therefore, it should be possible to encode the information in the thematic tree into long-term storage, based on changes to connection weights.

A wide range of imaging studies have revealed brain areas and activation components associated language processing. Such studies provide little information about the nature of the underlying neural representations, and will not be reviewed here. However, imaging studies in which frequency-band power is an-

alyzed could potentially be informative. If power in a certain band increases during a task, this may indicate that performance of the task relies on oscillatory activity in that frequency band.

A range of studies have shown an increase in theta-band power in tasks that employ verbal working memory [Kli99]. A study using intracranial electrodes (in epileptic patients) allowed a particularly precise measurement of the temporal aspect of theta-band synchronization [Rag01]. These subjects performed the Sternberg task, in which 1 - 4 digits were memorized, followed by a delay interval, and then a probe. The subject then indicated whether the probe appeared in the memorized list. Spectral analysis showed a sharp increase in theta power at the start of the memorization phase. Theta power was maintained during the delay phase, and returned to baseline levels after the probe. This pattern occurred only in the theta band. In an MEG of the Sternberg task, theta power systematically increased in frontal areas as the number of digits to be remembered increased from 1 to 3 to 5 to 7 [Jen02]. Together, these studies suggest that items in verbal working memory are stored on an oscillatory carrier wave, as in the Lisman and Idiart model [Lis95] (discussed in section 6.1.2). This view is further supported by a clever experiment in which auditory clicks were presented at varying rates during performance of the Sternberg task [Bur00]. When the frequency of clicks fell just below 21 Hz, RTs were slowed and when the frequency fell just above 21 Hz, RTs were speeded. The largest changes occurred on those trials in which the largest number of items had to be remembered. These results suggest that the clicks affected the duration of gamma cycles on which items were stored in working memory, (As gamma cycles fall in the range of 40 Hz, the 21 Hz stimuli would correspond to a harmonic of that frequency.)

An EEG study has linked these phenomena to sentence processing, showing that theta power in particular increased as a sentence was read [Bas02]. Another study has shown effects of grammatical class (noun vs verb) on theta power [Kha04]. However, semantic processing seems to have no effect on theta power. A comparison of two tasks (reading a sentence versus reading a sentence and giving the superordinate category of one of the words) showed no difference in the theta range between the reading-only and semantic task, while alpha power increased for the semantic task [Roh01]. In sum, these results suggest that theta oscillations may play a role in syntactic encoding in working memory during sentence processing.

## Chapter 12

### Behavioral Results on Parsing

Of course, it is also more difficult to investigate parsing behaviorally than letter position encoding. The most informative data come from when the parser breaks down. Such break-down is generally measured by off-line difficulty ratings, or by the on-line measure of reading times in a self-paced reading study. In such a study, the words or phrases are sequentially revealed, and the timing is controlled by the subject. This allows a record of how long the subject spends on processing each word or phrase.

The human parser experiences difficulty in two situations: complexity and reanalysis. If the structure of a sentence is too complex, it becomes too difficult to process, as for the doubly center-embedded relative clauses in (3). Alternatively, difficulty can arise when the structure of the sentence is ambiguous, and the wrong analysis is initially chosen. In some cases, an initial incorrect analysis can be easily reanalyzed to give the correct structure, while in other cases, it cannot. For example, the following sentence is very difficult to understand.

6. The doctor told the patient that he was seeing that it was time to leave.

Here *that he was seeing* is initially taken as the Theme (what the doctor told the patient), and it is difficult to reinterpret it as relative clause modifying *the*

*patient* when the actual Theme (*that it was time to leave*) is encountered.

It is possible that both types of phenomena arise from the way that intermediate representations in working memory are used to construct the thematic tree. In the complexity case, intermediate representations may become unable to support the generation of the thematic tree. For reanalysis, the nature of these representations may determine why some reanalyses are easy, and some are not. In the following, I will concentrate on complexity phenomena. I first review the experimental results, and then discuss psycholinguistic models and metrics that have been proposed to account for these results.

## 12.1 Complexity Phenomena

### 12.1.1 Center-Embedding versus Crossed-Serial Dependencies

In English, center-embedding occurs when an embedded clause follows the subject. In languages with other word orders, center-embedding can occur under different circumstances. In German, nested infinitival clauses result in center-embedding. For example, the sentence:

7. Joanna helped the men teach Hans to feed the horses.

is expressed as follows in German:

8. Johanna hat den Mannern Hans die Pferde futtern lehen helfen.

Joanna has the men Hans the horses to-feed to-teach helped.

In Dutch, the same sentence would be expressed using crossed-serial dependencies, where the first subject is associated with the first verb, the second with the second verb, etc. :



9. Jeanine heeft de mannen Hans de paarden helpen leren voeren.  
Joanna has the men Hans the horses helped to-teach to-feed.

A study of the relative ease comprehension for the Dutch versus German constructions showed that the Dutch version is easier [Bac86]. Despite the fact that center-embeddings are generally more common across natural and artificial (computer) languages, and are more complex according the Chomsky hierarchy [Cho59], crossed-serial dependencies are easier to process.

### 12.1.2 Different types of English doubly center-embedded clauses

A center-embedded clause in English could either be a *relative clause* (RC) or a *noun complement* (NC). In a relative clause, there is “gap” corresponding to the the noun phrase being modified. For example, in the sentence:

10. The dog that Mary adopted bit Tim.

*that Mary adopted* is a relative clause with a gap following *adopted*. This gap corresponds to *the dog* (*i.e.* Mary adopted the dog). In contrast, there is no gap in a noun complement, which can only follow a word whose meaning is related to a proposition. For example,

11. The fact that the dog bit Tom caused him to scream.

Here *that the dog bit Tom* is a noun complement. It is a complete clause, elaborating on *the fact*.

As mentioned in section 11.2, some doubly center-embedded clauses are very difficult to understand, while some are not. A sentence in which both are relative clauses (RC/RC) belongs to the former category, for example:

12. The man that the dog that Mary adopted bit screamed.

Yet, if the outer embedded clause is an NC (NC/RC), such a construction seems much easier [Gib98]:

13. The fact that the dog that Mary adopted bit Tom upset her.

However, the opposite ordering, an RC/NC, seems at least as difficult as an RC/RC [Gib98]:

14. The woman who the fact that Rover bit Tim upset yelled at the dog.

Next I present some phenomena related to embedded clauses.

### **12.1.3 Interference in Working Memory**

One possible source of difficulty in center embeddings is that multiple similar items (i.e., unattached subjects) must be maintained in working memory. Across a range of domains, it has been shown that similarity among items in working memory interferes with the ability to remember and differentiate those items. Thus, this general difficulty may also apply to syntactic working memory [Lew96].

Lewis and Nakayama [Lew02] investigated the effects of similarity in Japanese using off-line complexity ratings. In Japanese, objects precede verbs. Thus, a sentence with a sentential complement like:

15. John knows that Bill likes Sue.

would be expressed with the following word order, giving a center-embedding:

16. John [Bill Sue likes] that knows.

Noun phrases are case-marked with suffixes, indicating their role in the sentence, where *-ga* indicates a subject (nominative case), *-o* indicates an object (accusative case), and *-ni* indicates an indirect object (dative case). Thus the above sentence would have the following form:

17. NP-*ga* NP-*ga* NP-*o* V that V.

Due to these factors, many unattached NPs can be accumulated in working memory, and the effects of similarity can be easily investigated. In a pilot study, twenty different syntactic structures were used in which the following factors were manipulated: level of embedding (0 or 1 embedded clause), number of NPs (1 to 5), similarity - maximal number of NPs with the same case (1 to 3), and adjacency - maximal number of *adjacent* NPs with the same case (0, 2, or 3). Ease of understanding was rated on a scale from 1 (easy) to 7 (difficult). Regression analyses showed that a combination of similarity and adjacency was the best predictor of difficulty ratings, accounting for 73% of the variance. That is, perceived complexity increased as the number of NPs with the same case marking increased, and as their proximity to each other increased.

This phenomenon was investigated further in a study in which the number of nominative NPs was held constant at 2, and the total number of NPs (3 or 4) and number of adjacent nominative NPs (0 or 2) were manipulated. With 0 adjacent nominative NPs, the total number of NPs affected perceived difficulty (ratings of 3.0 vs 4.2 for 3 vs 4 NPs). With 2 adjacent NPs, difficulty was higher, and was unaffected by the total number of NPs (ratings of 5.07 vs 5.22 for 3 vs 4 NPs). Thus, the proximity of nominative NPs had the largest impact on difficulty ratings.

However, these findings do not reveal whether it is the syntactic category (i.e.,

nominative) or the surface form (i.e. both -ga marked) that matters in determining similarity. To get at this question, Lee and Nakayama [Lee03] performed a similar investigation in Korean, which is structurally similar to Japanese. However, Korean has two different nominative case markings (-ka or -i), depending on whether the noun ends in a vowel. Syntactic class was varied by topicalizing the main subject. A topicalized NP indicates the focus of the sentence. It carries a different case marking (-nun), and is not necessarily a subject.

In a self-paced reading study of sentences with sentential complements, the first NP had either the -ka, -i, or -nun case-marking and the second NP had either the -ka or -i marking. The results showed that topicalized sentences were easier than the nominative sentences. Within the nominative sentences, those with dissimilar sequences (-ka,-i or -i,-ka) were easier than those with similar sequences (-i,-i or -ka,-ka). Thus both syntactic class and surface form influenced difficulty.

#### 12.1.4 NP-type effects

Experiments in English and Dutch have shown that the syntactic type of subject NPs influences difficulty. In the following, I will refer to the first NP as N1, the second NP as N2, etc.

#### English

Off-line complexity ratings have shown that the difficulty of an RC/RC is influenced by the type of the innermost subject [Gib98, War02a]. If N3 is an indexical (first- or second-person) pronoun, an RC/RC seems easier than if N3 is a name or a full noun phrase (FNP, e.g., *the woman*), for example:

18. The man that the dog that I adopted bit screamed.

If N3 is a third-person pronoun with or without a referent, it seems somewhat more difficult than indexical pronoun, but easier than a name or FNP [War02a]:

19. According to Sue, the man that the dog that she adopted bit screamed.

The man that the dog that she adopted bit screamed.

One possible explanation for these effects is that a pronoun reduces interference in working memory, because the subjects are less similar to each other. However, if N3 is an quantified pronoun (such as *everyone*), an RC/RC seems easier than if N1 or N2 is a quantified pronoun [War02a]. There are two consecutive non-pronouns when either N1 or N3 is a pronoun, yet ease of processing differs. This effect of position suggests that the influence of N3-type is not merely a result of reducing the number of similar adjacent items in working memory.

## Dutch

Next we consider effects in crossed-serial dependencies, based on self-paced reading studies [Kaa04]. In each experiment, three subjects (N1-N3) and an object (N4) preceded three verbs, and the syntactic types of N2 and N3 were varied. In Exp. 1, N2 and N3 were either both pronouns or names, while N1 and N4 were both FNPs. In this case, NP type (of N2 and N3) had no effect on reading times at any of the verbs. In Exp. 2, N2 and N3 were either both pronouns or FNPs, while N1 was a name and N4 was an FNP. In this case, reading times increased at V1 under the FNP condition.

Why did the results differ across experiments? In Exp. 1, NP-type did not affect the maximal number of similar adjacent items. (2 pronouns vs 2 FNPs). In

Exp. 2, NP-type did affect similarity (2 pronouns vs. 3 FNPs - because N4 was a FNP). Thus an effect of NP-type only arose when similarity increased, suggesting that the effect of NP-type in Exp. 2 was due to interference in working memory. In line with this analysis, the disadvantage in the FNP condition was numerically twice as large when all three FNPs shared the same determiner as when they did not (114 ms vs 60 ms), suggesting a sensitivity to surface form.

## Summary

In English, but not in Dutch, making the innermost subject a pronoun affects processing difficulty. The dependence on position and contrast with Dutch suggest that this effect involves factors other than interference in working memory. In contrast, the effect of N2- and N3-type observed in Exp. 2 of the Dutch study can be accounted for by interference in working memory.

### 12.1.5 The RC/RC V2-drop effect

In the following, I will refer to the verbs of the inner RC, the outer RC, and the main clause as V1, V2, and V3, respectively. If V2 is omitted, an RC/RC seems as, or more, acceptable than the grammatical version [Gib99]:

20. The man that the dog that Mary adopted screamed.

This effect is specific to V2; if V1 or V3 is dropped, the sentence is not acceptable. However, if V2 is part of a right-branching RC, V2 cannot be acceptably dropped [Gib99]:

21. I know the man that the dog that Mary adopted bit.

I know the man that the dog that Mary adopted.

### 12.1.6 V2-drop x N3-type Interaction

I was curious whether the V2-drop and N3-type effects for English RC/RCs would interact. It may not be felicitous to drop V2 when N3 is a pronoun. To test this, I performed a self-paced reading study in which N3-type (first-person pronoun, third-person pronoun with a referent, or name) was crossed with grammaticality (V2 present or not) [Whi04d]. Thus stimuli were of the form:

22. [According to Sue], The/the trophy that the athlete that I/Sue/she admired greatly [won at the track meet] was stolen from the display case.

The preamble, *According to Sue*, was only present for the third-person pronoun conditions.

An statistically significant interaction between N3-type and grammaticality was found in the region of the final verb phase. For the grammatical sentences, there was a slow-down for the name condition relative to the two pronoun conditions. In contrast, for ungrammatical sentences, the name condition was numerically faster than the pronoun conditions. This reversal indicates that V2-drop was felicitous when N3 was a name, but not when it was a pronoun. Thus, these results show a non-local effect of N3-type. That is, the nature of the subject of the inner RC affects the processing of higher-level clauses (outer RC and main clause.)

In contrast to off-line complexity ratings [War02b], there was no difference in performance in the verbal regions for the first-person versus the third-person pronouns. Thus the increased off-line complexity ratings for third person pronouns may reflect an overall increase in difficulty related to binding the pronoun to its referent or to not having a referent. The present results indicate that the integration of subjects and verbs is unaffected by the type of pronoun.

### 12.1.7 Summary

Studies of consecutive NPs have indicated that processing difficulty increases as the number of similar NPs increases, and as the proximity between those items increases [Lew02, Lee03, Kaa04]. Similarity seems to depend on both syntactic and surface features of the NPs [Lee03, Kaa04].

Cross-linguistic comparisons have shown that crossed-serial dependencies are easier to process than center-embeddings [Bac86]. In English, an NC/RC is easier to process than an RC/RC or an RC/NC [Gib98]. The processing of an RC/RC is facilitated when N3 is pronoun, and this effect seems to go above and beyond interference in working memory [War02a]. A similar effect does not arise for crossed-serial dependencies in Dutch [Kaa04]. In English, when N3 is a FNP, it is felicitous to drop V2 [Gib99], but when N3 is a pronoun, it is not [Whi04d].

## 12.2 Accounts

Next I review some proposals as to the source of these complexity phenomena. The first is a psycholinguistic model, while the following two are complexity metrics. Thus, none of these proposals are couched in terms of a neurobiologically plausible model. However, it is of interest to examine the ability of these approaches to account for the above data.

### 12.2.1 Vosse & Kempen [Vos00]

This is an implemented, localist model, which is based on a lexicalist grammar. Each word is associated with a lexical frame, which is a predefined, elementary syntactic tree. The model creates and operates over a network of nodes which represent connections between lexical frames. A lexical frame *A* can attach to



a lexical frame  $B$  when there is an empty slot in  $B$  that is of the same phrasal type as  $A$ . Thus there is no grammar per se. Rather lexical frames compete with one another for attachment sites. This allows potential attachments that grammar-based parsing systems would never consider.

The implemented model specifies the lexical frames and the dynamics of the attachment competitions. The issue of how attachments could actually be represented in neural tissue is not considered; rather the modeling is at a higher level. A sentence is parsed correctly if all the proper attachments are made, and no improper attachments are formed. Like humans, the system could not parse RC/RCs or RC/NCs. This failure arose because the verbs engendered competitions that could not be resolved, due to the number of potential attachment sites (arising from three subjects). In contrast, the system could parse NC/RCs. However, the given explanation of why an NC/RC is successfully processed (p. 124) is unclear, and further discussion with the authors has not clarified the matter [pers. comm.].

Unlike humans, the system was not sensitive to N3-type; a pronoun N3 in an RC/RC still led to parsing failure. Also, the system cannot explain the V2-drop phenomenon. If V2 cannot be attached, replacing V2 with the final verb would simply result in that verb not being attached. In contrast, humans appear to not expect V2, but to attach the final verb properly.

### **12.2.2 Interference in Working Memory**

Lewis [Lew96] notes that similarity of between items stored in working memory causes interference in a range of different modalities. He suggests that such interference may also apply to syntactic representations in working memory. Such an approach could account for the effects of similarity-based interference for NPs

held in working memory [Lew02, Lee03, Kaa04].

However, this approach cannot fully capture other aspects of complexity phenomena. In particular, Lewis suggests that it may not be possible to maintain three unattached subjects in working memory, due to their syntactic similarity. This would account for the difficulty of an RC/RC or an RC/NC. However, it does not explain the relative ease of an NC/RC or of crossed-serial dependencies in Dutch.

### 12.2.3 Dependency Locality Theory

Gibson and colleagues were responsible for elucidating many of the above complexity phenomena. Their extensive work in this area has led to the Dependency Locality Theory (DLT) [Gib00], which provides a distance-based complexity metric. It is based on the idea that complexity increases as the distance increases between two items that must be integrated together in the syntactic tree. Distance is measured as the number of new discourse referents that intervene between these items, where a new discourse referent is a tensed verb, or an NP that is not an indexical (first- or second- person) pronoun.<sup>1</sup> Integration cost is taken to increase with distance because the activation of the first item is taken to decrease as activation is redirected to new discourse referents; thus more energy is required to reactivate the first item during integration. A cost of 1 Energy Unit (EU) is generated for each intervening discourse element, and for generating the new discourse referent itself. Perceived complexity corresponds to maximal integration cost. For example, for an RC/RC construction such as :

---

<sup>1</sup>The discourse is presumed to always include a speaker and a listener, so pronouns referring to either do not introduce a new referent.

23. The vase that the man who Jen dated bought fell.

the highest cost occurs at the verb *bought*, which has a cost of 7 EUs: 1 EU for the construction of *bought* + 2 EUs for attachment to *man* (across *Jen* and *dated*) + 4EUs for co-indexing the gap following *bought* with the relativizer *that* (across *man*, *Jen*, *dated*, and *bought*). It is proposed that this high cost corresponds to the unacceptability of such a structure.

For an NC with a transitive verb, an RC/NC has a larger maximal cost than an RC/RC (due to integrating across an explicit object in the NC). However an NC/RC has a lower cost than an RC/RC because a long-distance integration of a gap across an embedded clause is not required. This accounts for the difference in difficulty between an RC/NC and NC/RC [Gib98, Gib00].

While the DLT can explain a range of complexity phenomena, it has difficulty in fully accounting for some aspects of the phenomena associated with double center-embeddings. Under the DLT's assumption that an indexical pronoun does not introduce a new discourse referent, integrating across such an entity does not generate any cost, accounting for the N3-type effect. This would imply that there should also be an effect of N3-type in crossed-serial dependencies. However, as we have seen, making N2 and N3 pronouns had no effect, contrary to the DLT prediction [Kaa04]. Furthermore, Warren and Gibson [War02b] tested the discourse-referent hypothesis, and did not get the predicted results. In this study, subjects read critical sentences in which an object-extracted RC modified the main subject. The subject of the RC was a definite NP. Whether or not this NP had a referent was manipulated in a contextual sentence presented just before a critical sentence. The presence or absence of a previous referent affected reading times at the RC's verb, but not at the main verb. That is, an NP that added a new discourse referent incurred a local cost (at its own verb), but did not

affect processing in the higher clause (at the main verb), contrary to the DLT's prediction. A local effect of discourse-referent processing could still potentially explain the N3-type effect in off-line complexity ratings. However, it can't account for the interaction of N3-type with the felicity of V2-drop [Whi04d], because this is a non-local effect concerning higher level clauses (the outer RC and the main clause).

The DLT has difficulty in accounting for the V2-drop effect itself. An earlier version of the DLT, the SPLT [Gib98], posited that complexity corresponds to the storage cost of syntactic *predictions*, not integrations. Under that metric, it was proposed that the parser drops the prediction for the outer RC's verb due to high memory costs [Gib99]. However, an assumption underlying the SPLT was contradicted by experimental evidence [Gib00, Gib04]; the SPLT was transformed into the DLT, where prediction cost is constant, and integration cost increases with distance. Hence, under the DLT, prediction cost cannot explain V2-drop, since the prediction cost for the outer RC and the inner RC are the same. While it's true that V2 induces the highest integration cost, this cost is incurred *after* the verb is encountered, and thus cannot account for dropping the prediction of that verb before it occurs. Furthermore, integration cost at V2 is independent of whether the outer RC is center-embedded or right-branching, but V2-drop is felicitous only when the outer RC is center-embedded [Gib99]. Thus, integration cost cannot account for the V2-drop effect.

The DLT also makes the wrong prediction about complexity in some important cases. The RC/RC's high cost results from the summation of the integration costs for the second verb and the first RC's gap. However, if these costs are decoupled, as in the following sentence:

24. The woman who the man who Sue dates flirted with hit him.

complexity is still very high, while the maximal integration cost is lower. Here the intransitive verb *flirted* signals that the gap for the first *who* is not in the object position. So the integration cost of *flirted* is only 3EUs. The integration cost of the gap following *with* is 4EUs, and the integration cost is of *hit* is 5 EUs. Thus the maximal cost is only 5EUs. However a much easier sentence like:

25. The fact that the man who Sue is dating rides a motorcycle scares her.

has a higher integration cost, of 6EUs (at *scares*).

This analysis depends on the assumption that integration of a gap is not attempted following an intransitive verb, as is consistent with studies on filler-gap processing for intransitive verbs [Bol91, Sus01]. However, if it were argued that such an integration is attempted and does incur a cost, this claim would then destroy the DLT account of the difference between an NC/RC and an RC/NC. That account hinges on the assumption that there is no long-distance integration of a gap across the RC for an NC/RC. However, there is evidence that the RC possibility for an NC is actively evaluated. In a potential NC, a manipulation of the potential filler's appropriateness as the verb's object had an effect at the verb, indicating that the possibility of a gap is actively considered [Pea98]. Thus, if it were argued that an integration cost for a possible gap is incurred at an intransitive verb, such a cost would surely also apply to a potential NC's verb. However, in that case, there would be no difference in integration cost for an NC/RC versus an RC/NC. Nor could it be argued that the possibility of a gap in an NC is dropped in a potential NC/RC due to increased complexity; this would incorrectly predict that a potential NC/RC which turns out to be an RC/RC, such as (26), is uninterpretable.

26. The proposal that the student who Bill advises made at the meeting im-

pressed everyone.

Another incorrect prediction occurs for Japanese. A sentential complement within a sentential complement (SC/SC) of the form:

27. NP-nom [NP-nom [NP-nom V1 Comp] V2 Comp] V3

has its highest integration cost at  $V3 = 5\text{EUs}$ . An SC of the form:

28. NP-nom NP-dat [NP-nom NP-dat NP-acc V1 Comp] V2

has its highest integration cost at  $V2 = 6\text{EUs}$ . However, the SC is easier than the SC/SC [Bab99].

#### **12.2.4 Summary**

We have seen that none of the above approaches can fully account for the data. Vosse & Kempen's model [Vos00] replicates some complexity phenomena, but cannot explain the V2-drop or N3-type effects. The proposal of interference in working memory [Lew96] cannot explain the pattern of an NC/RC versus an RC/RC or an RC/NC. The DLT metric [Gib00] is based on the distance between items that must be integrated together. It is currently the leading account of complexity phenomena. However, it cannot account for the V2-drop effect, the interaction of the V2-drop effect with N3-type, the lack of N3-type effect in crossed-serial dependencies, and the difficulty of an RC/RC when V2 is an intransitive verb.

## Chapter 13

### Parsing Models

In this chapter, I review those models that deal more directly with parsing and hierarchical representations. The desiderata for such a model are as follows.

- Neurobiologically plausible hierarchical representation of thematic roles (thematic tree).
- Thematic tree should be suitable for long-term storage.
- Neurobiologically plausible working-memory representations that support construction of the thematic tree.
- Explanation of similarity-based interference in working memory.
- Parsing algorithm for using working-memory representations to construct thematic tree.
- Parsing algorithm should account for all complexity phenomena not explained by similarity-based interference.

First I consider possible solutions to general problems related to representing the thematic tree. Then I review various parsing models. In each section, I discuss how well these models and theories meet the above criteria.

## 13.1 Representation of the Thematic Tree on a Computer

I start with a discussion of how a thematic tree would be represented on a computer, and which aspects of such an encoding are neurally plausible and which are not. It is hoped that such a discussion will illuminate the difficulties involved in formulating a neurally plausible representation of the thematic tree.

### 13.1.1 How

Computer memory can be conceptualized as an array of registers. A memory address is associated with each register, where memory addresses systematically increase as array position increases. An address allows access to a particular register. High-level computer languages allow a variable name to be mapped onto a memory address. (This mapping is done automatically by the compiler.) Thus items can be stored and retrieved from memory based on variable names.

The most fundamental requirement of the thematic tree is that words are associated with thematic roles. On a computer, this is accomplished by creating variables and setting those variables to certain values. For example, an *Agent* variable could be set to a pattern that encodes *Mary*. Thus some memory register is labeled *Agent* and set to a particular value, which represents *Mary*.

To represent a hierarchy, it must be possible to combine multiple bindings together into a unit, and to refer to that entire unit. A high-level computer language allows a data structure, called a record, which groups different items together. For example, a record might consist of *Theme*, *Agent*, and *Verb* variables. The compiler maps these variables to consecutive memory addresses. Thus variables are grouped together by putting them next to each other in memory.

The memory address of the first variable can then be used to refer to the



entire entity. This is known as a *pointer*. Thus, a variable could take a pointer as its value, indicating that the value corresponds to the entire unit starting at that memory address. For example to encode (Agent = Mary, Verb = knows, Theme = (Agent = Ted, Verb = likes, and Theme = Sue)), two records are created, each having the *Agent*, *Verb*, and *Theme* variables. Call one record *Main*, and the other *Sub*. The variables in *Sub* would be set to the corresponding values from the embedded clause (i.e., Ted, likes, and Sue). The *Agent* and *Verb* in *Main* would be set to their corresponding values from the main clause, while the *Theme* would be set to the address of *Sub*. See Figure 13.1.

In summary, binding is done by assigning a particular pattern to a particular memory address. Hierarchy is created by assigning variables to consecutive memory addresses, and referring to the resulting unit by its memory address.

Note, however, that the two kinds of structure building operations - binding and hierarchy formation - are not necessarily logically different. A binding is an association of *terminal* items - a word and a role. Hierarchy is created by associations of bindings with other bindings. Thus, in both cases, two or more items are associated together. In a computer, the basic binding operation corresponds to assigning a pattern to a memory location. This operation cannot be directly recursively applied, because it is not possible to physically assign one location to another location. Rather, to perform higher-level associations, a location is referred to by its memory address. Thus, there is a dichotomy between the way in which a basic binding is implemented and a hierarchy is formed. This dichotomy arises because one component of a basic binding is a physical location; this forces a different way of associating bindings with each other, based on referring to a location by a unique identifier (its memory address.)

1200	Mary
1232	knows
1264	1392
1296	
1328	
1360	
1392	Ted
1424	likes
1456	Sue

Figure 13.1: Example of encoding *Mary knows that Ted likes Sue* in computer memory. The left column represents memory addresses, which systematically increase. The right column represents registers. The programmer would declare a record having *Agent*, *Verb*, and *Theme* variables. For each instance of this record the compiler would map these variables onto specific consecutive addresses. Here the record *Main* starts at 1200 and the record *Sub* starts at 1392. The value of *Main*'s *Theme* variable is a pointer to *Sub*. *Mary*, *knows*, *Ted*, *etc.* correspond to numbers that have been associated with each token. (For simplicity, the problem of how to determine whether a register's value should be interpreted as a memory address is ignored. )

### 13.1.2 Difference from Neural Networks

In a computer, a central executive governs serial access to memory. In contrast, in a biological neural network, there are many, simple, massively interconnected processing units. Of course, it would be possible to construct a computer-like memory in an artificial neural network. For example, a set of nodes could be wired together to form register-like group, and such registers could be wired together to form record-like units. Each unit could have an identifying number associated with it (perhaps coded within its connection weights), that would act like a memory address. Thus values could be filled into the record-like units, and the identifier of a unit could be used like a pointer to link together different units. Marcus [Mar01] has proposed such a scheme, where each record-like unit (called a treelet) encodes hierarchical relationships between the registers. However, given its massive parallelism, it seems highly unlikely that the brain emulates a computer-like architecture.

It is plausible that the basic binding operation could be performed in the same way as a computer. That is, a group of nodes could encode a particular role, and a pattern across those nodes could represent the value. For example, a certain group of nodes could be used to represent the *Agent*, where the activation pattern across those nodes could encode *Mary*. A different set of nodes could encode the *Theme*, etc.

However, this computer-like approach breaks down when it comes to encoding hierarchy. Without memory addresses, it is unclear how items can be grouped into a unit. Two basic approaches have been proposed: combining activity patterns to yield a new pattern, or inducing correlated firing between two patterns.

## 13.2 Possible Neural Network Representations of the Thematic Tree

### 13.2.1 Production of a New Pattern

One approach to the binding problem is to represent each item by a large vector (i.e. a distributed activation pattern over  $n$  nodes), and to define operations which combine two or more vectors to yield a new vector (activation pattern). This new vector could then be combined with other vectors to produce a hierarchical encoding.

Touretzky and Hinton proposed a scheme based on the outer product of the two vectors [Tou88]. (That is, the resultant vector is comprised of all pairwise products between the items in the two vectors.) However, the size of the resultant vector is the product of the dimensions of the constituent vectors, giving an unbounded increase in size as more and more bindings are performed. Instead, to avoid exponential explosion and to allow calculations to be performed iteratively over a fixed set of cells, the combinatory operation should yield a vector that is the same length as the constituent vectors. Thus the combination is a reduced representation (RR) of the constituent vectors [Hin90].

### Reduced Representations that are Learned

Pollack [Pol90] proposed a scheme wherein the reduced representation is comprised of the hidden units' activations in a network trained by back-propagation to auto-associate. See Figure 13.2. Rohde [Roh02] used a similar approach in a system which developed an RR representation of the syntactic structure of a sentence. This encoding could be queried to yield the relationships specified by

the sentence. Such an approach has the advantage that the rules of processing (the grammar) are learned along with the representations. However, we will see below that such an architecture is not actually robust enough to parse and encode arbitrary hierarchical structure.

## Reduced Representations based on Statistical Properties

A different approach is to predefine combinatory vector operators with the desired properties. Under this method, each item vector is large (dimension 1,000 to 10,000) and satisfies certain statistical properties; the combinatory operators rely on these statistical properties. Thus item representations do not directly encode any semantic information about an item, but rather act as an abstract representation that allows combination with other items. Plate [Pla95] has proposed a binding scheme for real-valued vectors, based on the convolution of their outer product. Kanerva [Kan95] has proposed a scheme that operates in a bit-wise fashion over binary vectors, where each element has an equal probability of being 0 or 1. Because Kanerva's scheme is simpler, we will focus on it.

Both proposals employ two different combinatory operators, corresponding to the bind and group (merge) operations. Let '@' represent the binding operation, and '+' represent the grouping (merge) operation. In Kanerva's scheme, the bind operation is bit-wise exclusive-or. That is, the two constituent vectors are aligned; at each position, if only one element is a 1, the result is a 1; otherwise it is a 0. See Figure 13.3. Merging is implemented as a normalized sum of the constituent vectors, by taking a bitwise majority. That is, at each position, if there are more 1's than 0's the result is a 1; otherwise it is a 0. Ties (which could arise for an even number of constituent vectors) are broken probabilistically, with equal chance of giving a 0 or 1.

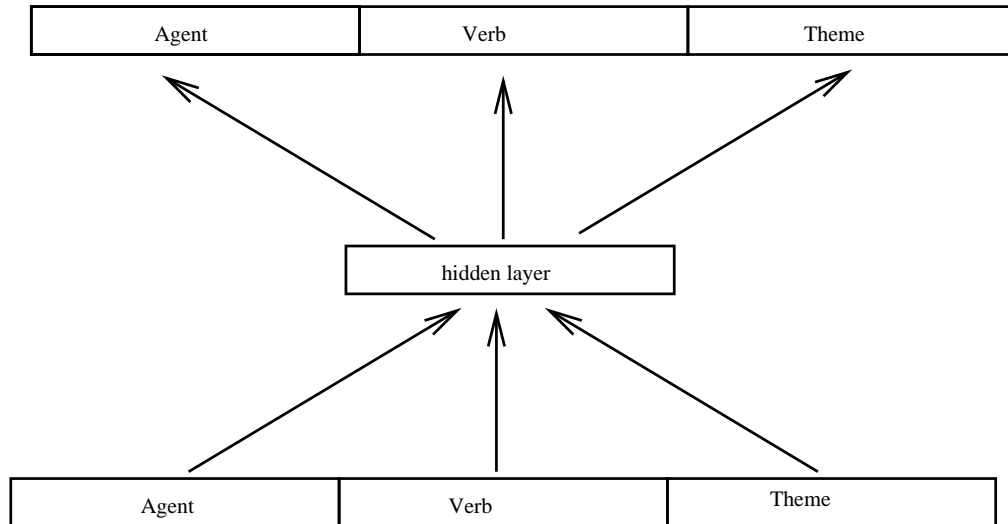


Figure 13.2: Example of network that learns to form an RR encoding. Each box represents a group of nodes of the same size, and each arrow represents full interconnectivity between two groups of nodes. For each training item, the input and output layers are set to the same value. Using the back-propagation training algorithm, the network learns to recreate the input on the output layer. As a result, the hidden layer (in conjunction with the learned weights) forms a condensed representation of the input. This condensed representation could then be used as one of the values on the input layer. For example, in the *Mary knows Ted likes Sue* example, the patterns for *Ted*, *likes*, and *Sue* would first be activated over the corresponding sets of input nodes. The resulting pattern on the hidden layer constitutes an RR encoding of this information. Then the input layer is set to *Agent = Mary*, *Verb = knows*, and *Theme = the hidden layer pattern*. The new hidden layer pattern then represents the encoding of the entire sentence. Such an encoding is decoded by activating the pattern on the hidden layer to get the component values on the output layer. An output item that is itself an RR encoding can then be fed back to the hidden layer again to be decoded.

$$\begin{array}{r}
011010\dots \\
@110011\dots \\
\hline
101001\dots
\end{array}
\qquad
\begin{array}{r}
011010\dots \\
110010\dots \\
+110011\dots \\
\hline
110010\dots
\end{array}$$

Figure 13.3: Example of bind and merge operations.

Of course, inverse operators must also be specified, so that information can be extracted from an RR encoding. Because composition of two vectors yields a vector in the same representational space, there is compression of the constituent vectors, thereby introducing noise. In order to clean up noisy vectors, it is assumed that there is an item memory which stores the patterns of all base vectors. Such a memory could be based on an associative recurrent network. When presented with a vector, the item memory activates any vector that has a similarity measure above some threshold. Given two vectors, similarity is measured as the fraction of elements that have the same value in both vectors. For unrelated vectors, this measure has the expected value of 0.5. That is, because each bit in a vector has equal probability of being a 0 or 1, the probability that the corresponding bit in another unrelated vector has the same value is 50%. The merge operation yields a result that is similar to its constituent vectors. For example,  $a + b$  gives a similarity with  $a$  of .75 and with  $b$  of .75. In contrast, the bind operator yields a result that is not similar to its constituent vectors.

Because merge yields a vector similar to its constituents, the unmerge operation is performed by comparing a vector to item memory, to retrieve all similar vectors. The unbind operator is the same as the bind operator - bitwise exclusive-or. When this operation is used for unbinding it will be represented as '#'. An unbind is then followed by a comparison to item memory. For example, consider

the vector  $a@b + c@d$ . It is the case that exclusive-or distributes over the merge operation.<sup>1</sup> Thus, unbinding with  $b$  would give  $a@b\#b + c@d\#b = a + c@d\#b$ . The resulting vector is similar to  $a$ . It is also similar to  $c@d\#b$ , but that vector is not stored in item memory. Thus, unbinding with  $b$  and comparing with item memory gives  $a$ , as desired.

Systematic application of the compositional operations allows encoding of hierarchical structure. First let's consider why two operators are necessary. In order to encode structure, an operator cannot be associative. That is, for some operator  $\odot$ , it should not be the case that  $(a \odot b) \odot c = a \odot (b \odot c)$ , because if this were the case, grouping information would be lost, so hierarchy could not be represented. The merge and bind operators are both associative, so neither one alone can create a structured representation. However, it is the case that  $(a + b + c)@d \neq a + b + (c@d)$ . This can be seen by expanding first expression to  $a@d + b@d + c@d$ , which does not equal  $a + b + c@d$ . Thus the combination of these two operators allows grouping information to be encoded.

How can these operations be used to represent the thematic tree? Under such an encoding scheme, a role is not represented by a certain group of nodes, but rather is also represented by a pattern. The representation of our favorite example could be *Agent @ Mary + Verb @ knows + Theme @ (Agent @ Ted + Verb @ knows + Theme @ Sue)*. Here each constituent item is a large, binary vector with the property that each element has an equal probability of being a 0 or a 1. The whole expression denotes another vector with the same dimension and properties. As discussed above, an item vector does not encode any lexical

---

<sup>1</sup>That is,  $a@(b + c + d) = a@b + a@c + a@d$ . When an odd number of items is being merged, this equality is exact. For an even number of items,  $a@(b + c) \approx a@b + a@c$ , because noise is introduced during each merge (to break ties).



information about that word. Rather, an item vector could be associated with a particular word via corresponding connection weights which allow activation of a neural assembly which does encode semantic and phonological information about that word.

As more and more combinatory operations are performed, the resulting representation becomes noisier and noisier, until it may become impossible to reliably extract the encoded information. This problem can be solved by recording intermediate results. In the above example, the vector *Agent @ Ted + Verb @ knows + Theme @ Sue* could be saved in item memory.

### 13.2.2 Temporal Encoding

An alternative possibility is to represent relationships by the timing of firing of the constituent items. Shastri [Sha93, Sha99], and Hummel and Holyoak [Hum97] have used synchronous firing to encode thematic binding relations for certain classes of propositions/sentences. For example, Mary is encoded as an Agent by having the nodes representing Mary and Agent fire together. Other pairings fire synchronously during other time slots. See Figure 13.4.

Electrophysiological studies have produced evidence that the brain may indeed rely on a temporal synchrony for some types of binding. Various studies on the visual system have shown that cells representing the features of a single object synchronize, while those representing features of different objects do not. Such low-level synchronization occurs in the gamma range [Eng01]. It is unclear if synchronization plays a role for higher level processing, although brain-imaging evidence suggests that verbal working memory relies on a temporal encoding involving oscillatory activity in the theta range [Kli96, Kli99, Rag01, Jen02]. Accordingly, it has been suggested that different types of binding rely on different

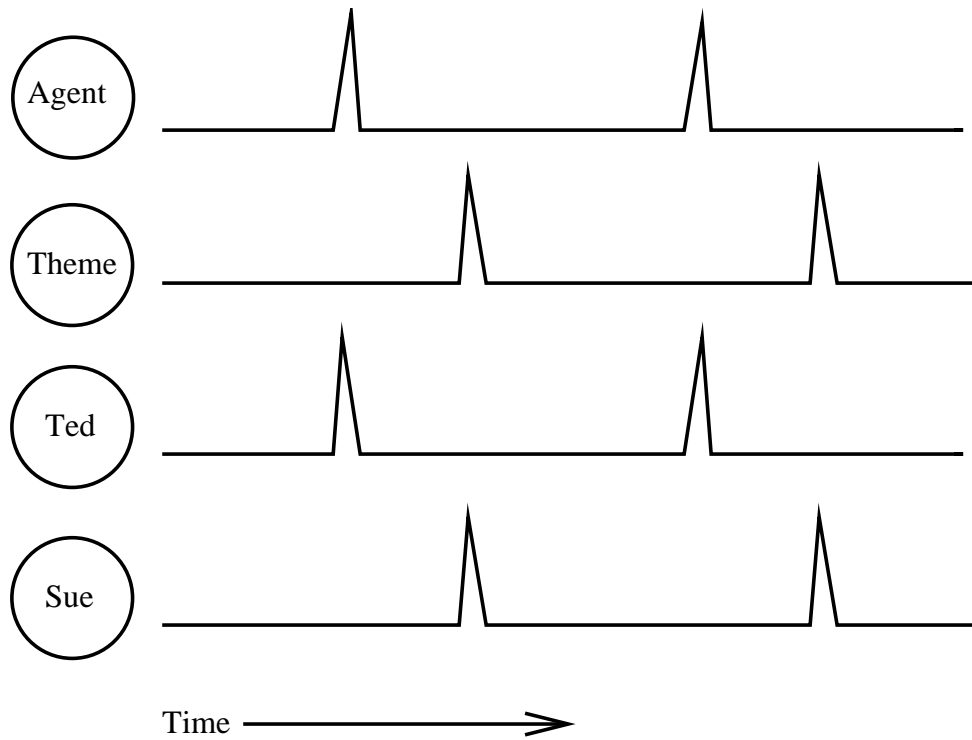


Figure 13.4: Example of temporal encoding of  $Ted = Agent$  and  $Sue = Theme$ . The lines to the right of each node represent the firing pattern for that node. For simplicity, each word and role is represented here as a single node. However, the same type of encoding could be used for a distributed representation of each item.

frequencies, with low-level sensory binding occurring the gamma band, access to distributed representations in semantic long-term memory relying in the alpha band, and encoding in working and short-term memory occurring in the theta band [Kli96, Kli99].

However, a weakness of the temporal approach for a parse representation is that time is linear. It is not clear how to map hierarchical structure onto a linear encoding in a general manner. While one frequency could be nested inside another, it seems that there would be an insufficient number of possible harmonics to represent complex syntactic structure. Another possibility, adopted by [Hum97], is for all of the nodes along some path of the tree to fire synchronously. Each path fires in a different temporal slot, until the whole tree is traversed. However, this raises the issue of how the cells representing the nodes are recruited, and how proper timing is coordinated among these cells. In the Hummel and Holyoak model, timing was coordinated by excitatory and inhibitory connections. Thus the desired tree had to already be directly encoded by the connectivity between cells in order to instantiate the temporal encoding. [Sha93] assumed that temporal pairings could be generated on the fly. However, they did not address how this was accomplished, nor how nested relationships would be represented.

Thus, no one has shown how a temporal encoding of hierarchical structure could be generated on the fly in a general way. This requires solving two problems: (1) how syntactic structure is mapped onto the temporal encoding; (2) how this temporal encoding is activated during processing. Another problem is that a temporal encoding is not directly suitable for long-term storage. However, storage could be based on a record of information that would allow a temporal encoding to be re-instantiated.

### 13.2.3 Summary and Conclusions

Due to the distributed nature of neural representations, it is likely that bindings are represented by associating patterns of activity. There have been two types of proposals as to how this may be accomplished.

- In an RR encoding, patterns are combined via the creation of a new pattern. This approach seems well-suited to representing hierarchical structure. It also provides a representation suitable for long-term storage, as the resulting pattern can be stored via connection weights.
- In a temporal encoding, patterns are associated by correlated firing of the activated nodes. It is difficult to see how to encode arbitrary hierarchical structure in such a framework, and the resulting representations are not directly suitable for storage in long-term memory. However, brain imaging evidence suggests that verbal working memory employs a temporal encoding (based on a theta-band carrier wave), and that the same type of working-memory representation may be used during sentence processing.

Thus, an RR encoding seems more suitable for representing the thematic tree, but experimental evidence indicates that a temporal encoding may be employed during sentence processing. However, recall that it is not sufficient to simply specify how relationships are represented in a thematic tree. It is also necessary to specify how the thematic tree is constructed from a sequence of words. To process center-embeddings and crossed-serial dependencies, stack-like and queue-like representations are required in working memory. Note that such working memory representations are separate from the thematic tree. Thus, it may be the case that working memory relies on a temporal representation which subserves the construction an RR encoding of the thematic tree.

Furthermore, a temporal encoding is more appropriate than an RR encoding for intermediate representations. In an RR encoding, the constituent vectors lose their identity, because they are combined into a new pattern. In contrast, in a temporal coding, the constituents retain their individuality and remain directly accessible. Because the purpose of working memory is to provide access to individual, previously processed constituents (e.g., unattached subjects resulting from center-embedded clauses), a representation that retains their individuality is desirable. Furthermore, we have seen from the LPE model how to represent order information temporally. Given that processing of center-embeddings and crossed-serial dependencies relies on order information, such a representation may be well suited for intermediate representations.

In sum, an RR encoding is well suited for the long-term storage of arbitrary hierarchical structure. A temporal encoding is well suited for representing individual, ordered constituents in working memory, and is consistent with brain imaging data.

### **13.3 Parsing Models**

In the previous section, I reviewed how a thematic tree might be represented. In the following, I review model that deal more directly with the parsing process - how a sequence of words is converted into the thematic tree.

#### **13.3.1 SRNs**

A recurrent neural network has feedback connections. (See Figure 13.5.) Therefore, information from one time step of processing can influence a subsequent time step. Such feedback connections essentially provide a memory that

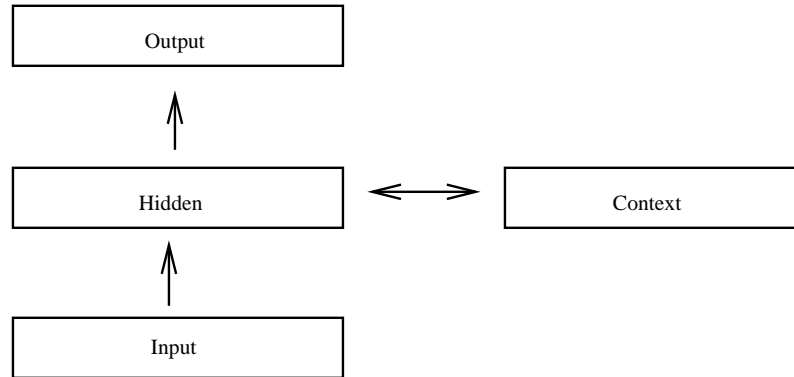


Figure 13.5: Architecture of a recurrent network. The hidden units connect into the context units, which feed back to the hidden units. Thus the hidden units' previous activations can affect their subsequent activations.

can be used to process time-varying input, such as a sequence of words comprising a sentence. Elman [Elm90] showed that Simple Recurrent Neural networks (SRNs) trained on sequences via a variant of the back-propagation algorithm can learn to predict the next item in the sequence. Thus, the network learns the structure of set of sequences (i.e., a grammar).

Although some have hailed such results as demonstrating the ability of SRNs to handle natural language [Chr99], such a conclusion is clearly unwarranted. It is insufficient to simply predict upcoming words. Rather, as words are encountered they must be integrated into a representation of the sentence's meaning. Thus some researchers have created parsing systems that use an SRN-like network to learn grammatical rules, supplemented by another network which allows hierarchical representations.

One approach used a temporal specification of the syntactic tree, in which each constituent was assigned to a separate phase, and the relationships between constituents were encoded by the firing of Parent, Sibling, and Grandparent nodes [Lan01]. However, to represent the structure of a complex sentence, a large

number of distinct phases would be required, which is an unrealistic assumption.

In another approach, the meaning of a sentence was represented as a list of three-part propositions, and a subset of the system was first trained to create a distributed (RR) encoding of any sequence of propositions [Roh02]. For example, the encoding of:

29. Jim knows Sue likes big dogs.

is:

- (knows, agent, jim)
- (knows, theme, likes)
- (likes, agent, sue)
- (likes, theme, dogs)
- (dogs, mod, big)

Here, each word denotes a distributed encoding, based on semantic features. Note that this representation does not directly encode hierarchical structure; rather, structure is inferred from matching words. However, if a word is repeated in a sentence, this will lead to ambiguity. For example, the encoding of

30. Jim knows Sue knows Don.

is:

- (knows, agent, jim)
- (knows, theme, knows)

- (knows, agent, sue)
- (knows, theme, don)

This encoding could mean *Sue knows Jim knows Don*. This problem could be remedied by associating an identifier with each instance of a word, requiring an additional mechanism. In contrast, if hierarchy is directly represented, this is not a problem, as repeated words are differentiated by their positions in the hierarchy.

Once the RR encoding mechanism was trained, the system was then trained to produce such an RR encoding for a sentence, as follows. The propositions representing the meaning of a sentence were fed through the system to get their RR encoding. Then the words in that sentence were presented, and the SRN part of the system was trained to produce that RR encoding in response to that sequence of words. Following training on a wide range of sentences, the representational ability of the system was tested by presenting a novel sentence, and querying the resulting RR encoding on each proposition comprising the meaning of that sentence.

While the model showed good generalization abilities for simple sentences, performance rapidly deteriorated for more complex structures. For example, the average error rate for a sentence with six propositions was approximately 10%, according to the less stringent multiple-choice criterion. That is, the response to a query was counted as correct if it was closer to the correct response than to other distractor items. This error rate is per proposition, so the probability of correctly encoding the entire sentence is  $(1.0 - 0.1)^6$ , about 50 %. For eight-proposition sentences, the error rate was about 20%, so the probability of correctly representing a sentence was only about 15%.

Error rates on center-embedded structures were particularly high. For a four-



proposition sentence having a subject-modifying RC, the error rate was 25%, so the probability of correctly encoding the entire sentence was about 30%. In contrast, humans can act out the meaning of such sentences with 90% accuracy [Cap96]. Thus, these simulations have not demonstrated that such a connectionist system is capable of developing representations than can parse and encode complex syntactic structure.

While the inability of recurrent networks to handle center-embeddings has been touted as a desirable feature because humans also have difficulty with center-embeddings [Chr99, Ore00], we have seen that some doubly center-embedded clauses are actually rather easily processed by humans [Gib98].

Thus SRN-based parsers have not demonstrated the strong generativity necessary for handling natural language. This problem is especially acute for center-embeddings, but is also present for right-branching structures. Because such systems do not explicitly model variables and recursive processing, embeddings cannot reliably be processed.

### **13.3.2 LTSMs**

Some researchers have investigated networks more suitable for processing center-embeddings. The primary reason that an SRN has difficulty in processing center-embeddings is that previous information becomes more and more degraded at each time step. This degraded information cannot then sufficiently influence the error measures that drive learning in the back-propagation algorithm. Thus, the network has difficulty in learning the correspondence between widely separated items (such as the main subject and main verb in a sentence with a center-embedded clause). One way to solve this problem is to provide separate, gated blocks of context units (i.e., registers). That is, each register has its own gating

network. When the gating network is “open”, information can be written to a register. When it is “closed”, the information in a register cannot be overwritten, but register activations can still drive the learning process. Thus information can be held over time without becoming degraded. This architecture is called a *Long Short-Term Memory* (LSTM) network [Hoc97].

Gers and Schmidhuber showed that an LSTM network can learn to predict the upcoming token for strings of the form  $a^n b^n$  [Ger01] (corresponding to center-embedded clauses, as discussed in section 11.2). The network could generalize to larger values of  $n$  that it had been trained on, because it had learned to implement a counter in a register. For each  $a$ , activation within the register increased by a fixed amount. When a  $b$  occurred, activation was decreased by the same amount. Thus, an end-of-string token was predicted when the counter reached 0.

Again, this system makes predictions, without forming a structured representation. To create a structured representation, a counter would not be sufficient. Rather the system would have to also learn to store previous  $a$  values, to be integrated with the appropriate  $b$  values to form some type of hierarchical encoding. For example, each  $a$  value could be stored in its own register. If this approach were taken, the network would not be able to generalize to higher values of  $n$  than it had been trained on, because it would have to learn to store each  $a$  in a particular, separate register. As two center-embeddings does seem to be the limit for humans, this is not necessarily a problem. However, it is unclear how such a system could account for the complexity phenomena. For example, if such a system could parse an NC/RC it would probably be able to parse an RC/NC without difficulty, contrary to human performance.

Furthermore, parsing based on a counter mechanism is suspect. Linguists have noted that natural languages are all based on structural grammatical rules,

rather than numeric rules [Cho80, Ber84]. For example, no language has a rule that applies to the  $n$ th word of a sentence.

The SRN and LSTM have the advantage that grammatical rules are learned, based on the input characteristics. However, linguists have forcefully argued that children possess grammatical knowledge that cannot be derived solely from the statistics of the input [Cho59, Jac02]. It may be the case that an LSTM-like system in the brain learns to parse language based on some pre-existing linguistic primitives. For example, the ability to manipulate stack-like representations may be innate; an LSTM-like system may then learn what operations to perform when.

O'Reilly and Frank [Ore03] have proposed a LSTM-like system which uses neurobiologically plausible supervised and reinforcement learning. The reinforcement learning part of the system (modeled after the basal ganglia) learns to gate relevant information into registers, while the supervised learning part of the system (modeled after the prefrontal cortex) learns what transformations to perform on that information. Such a system has good potential for learning grammatical rules (which would operate over pre-existing linguistic representations).

### 13.3.3 Pulvermuller

The previous models used distributed representations, and learned from examples. The present model uses localist representations, with hand-coded weights and activation functions. In [Pul03], Pulvermuller presents a proposal for how a grammar could be neurally implemented. This proposal is based on a set of pairwise sequence detectors. A sequence detector is activated by A followed by B, not B followed by A, where A and B are syntactic categories activated by word nodes. This mechanism is based on nodes that support differing activation states (inactivation, primed, ignition, and reverberation) and on connections of differing

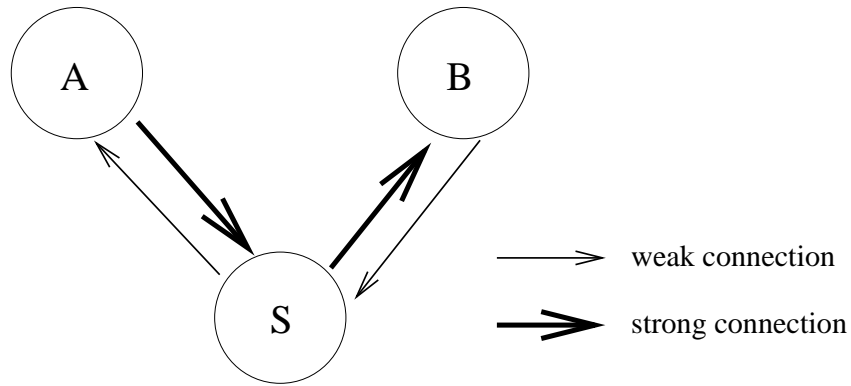


Figure 13.6: Example of detector S which recognizes sequence A B, from [Pul03].

strengths (strong, weak). Essentially, external input leads to ignition and then reverberation. Nodes that receive reverberating input from strong connections enter the primed state. If a node is already in a primed state, and receives a second volley of reverberatory input (over any connection strength), it too ignites. These dynamics allow the sequence node S in Figure 13.6 to become ignited for A then B, but not B then A. If A is activated first, node S is primed, and then receives reverberatory input when B is activated, leading it to ignite. However, if B is activated first, node S is not primed (due to the weak connection), and then A fails to ignite S.

Pulvermuller proposes that grammars are based on sets of such pairwise sequence detectors. He also assumes that there are different levels of reverberation, which could be used to implement stack-like processing. Thus this work focuses on how word sequences could be recognized. Given the localist representations assumed in the model, it is difficult to see how a neurally plausible representation of hierarchical structure could be obtained.

### 13.3.4 Summary

These models have focused on demonstrating the capacity to parse to natural language, and cannot account for the detailed pattern of complexity phenomena discussed in the previous chapter. In Rohde's model [Roh02], the thematic tree was represented as an RR encoded list of triplets, where the RR encoding was developed via learning (back propagation). The grammatical rules for constructing this representation from a string of words were also learned. However, the trained network could not reliably process complex sentences, showing particular problems with center-embedding. Learning of grammatical rules based on a LSTM-like model [Hoc97, Ore03] could potentially produce better results. In order to allow strong generativity, such learning should be based on grammatical primitives. Localist models have not demonstrated how hierarchical structure could be represented in a neurally plausible fashion [Vos00, Pul03].

## Chapter 14

### The TPARRSE Model

Having discussed the neurobiological and experimental constraints on parsing, and related research, I now turn to my proposed parsing model. As we have seen, there are actually three somewhat independent aspects to parsing: (1) the rules which determine what to do with an incoming word, based on the previously processed words and the grammar of the language; (2) the working memory representations that support the application of the grammatical rules to form the thematic tree; (3) the representation of the thematic tree itself. As discussed previously, the model focuses on neurobiologically plausible accounts of (2) and (3). For now, the algorithm that operates over these representations is considered at a symbolic level. Also as discussed, this is a theoretical model. It has not been implemented in full. Rather the representations and connectivity are specified based on computational principles. I start with a brief overview of the model, and then specify the model in detail.

As discussed in section 13.2.3, an RR encoding is suitable for representing the thematic tree, in that it allows representation of hierarchy, and is suitable for long-term storage. However, it is less suitable for a working memory representation, because individual constituents are no longer directly accessible, due to the distributed nature of the representation. In contrast, a temporal encoding retains

the individuality of constituent items, but is less suitable for representing hierarchy and for long-term storage. Therefore, a temporal encoding is more suitable for the working memory encoding. This assumption is in line with EEG evidence for oscillatory phenomena associated with holding items in verbal working memory.

Thus, the model proposes a dual representation - a distributed RR encoding of hierarchical structure which is generated from a temporal working-memory representation. The model consists of the following specifications:

- The basic RR encoding operations (RR primitives)
- How the thematic tree is represented via RR primitives
- The basic WM operations (WM primitives)
- How syntactic information is recorded via WM primitives
- The parsing algorithm that operates over the WM primitives to create the RR encoding of the thematic tree.

The resulting model is dubbed TPARRSE (Temporal Parsing And Reduced Representation Semantic Encoding).

The underlying principles of the model are best presented incrementally. For unambiguous, right-branching sentences, the thematic tree can be produced without reliance on the temporal WM representation. Therefore, I first present the RR encoding, and the portion of the parsing algorithm that handles right-branching sentences. I then present the temporal WM encoding, and the full parsing algorithm.

## 14.1 RR encoding

Unlike models in which the system is trained to form an RR encoding, I assume that the RR primitives are innate. This has several advantages. (a) It provides a systematic way of representing structure. Therefore, it is possible to encode any complex set of relationships, because the combinatory operators can reliably be recursively applied. (b) Innate combinatory operations allow different brain areas to use a uniform representation. It is a general feature of neural processing that task performance is distributed across different brain areas. If each brain area involved in parsing were to develop its own RR encoding, this would make communication between areas much less efficient. (c) Specific properties of the RR encoding can be exploited in the parsing algorithm.

Therefore, I assume that the thematic tree is represented via an RR encoding such as those proposed by Plate and Kanerva, as discussed in section 13.2.1. In Chapter 15, I present an example using Kanerva's system. This system was chosen because it is simpler. However, the general properties of the primitives are common to both systems. The parsing model is based on these general properties, as described next.

### 14.1.1 Primitives

Each terminal item (word, morpheme, or thematic role) is represented by a large vector with certain statistical properties. There are two combinatory operations: bind (@) and merge (+). Merge creates a vector that is similar to the constituent vectors, while merge creates a vector that is not similar to the constituent vectors. In the following, vectors will be given in **boldface**.

There is an item memory which records the identity of all terminal items.



When a vector is compared to item memory, item memory returns all vectors which are more similar to the comparison vector than would be expected by chance. Thus unmerging is performed by comparing a vector to item memory.

In addition there is an unbind operation ( $\#$ ), such that  $\mathbf{a}@\mathbf{b}\#\mathbf{b} = \mathbf{a}$ . Under the decoding operations, bind distributes over merge. That is, using unbind and unmerge to decode  $\mathbf{a}@(\mathbf{b}+\mathbf{c})$  gives the same result as decoding  $\mathbf{a}@\mathbf{b} + \mathbf{a}@\mathbf{c}$ . As we see in section 14.1.3, this distributivity is crucial in allowing incremental construction of the thematic tree.

In section 13.2.1, Kanerva's specification of such system [Kan95], which relied on binary vectors, was presented. This system has the drawback that the bind and unbind operators are the same as each other. Thus, it is not possible to represent  $\mathbf{a}@\mathbf{b}@\mathbf{b}$  because this gives  $\mathbf{a}$ . Plate [Pla95] presented a more complex system based on real-valued vectors, which does not have this drawback. However, it relies on high precision in activation values, which may not be realistic.

As discussed above, the TPARRSE model relies on the general properties given here. Plate and Kanerva's proposals are existence proofs that systems with these general properties exist. The proposal is that the brain uses an encoding with similar properties, although this encoding may not directly correspond to either system.

### 14.1.2 Representation of the Thematic Tree

Next we consider how these RR primitives are used to represent the thematic tree. This section focuses on how information is encoded, not how the representation is created. (The latter topic is discussed in the sections specifying the parsing algorithm.)

We first discuss some basic linguistic definitions, and then specify how these

ideas are implemented in the RR encoding. In analyzing language, verbs are often thought of as functions, or *predicates*, which take *arguments*. For example, the verb *loves* is a function that takes Agent and Theme arguments, and specifies a relationship between those two entities. The number and type of these argument categories are determined by the predicate. *Love* takes two arguments, while a verb like *sleep* only takes one (the Agent). Other parts of speech are also predicates and can impose similar restrictions. For example the adverb *because* requires an entire clause or proposition for syntactic and semantic completeness:

31. \*Because John, ...

Because John is sick, ...

A category whose occurrence is not restricted by any semantic feature of another category, but rather can co-occur with any member of a certain part-of-speech class, is called a modifier or adjunct. For example, the time modifier *on Tuesday* may appear with any verb.

32. I slept on Tuesday.

I loved the movie on Tuesday.

Recall that the bind operator creates a new item which is not similar to its constituent items. Therefore, it is used to represent the predicate-argument relationship. Because the resultant vector is unlike the constituent vectors, the bind operation encapsulates the constituent vectors, allowing a hierarchical encoding of argument structure. In contrast, the merge operator is used to join together items within a clause, such as a verb's arguments, or an argument and an adjunct.

For example, the RR encoding of:

33. Sue kissed Bill on Saturday.

is:

**sue + kissed@bill + Vmod@on@saturday**

This encodes that *sue* is the subject of *kissed*, and *Bill* is the object of *kissed*<sup>1</sup>. *Saturday* is the object of *on*, and the prepositional phrase (PP) *on Saturday* modifies the verb *kissed*. The verb's arguments and adjunct (the PP) are joined together by the merge operator.

Verbs and prepositions are bound directly to their objects, as the presence of these categories is predicted by the semantic properties of that verb or preposition. The subject of the sentence remains unbound. Other semantically determined relationships are represented by special predefined items (identified with capital letters). For example, a PP which is a verb adjunct is bound to the predefined item **Vmod**.

As we discuss in section 15.1, in order to decode an RR encoding, it is necessary to know the identities of the predicates. Therefore, each predicate is also bound to a predefined “hook” **P**, which can be used to retrieve the predicate. Thus the encoding of (33) is actually:

**sue + kissed@(P + bill) + Vmod@(P + on@(P + saturday))**

For brevity, we will continue to use the notation **predicate @ argument** to mean **predicate @ (P + argument)**. If a verb does not have an object, it is only bound to **P**. In this case, **P** will be given explicitly.

In a passive sentence, the subject is the Theme and not the Agent, as in the sentence:

---

<sup>1</sup>The representation of **kissed** would actually be **Past + kiss @ (...)** . However, for brevity, I will treat verbs as unitary items.

34. Bill was kissed by Sue.

In this case, the verb is bound to a trace of the subject, denoted **Subj**, and the Agent is expressed as a predicate giving:

**bill + kissed@Subj + Agent@sue**

This encodes the thematic roles of the verb's arguments, while retaining the information that the subject *Bill* is the focus of the sentence.

A ditransitive verb, such as *gave*, requires an additional thematic role, the Goal. For example, in the sentence:

35. John gave Bill the dog.

*Bill* is the Goal. The RR encoding of the sentence is:

**john + gave@(the + dog) + Goal@bill**

If an argument is itself a clause, the same encoding rules recursively apply. For example, in:

36. Mary said that the man arrived from the beach.

the Theme of the verb *said* is the sentential complement *the man arrived from the beach*. The RR encoding is:

**john + said@(the + man + arrived@P + Vmod@from@(the + beach))**

Here **said** is bound to the encoding of its sentential complement.<sup>2</sup>

---

<sup>2</sup>We only consider sentential clauses whose complementizer does not add additional semantics. That is, we don't consider sentences involving wh-movement, such as *John knows when Mary came*. Such sentences would have to be handled differently, in order to include the complementizer.

A traditional syntactic tree uses geometry to encode semantic dependencies. For example, all phrases which constitute a clause *C* lie below the clausal node representing *C*. In the RR encoding, we delineate such relationships by using the term *enclosing scope*. This refers to the items to which an item **x** is bound. The enclosing scope determines where **x** is attached in the thematic tree.

For example, in the sentential complement above, the phrases **the + man, arrived@P**, and **Vmod@from@(the + beach)** all have the enclosing scope **said**. This gives an implicit representation of co-constituency. Because **Vmod@from@(the + beach)** has the same enclosing scope as the verb **arrived**, it modifies that verb. The enclosing scope of the PP **from@(the + beach)** is **Vmod**, indicating that it modifies a verb. If this PP were not bound to **Vmod**, it would be associated with the subject of the complement clause instead, yielding a reading equivalent to:

37. John said that the man from the beach arrived.

Because *bind* distributes over *merge*, the RR encoding of the above example is equivalent to:

**john + said @(the + man) + said@arrived@P +  
said@Vmod@(the + beach)**

Thus, an RR encoding is comprised of NPs having various enclosing scopes; each NP's enclosing scope specifies its role. For example, the enclosing scope of **the + beach** is **said @Vmod@from**, indicating that this NP is the object of the preposition **from**, and this PP modifies the verb having the enclosing scope **said**. Therefore, an RR encoding can be incrementally constructed by maintaining the enclosing scopes that apply to each NP or clause, as we see next.

### 14.1.3 Generating the RR encoding

We now turn to how we generate the RR encoding incrementally during parsing. For now, we address unambiguous, right-branching sentences. A right-branching clause begins to the right (i.e., at the end) of the parent clause. These structures are easy to process because there are no incomplete dependencies in the parent clause when the embedded clause is introduced. Therefore, the RR encoding of such sentences can be produced directly from the input, without relying on the temporal WM representation.

#### Two Stages of RR encoding

Within the RR portion of the system, there are two stages. The first stage groups together words into phrases. When a new word signals the conclusion of a category, its RR encoding and its syntactic type are passed to the second stage. Thus the types of items received at the second stage are: noun phrases, adjective phrases, verbals (verb plus auxiliaries), adverbs, prepositional phrases modifying verbs, complementizers (which introduce complete embedded clauses), relative pronouns (which introduce relative clauses), and conjunctions. The second stage uses the syntactic information to incrementally attach the first-stage pieces into higher level structure, yielding clauses.

The assumption of multiple stages of RR encoding is driven by several factors. Processing of phrases qualitatively differs from processing of clauses in that phrases can be parsed via a finite-state machine, while clauses cannot. That is, phrases cannot be center-embedded within one another. For example, consider the adjective phrase *pretty in pink* and the noun phrase *the girl*. It is not possible to say *the pretty in pink girl*. It is computationally more efficient to process

phrases differently than clauses because phrases can be processed via a more restricted mechanism [Abn89]. The segmentation of sentences into phrases is also supported by prosodic patterns [Abn95].

Furthermore, this two-stage approach is reminiscent of the dynamic programming schemes proposed in symbolic, context-free processing systems [She76]. These systems decouple the processing of the internal details of a phrase from the determination of the hierarchical position of the phrase in the tree. Intuitively, this captures the insight that a category is likely to have the same internal structure irrespective of where it is attached in the tree. If reanalysis is required, we save processing resources if reprocessing the internal details of the category is not part of restructuring the category as a whole within a tree [She76, Lew95]. This is another reason that is computationally more efficient to process phrases separately from clauses.

The distinction between phrasal processing and clausal processing is borne out via reanalysis phenomena. When processing a noun phrase, its internal structure can easily be revised as more information becomes available. For example, the interpretation of *the brown dog* is readily changed if the word *house* follows. However, once processing of a noun phrase is complete, is difficult to restructure that phrase. Consider [Mar80]:

38. The cotton clothing is made from grows in Mississippi.

Once *the cotton clothing* has been detected as a noun phrase, it is difficult to reinterpret it as the NP *the cotton* followed by a relative clause starting with *clothing*. In this unusual case, the detection of a phrase boundary fails, yielding a difficult reanalysis. These phenomena are consistent with a parsing strategy that uses a greedy algorithm to process phrases (i.e., if the next word can be incorporated

into the current phrase, do so), followed by a separate process that composes phrases into clauses [Abn91]. In the remainder of this article, we concentrate on how these phrasal building blocks are combined in the clausal stage of analysis.<sup>3</sup>

## RR Processing Units

The RR encoding of a sentence is generated clause by clause, in order to minimize the amount of information stored in WM. In addition to the temporal representation of syntactic structure, WM contains “variables” that encode other information needed for the parsing process. Each variable corresponds to a neural area dedicated to representing specific information.

The variable **CurRR** holds the RR encoding of the clause currently being processed, **CurSc** holds the enclosing scope within the current clause, and **TotSc** holds the enclosing scope for the current clause as a whole. **TotRR** holds the RR encoding of the entire sentence.

A sentence is processed as follows. A verb, thematic role, or **Vmod** is stored in **CurSc**. Each NP or PP is bound to **CurSc** and the result is merged with **CurRR**, forming the encoding of the current clause. When the current clause is complete, it is *chunked*: **CurRR** is bound to **TotSc**, and the result is merged with **TotRR**; if appropriate, **CurSc** and a clausal predicate are incorporated into **TotSc**. At the conclusion of the sentence, **TotRR** holds the encoding of the

---

<sup>3</sup>This is not to say that there is no interaction between the two stages. For example, in processing the sentence *John gave her earrings*, the first stage could be “aware” that consecutive arguments are expected after *gave*. Therefore, it is more efficient to interpret *her earrings* as two NPs (where *her* is an accusative pronoun) than as a single NP (where *her* is a possessive pronoun), because the former possibility completes the argument structure of the verb. Thus it is assumed that clausal information could influence processing in the first stage.



entire thematic tree. This algorithm is specified in more detail in Figure 14.1.<sup>4</sup>

## An Example

Table 14.1 presents the processing of the following sentence:

39. Sue likes the vase that Joe bought.

At the conclusion of the sentence :

$$\mathbf{CurRR} = \mathbf{sue} + \mathbf{likes@}(\mathbf{the} + \mathbf{vase}) + \mathbf{likes@C@}(\mathbf{joe} + \mathbf{bought@}(\mathbf{the} + \mathbf{vase}))$$

**C** is a predefined predicate applied to an embedded clause which is not a verbal or adverbial argument, such as a relative clause. Because bind distributes over merge, this is equivalent to:

$$\mathbf{sue} + \mathbf{likes@}(\mathbf{the} + \mathbf{vase} + \mathbf{C@}(\mathbf{joe} + \mathbf{bought@}(\mathbf{the} + \mathbf{vase})))$$

Thus, placing the encoding of the RC within the enclosing scope **likes** attaches it to the NP which is also in that enclosing scope, namely **the + vase**. Therefore, upon encountering the RC following *the vase*, it is not necessary to alter the existing RR encoding of *the vase* in order to convert it into an NP modified by an RC. Rather, merging of new information is all that is required. Thus the specific form and properties of the RR encoding allow incremental construction of the thematic tree.

Once categories are bound to each other and RR encoded, they are not directly decomposable, but operations can apply to the vector as a whole. For example, **CurRR** can merge as a unit to **TotRR**.

---

<sup>4</sup>This algorithm assumes that the verb is in the active voice. The passive voice will be addressed in future work on reanalysis.

<b>x</b>	<b>CurSc</b>	<b>CurRR</b>	<b>TotSc</b>
sue		sue	
likes	likes	sue	
the+vase	likes	sue + likes @ (the + vase)	
that			likes@C
joe		joe	likes@C
bought	bought	joe	likes@C
the+vase	bought	joe + bought @ (the +vase)	likes@C

Table 14.1: WM variables after each item **x** is processed from sentence 39. The relative pronoun *that* introduces the predicate **C** and starts a new clause, giving **TotRR** = sue + likes@(the + vase). It also causes its referent, **the + vase**, to be stored, so that it can be accessed when a gap is encountered. During processing of the relative clause, the parser determines that the object of *bought* is a gap, corresponding to the referent of the relative pronoun. At the end of the sentence, chunking is invoked, yielding the final value of **TotRR** given in the text.

In addition to the processing outlined here, there must also be integration of grammatical and semantic features. For instance, the subject must match the verb's features, such as number, person, and animacy. The details of these integrations are beyond the scope of the article. However, we do assume that such information must be available, and that the RR encoding of a phrase allows access to these features.

It is not always the case that a clause following a verb is part of its argument. Consider:

40. Mary kissed Bill when Joe won.

Here a main clause is followed by an adverbial clause. The enclosing scope of the adverbial should indicate that the attachment point is outside of the verb phrase.

<sup>5</sup> The desired RR encoding is:

**mary + kissed@bill + when@(joe + won@P)**

In this case, **kissed** should not be transferred from **CurSc** to **TotSc** after *when* is encountered. Thus, when a new clause is initiated with an adverb, **CurSc** is erased without being incorporated into **TotSc**.

## 14.2 Temporal Working Memory

As discussed in section 11.2, stack-like functionality is necessary for parsing center-embedded clauses. I propose that a serial list provides this functionality. The serial list is based on the same principles as the serial representation of letter

---

<sup>5</sup>Syntactic tests into possible co-referents for pronouns in the adverbial clause show us that the adverb should be attached outside of the verb phrase in what linguists refer to as an adjunction structure.

```

/* Initialize*/
set WM variables to empty

/* process input */
for each item x
  if (current clause is complete) /* chunk */

    /* integrate current clause */
    TotRR = TotRR + TotSc @ CurRR
    CurRR = empty

    /* integrate current scope */
    (if x is not an adverb) TotSc = TotSc @ CurSc
    CurSc = empty
  end if

  if start of new clause /* branch */

    /* integrate new scope */
    if (x is a relative pronoun) TotSc = TotSc @ C
    else (if x is an adverb) TotSc = TotSc @ x
  end if

  /* integrate x itself */
  if (x is a verb) CurSc = x
  else if (x is a PP) CurSc = Vmod

  if (x is an NP or PP) CurRR = CurRR + CurSc @ x

end for

```

Figure 14.1: Basic algorithm for generating the RR encoding of a sentence having only right-branching clauses.

order used in the SERIOL model. In the TPARRSE model, two lists are used, and the relative timing of firing across lists encodes syntactic information. Such a representation could be used like a stack, or could be used to parse crossed-serial dependencies. I first present the basic operations on a serial list, and then discuss how syntactic information is represented. Then we will be in a position to see how temporal WM is used to parse center-embedded clauses.

### 14.2.1 Primitives

The neural substrate of a temporal list is the same as in the SERIOL model, described in section 6.2.2. For ease of presentation, I will first consider list items that are comprised of a single node. Then vector list items will be addressed.

Nodes that represent list items oscillate in synchrony and send lateral inhibition to each other [Lis95]. Timing of firing of an oscillatory node is driven by input level. A high input level allows a node to fire near the trough of the cycle (where excitability is low). Lower input levels push firing later into the cycle, because firing is delayed until excitability increases enough to cross threshold. It has been proposed that an after-depolarization (ADP) can maintain short-term memory across oscillatory cycles in the absence of external input [Lis95]. The ADP is a slow, steady increase in excitability observed in cortical cells following spiking, peaking at approximately 200 ms post-spike [And91]. The temporal gradient of the ADP can maintain the firing order of elements across oscillatory cycles, as demonstrated in a simulation [Lis95]. For example, consider nodes A, B, C firing in sequence during one oscillatory cycle. During the next cycle, node A will have the highest ADP (because its ADP has been increasing for the longest period of time), and node C will have the lowest ADP (because it fired most recently). Therefore, node A will cross firing threshold first, then node B,

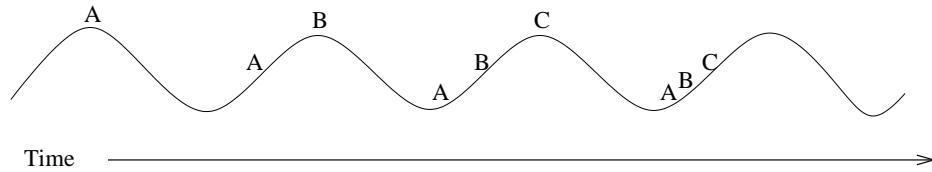


Figure 14.2: Illustration of timing of firing of list elements A, B, and C. Each new element is activated at the peak of the oscillatory cycle. Previously activated items move forward with respect to the cycle, due to the ADP. Over time, A, B, and C come to fire successively within a single cycle.

then node C. Thus, the firing order is preserved across cycles, providing a working memory. The lateral inhibition between nodes is required to maintain this sequential firing. If this lateral inhibition is removed, nodes A, B, and C will eventually start to fire at the same time as each other.

How then is the initial firing pattern established? As long as a node is first activated after all active nodes have already fired (near the peak of the oscillatory cycle), the correct firing order will be maintained. In successive cycles, each newly activated item will fire earlier within the cycle (as a result of the ADP), until it can fire no earlier, due to lateral inhibition from previous node, or to reaching the trough of the cycle (for the node activated first). See Figure 14.2. Thus, one basic operation is the Append operation, which adds a new item to a list by activating the corresponding node(s) during the peak of the oscillatory cycle. The new item then comes to fire after all of the previously active items.

Read out from a list occurs implicitly during every oscillatory cycle, as each item fires. This firing could be used to drive other computations. It is assumed that any such computations are always activated during the trough of the oscillatory cycle, so as to recover the full list.

The proposed list items are not single units, but rather large, binary vec-

tors. I propose that a bank of cells exists for each vector position. A vector is represented by synchronous firing across these banks. All vectors in a list are represented by these same banks of cells. That is, a subset of cells in each bank is activated for each vector. Thus, for a 1 (or 0) in the same position in two different vectors, different subsets of cells from the same bank are active on different temporal subcycles. Each bank of cells consists of two populations, one population representing 1, and the other population representing 0. I assume that each subset is activated stochastically from the population of cells; it is possible for a cell that is already representing an item  $\mathbf{x}$  to be recruited to represent a different item  $\mathbf{y}$ . However, it is unlikely that all cells representing  $\mathbf{x}$  will be reassigned. Thus, the activation of a new item can reduce the activation level of previous items somewhat.

The two populations of cells within a bank reciprocally inhibit each other through fast connections, because the two possibilities (0 or 1) are mutually exclusive within a single sub-cycle (i.e. within a vector position). Cells across banks (positions) send fast but weak excitation to each other, to promote nearly synchronous firing within a subcycle, preventing firing drift across vector positions. Cells also inhibit one another across banks via slower inhibitory connections. These slower inhibitory connections create the sub-cycles and maintain sequential firing. Thus the fast excitatory and inhibitory connections serve to coordinate firing within a vector (sub-cycle), while the slower inhibitory connections serve to keep different vectors separate from one another (across sub-cycles). See Figure 14.3. Indeed, Abbott [Abb91] has demonstrated that a network with fast excitation, and fast, as well as slow, inhibition allows convergence to a series of attractor states. The fast connections allow convergence to a stored pattern, while the slow inhibition deactivates that pattern, allowing formation of a new pattern. In the

Abbott model, the patterns are determined by connection weights. In the present model, the patterns (vectors) are determined by ADP level, and the slow inhibition serves primarily to separate patterns, rather than to directly deactivate patterns.<sup>6</sup>

Thus oscillatory cells with the above inter-connectivity allow a repeating encoding of a sequence of vector items. It would be possible to have another, separate set of oscillatory cells with the same inter-connectivity (within that set). Thus separate temporal lists could be maintained. If the cells oscillate in synchrony across lists, this would allow synchronization of firing across lists. For example, the first item activated on list A (denoted  $A_1$ ) would come to fire at the trough of the oscillatory cycle, as would the first item activated on list B ( $B_1$ ). Thus  $A_1$  and  $B_1$  could be initially activated at different times, but would come to fire in synchrony.

The other important memory function is the Delete operation. An list item is deleted via inhibition. If all active items are to be deleted, a general inhibitory signal could be broadcast. However, if a subset of the active items is to be deleted, the specific items to be inhibited must be identified. How might this occur? One possibility is deletion is based on serial position. That is, it could be recorded that deletion should start at the  $n$ th item on a list. However, this would require the addition of a counting mechanism. Furthermore, such a mechanism is suspect,

---

<sup>6</sup>The specification of separate populations representing 0 and 1 is contingent on the particular RR encoding used in the model (based on binary vectors). However, the general scheme of an array of cell banks is applicable to any type of RR encoding, although it would be more difficult to maintain a WM memory representation if the activation level within a position mattered. In the current scheme, a cell is either active or not; the positional value is determined by which cells are firing (0 or 1). If the positional value depended on activation level, then both activation level and timing of firing would have to be preserved in the WM representation.



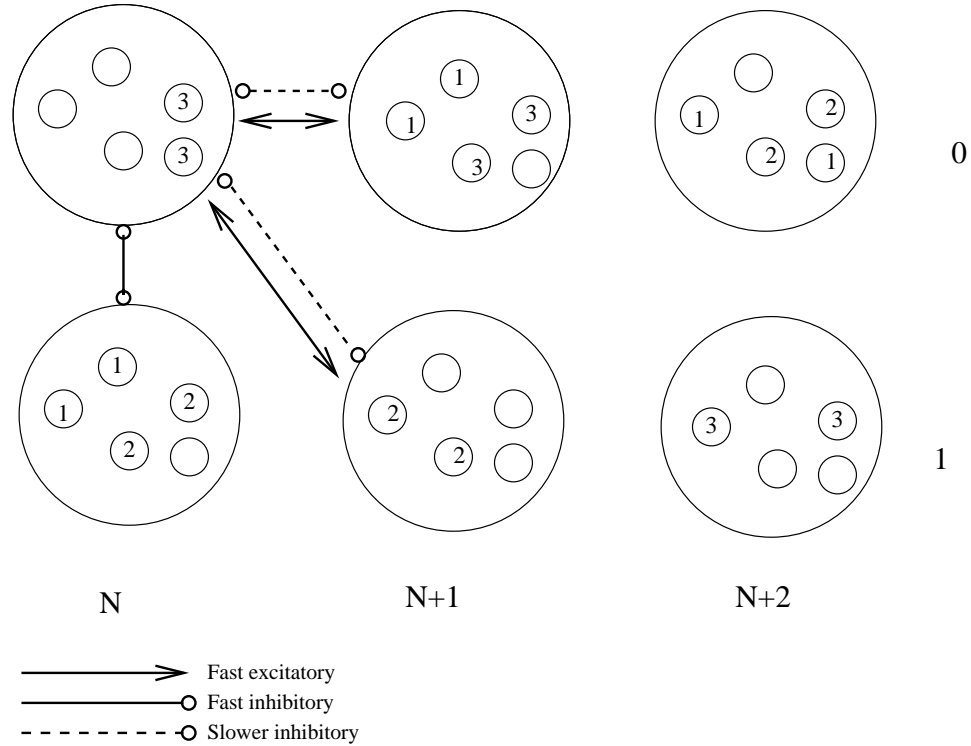


Figure 14.3: Proposed architecture for a WM list, illustrated for positions N to N+2. In this example, 100, 110, and 001 are encoded across those positions on successive oscillatory subcycles. Each large circle represents a bank of nodes coding for the same value and position. A subset of those nodes is shown by the small circles. Each column represents a vector position. The top row encodes 0's, while the bottom row encodes 1's. The number in each node reflects the oscillatory subcycle in which it fires. Fast connections coordinate firing within a sub-cycle, while slower inhibitory connections separate subcycles.

as linguists have observed that natural language seems to crucially eschew use of counting predicates [Cho80, Ber84]. Rather, natural language operates under structural constraints. Therefore, it is assumed that partial deletion from a list is based on the *structural identity* of a list item. That is, when a list item that will later require a partial deletion is first activated, the syntactic features of that item are stored in a WM variable. When partial deletion is required, inhibition is triggered when the currently firing list item matches the stored value. Comparison between activation patterns is generally taken to be a fundamental function of neural networks. Thus identification based on identity does not require any additional mechanisms. Further details of this deletion process will be discussed in the following section.

## 14.2.2 Representation of Syntactic Information

We have seen how separate, but synchronized, temporal lists could be neurally instantiated in WM. I propose that working memory uses such lists to store syntactic information about all incomplete clauses. This parallel representation of the sentence serves two purposes. (1) When the RR encoding cannot be directly generated from the input (in the case of center-embedding or crossed-serial dependencies), it allows processing of interrupted clauses to be re-instantiated. (2) When a parsing error occurs because the wrong choice was made at a point of ambiguity, the temporal encoding allows an alternative RR encoding to be generated (in some cases). The present work focuses on the first possibility. The second will be addressed in future work.

I propose that syntactic structure is encoded by employing two separate lists - one for noun phrases, and one for predicates and verb adjuncts. The lists are synchronized so that items which are in the same position in different lists fire

at the same time. The relationship between an NP and a verb is encoded by their relative firing times. Subjects and adjuncts fire asynchronously with their verbs (with subjects firing prior to their verbs in English), while objects fire synchronously with their verbs. For example, the WM representation of (41) is:

<b>sue</b>	<b>bill</b>	
<b>E</b>	<b>called</b>	<b>on@Monday</b>

41. Sue called Bill on Monday.

The top row is the noun list, and the bottom row is the verb list. Each column represents a temporal slot. That is, the first items in each list fire together, then the next items, and so on. The filler item **E** occupies the first slot on the verb list in order to establish the proper firing relationships. A PP which modifies a verb is recorded on the predicate list.

Each item on the lists is accompanied by a *tag field*, which records syntactic features. Recall that the first stage of RR processing returns syntactic information along with the encoding of the phrase. The syntactic information specifies the type of the phrase; this type determines which list the phrase is Appended to, and is used to update the internal state of the parser. When the lists are used to generate an RR encoding, it is necessary to access this syntactic information so that the parser can return to the correct internal state. A tag-field item is not a vector, but rather is a unitary feature. For example, the encoding of 41 with the tag fields is:

NP	NP	
<b>sue</b>	<b>bill</b>	
<b>E</b>	<b>called</b>	<b>on@Monday</b>
E	V	PP

where the corresponding tag fields are given in the outer rows. *NP*, *V*, etc. represent that nodes representing that feature are active during that temporal slot.

The start of an embedded clause is also marked in the tag field. For example the encoding of (42) up to *fell* is:

E	NP	NP, Cl, GapReq	Gap
<b>E</b>	<b>the + vase</b>	<b>sue</b>	<b>Gap</b>
<b>on@wednesday</b>	<b>E</b>	<b>E</b>	<b>bought</b>
PP	E	E	V

42. On Wednesday, the vase that Sue bought fell.

*Cl* marks **sue** as the start of an embedded clause, and *GapReq* indicates that a gap is required (i.e., it is a relative clause). If the relative clause is subject extracted, the verb is tagged instead. For example, the encoding of (43) is:

E	NP	NP, Cl, Gap
<b>E</b>	<b>the +woman</b>	<b>bill</b>
<b>on@wednesday</b>	<b>E</b>	<b>knows</b>
PP	E	E

43. On Wednesday, the woman who knows Bill ...

This unambiguously and efficiently encodes a relative clause with a gap in the subject position.

The proposed temporal encodings can be generated by sequentially Appending each item as it is received to the proper list (under the assumption that a subject also generates an Append(E, verb list) and a PP generates an Append(E, noun list)). Recall that each item comes to fire as early as possible on its respective

list. Thus the first items on both lists will fire together, then the second items, etc.

As discussed in section 13.2.2, the problem of how to represent a hierarchical structure temporally is a difficult one. There must be a mapping from a two-dimensional structure (a tree) onto one-dimensional structure (time). Given this mapping, there must be a way to initiate the correct firing times on the fly. Our proposal for the structure of WM solves these problems. Because the oscillatory cycle provides a reference frame, two items that are activated at different times on different lists can come to fire synchronously. Because the order of firing is maintained across cycles, information can be encoded by the sequence of firing, not just by the synchrony of firing. Because each list item is an RR encoding, this allows structure to be represented within a single temporal slot. Thus this scheme solves the problem of how to temporally represent arbitrary hierarchical on the fly. This temporal representation maintains the identity of phrasal subcomponents, allowing them to be referenced separately from each other. In the next section, I discuss how this representation is used for parsing of center-embedded clauses.

### 14.3 Processing Center-embedded Clauses

Now, we will see how the addition of the temporal encoding allows us to handle the problem of center-embedding. Recall that the encoding of a clause is constructed in **CurRR**, and then transferred to **TotRR** when a new clause begins. However, when a center-embedded clause is encountered, the RR encoding of the current clause is not integrated into **TotRR**. Rather, **CurRR** and **CurSc** are set to empty, so that they will only encode the embedded clause. When the embedded clause is complete, its temporal representation is deleted from the lists

and is replaced with its RR encoding.

The information on the lists is then used to re-instantiate the RR encoding of the higher clause; this process is denoted *WM-RR encoding*. During WM-RR encoding, the information on the lists is RR-encoded as each pair of list entries fire. Afterward, the parser is in the same state that it was prior to the center-embedded clause, except that the RR encoding of the center-embedded clause is included in **CurRR**.

Next we consider an example. This processing requires an additional WM variable, **CntrSc**, which maintains the enclosing scope of a center-embedded clause. We will see how parsing of (42) proceeds.

- Initially, the WM variables and lists are empty.

After *the vase*, **CurRR** = **Vmod@(on@wednesday) + the + vase**, **CurSc** = **empty**, and the lists are:

E	NP
<b>E</b>	<b>the + vase</b>
<b>on@wednesday</b>	<b>E</b>
PP	E

- At *that*, **CntrSc** is set to **C**, to record the enclosing scope of the upcoming clause. **CurSc** and **CurRR** are set to empty, so that only the upcoming clause will be RR encoded.

After *bought*, **CurRR** = **sue + bought @(the + vase)**, **CurSc** = **bought**, and the lists are:

E	NP	NP, Cl, GapReq	Gap
<b>E</b>	<b>the + vase</b>	<b>sue</b>	<b>Gap</b>
<b>on@wednesday</b>	<b>E</b>	<b>E</b>	<b>bought</b>
PP	E	E	V

- At *fell*, the center-embedded clause is complete. A partial delete of list items is initiated at the start of the embedded clause, and then **CntrSc@CurRR** is Appended to the lists, giving:

E	NP	ChunkedCl
<b>E</b>	<b>the + vase</b>	<b>C@(sue+bought@(the+vase))</b>
<b>on@wednesday</b>	<b>E</b>	<b>E</b>
PP	E	E

**CurRR**, **CurSc** and **CntrSc** are set to empty. The information on the lists is then used to recover the RR encoding of the main clause, giving:

**CurRR** = **Vmod@(on@wednesday) + the + vase + C@(sue + bought@(the + vase))**

Importantly, **the + vase** is encountered as a separate entity, in principle allowing access to grammatical and semantic features necessary for integration with the upcoming verb. (However, details of such integration are beyond the scope of the present work.)

- Now *fell* can be processed as usual (i.e., directly incorporated into the current clause).

Thus the lists work like a stack to maintain unattached subjects. A center-embedded clause overwrites processing of the higher clause; the RR encoding of

the higher clause is later re-generated from the temporal WM encoding. Next we consider in more detail the deletion of the center-embedded from the lists.

## 14.4 Partial Deletion

Recall that partial deletion is performed by storing the identity of the target item, and then matching to this stored value. Given that rules in natural language operate on syntactic structures, I assume that an item is identified by its syntactic features. When a center-embedded clause is encountered, the corresponding syntactic features are recorded in a WM variable denoted *Dtag*. In our example, *Dtag* would be set to  $(Np, Cl, GapReq)$ . When the embedded clause is complete, inhibition is initiated at the item having these syntactic features. That item, and all successive items on both lists are inhibited, thereby deleting the temporal representation of the relative clause from working memory.

Next we consider lower-level details of how the inhibition is triggered. I propose that matching works on the principal of dis-inhibition. That is, *Dtag* features inhibit the inhibition of list items, while tag-field features inhibit *Dtag* features. When deletion is required, its initiation is inhibited by the activity of *Dtag* features. When the corresponding features are active in the tag field, they inhibit the *Dtag* features. Therefore, inhibition is no longer inhibited by *Dtag*, and is triggered. Inhibition then continues for the remainder of the oscillatory cycle. If no features are initially active in *Dtag*, a full deletion of all list items is automatically carried out. See Figure 14.4 for a schematic of the proposed network. When deletion occurs, only those cells that are currently firing should be inhibited. This is accomplished via a gating mechanism that allows the inhibitory signal to reach a cell only if that cell is active. (See Figure 14.4.)



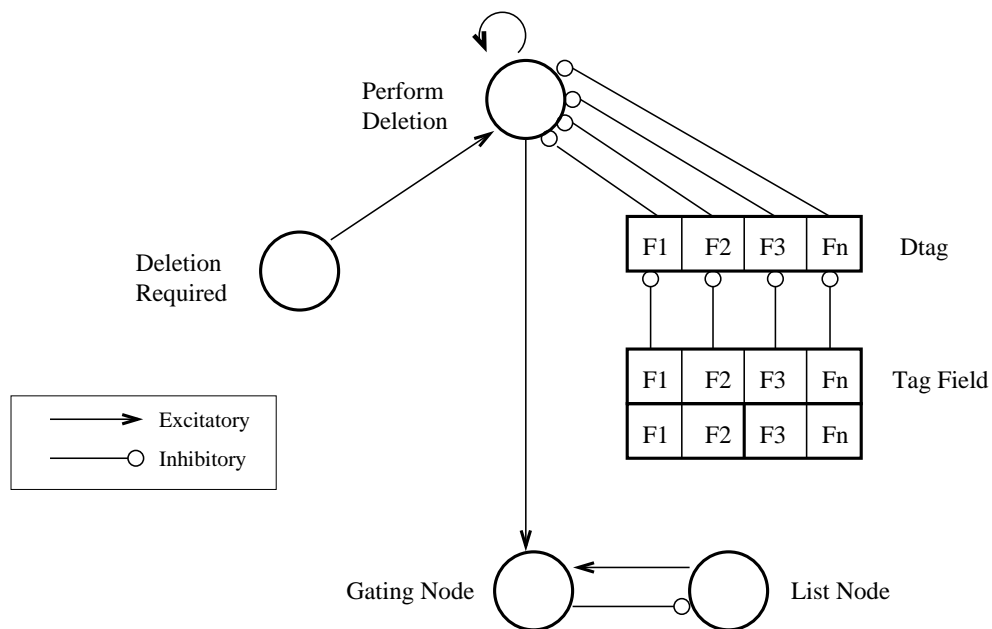


Figure 14.4: Proposed architecture of deletion network. The tag field is comprised of syntactic features F1, F2, F3 ... Fn, with multiple instances of each feature (two instances shown here). Each feature has inhibitory connections to the corresponding feature in Dtag, and each feature in Dtag inhibits the node which drives the deletion process. When the tag-field features inhibit all of the Dtag features, the perform-deletion node is activated and deletion is initiated. Deletion is sustained via the self-excitatory connection. The gating node becomes activated only if it receives excitation from both the perform-deletion node and the list node. In that case, the list node is inhibited. Thus inhibition only applies to active list nodes, and does not affect list nodes that fired prior to the initiation of deletion. (Only a single list node is shown. A similar circuit is required for each list node.)

Note that the proposed matching mechanism does not require an exact match between the tag field and *Dtag*. There is an asymmetry. For deletion to be initiated, all of the active *Dtag* features must also be active in the tag field, but all tag-field features do not have to be active in *Dtag*. Thus, a match occurs when the tag-field features are a superset of the *Dtag* features, but not when the *Dtag* features are a superset of the tag-field features. As we will see in Chapter 16, this asymmetry is crucial in explaining complexity phenomena.

## 14.5 Arbitrary Hierarchical Structure

Thus far we have seen how a single right-branching or center-embedded clause is handled. However, arbitrary combinations of branching patterns can occur in natural language. The parsing algorithm can easily be extended to handle arbitrary hierarchical structure. Aside from right-branching and center-branching structures, a clause could also be preposed or left-branching. A preposed clause occurs when an adverbial is moved to the front of the higher clause, as in:

44. When the vase fell, Mary was upset.

A left-branching clause is a sentential subject, as in:

45. That the vase fell upset Mary.

Preposed and left-branching clauses are processed the same way as center-embedded clauses. If a right-branching clause occurs within a non-right-branching clause, its predicate is stored in **CntrSc**, so that it can be deleted when the higher clause is complete. Thus **CntrSc** holds any clausal predicates that must be deleted at some point, while **TotSc** holds purely right-branching predicates.

If a predicate is assigned to **CntrSc**, it is also recorded on the lists, so that it can be recovered if necessary. That is, if a non-right-branching clause *A* is being

processed, and another embedded clause  $B$  is encountered,  $B$  will overwrite the information in **CntrSc** pertaining to  $A$ . Once the processing of  $B$  is complete, the information pertaining to  $A$  can be re-instantiated during WM-RR encoding. This allows recursion, and therefore arbitrary branching patterns can be processed. In contrast, predicates that are assigned to **TotSc** are not maintained on the lists, because they cannot be overwritten, and should never be re-processed. For example, the  $Cl$  syntactic feature (which corresponds to the **C** predicate) is activated in the tag field of a center-embedded RC, but is not activated for a purely right-branching RC (because **C** is stored in **TotSc** in this case). In section 15.3, the full parsing algorithm and simulations of the algorithm are presented.

## Chapter 15

### Computational Demonstrations

In this chapter, I present implemented demonstrations of some aspects of the above theoretical model. In the first section, I present the decoding of an RR encoding of a sentence. In the second section, I show that the proposed single-node dynamics of a WM allow a serial encoding of items activated during different oscillatory cycles. In the third section, a simulation of the full parsing algorithm is presented.

#### 15.1 Decoding an RR encoding

In the following, I use Kanerva's [Kan95] scheme to demonstrate that the information recorded in the RR encoding of a sentence can indeed be extracted. Recall that the unmerge operation is performed by comparing to item memory, and that all predicates in the RR encoding are bound to  $\mathbf{P}$  to allow them to be recovered. Each unbind is followed by a comparison to item memory, to clean up the result of the unbind. Decoding proceeds as follows. First the vector is compared to item memory. This retrieves the unbound constituent, i.e., the Agent. Then the vector is unbound with  $\mathbf{P}$  to retrieve any predicates. The vector is unbound with each of those predicates to retrieve their arguments.

We first illustrate decoding at the conceptual level, then we present a numerical example. Consider the vector:

$$\mathbf{Q} = \mathbf{ann} + \mathbf{loves}@\mathbf{(P + joe)}$$

where **ann**, **loves**, and **joe** are stored in item memory. Comparing **Q** to item memory yields the Agent **ann**, because it is the only constituent vector that is unbound. The vector **Q** is then unbound with **P**, yielding:

$$\mathbf{Q}\#\mathbf{P} = \mathbf{ann}\#\mathbf{P} + \mathbf{loves}@\mathbf{P}\#\mathbf{P} + \mathbf{loves}@\mathbf{joe}\#\mathbf{P}$$

which is similar to **loves**. It is also similar to **ann#P** and to **loves@joe#P**, but these vectors are not stored in memory and act as noise. Thus, comparing **Q#P** to memory yields **loves**. The vector **Q** can then be unbound with **loves** to yield **joe**.

If too many relationships were encoded within a single vector, so much noise could be introduced that it would not be possible to retrieve the base vectors. This can be remedied by storing intermediate results. The result of an unbinding which is not a base vector can then be cleaned up by retrieving the similar item from memory. Therefore, we assume that if an argument is itself a clause, its encoding has also been recorded.

Next I present a numerical example of the encoding and decoding of a sentence containing an embedded clause. The sentence:

46. John told Mary that Bill gave Sue money.

was encoded, where **john**, **mary**, **bill**, **sue**, **told**, **gave**, **Goal**, and **money** were vectors stored in item memory. 3000 distractor vectors were also stored in item memory. Vectors were of dimension 10,000, and were randomly generated under

the constraint that each vector position had equal probability of being a 0 or a 1. The encoding of the sentential complement *Bill gave Sue money* was also recorded in memory:

$$\mathbf{V1} = \mathbf{bill} + \mathbf{Goal@}(\mathbf{sue} + \mathbf{P}) + \mathbf{gave@}(\mathbf{money} + \mathbf{P})$$

The encoding of the entire sentence was:

$$\mathbf{V} = \mathbf{john} + \mathbf{Goal@}(\mathbf{mary} + \mathbf{P}) + \mathbf{told@}(\mathbf{V1} + \mathbf{P})$$

$\mathbf{V}$  was then decoded as follows. (All items exceeding a similarity cutoff of 0.52 are presented, with the similarity value in parentheses.)  $\mathbf{V}$  was compared to item memory to get the subject. This yielded **john** (0.75; i.e., 75% of  $\mathbf{V}$ 's positional values matched **john**'s).  $\mathbf{V}$  was unbound with  $\mathbf{P}$ , and the result was compared to memory to get any predicates. This yielded **told** (0.63) and **Goal** (0.62).  $\mathbf{V}$  was unbound with **told** to get the Theme, yielding  $\mathbf{V1}$  (0.62) and **bill** (0.57). This similarity to **bill** is appropriate since  $\mathbf{V}$  contains **told@bill** (because **bill** is the subject of  $\mathbf{V1}$ )<sup>1</sup>.  $\mathbf{V}$  was unbound with **Goal**, yielding **mary** (0.62). This process was repeated with  $\mathbf{V1}$ . The subject was **bill** (0.75). Unbinding  $\mathbf{V1}$  with  $\mathbf{P}$  yielded **gave** (0.63) and **Goal** (0.62). Unbinding  $\mathbf{V1}$  with **gave** yielded **money** (0.63) and unbinding with **Goal** yielded **sue** (0.63). Thus, it was possible to retrieve the information stored in the RR encoding of the sentence.

The goal of this implementation was to demonstrate the feasibility of RR encoding and decoding. Thus the operations were performed algorithmically (

---

<sup>1</sup>Thus the Theme and the subject of a sentential complement are represented in the same way; both are bound to the verb. This similarity in representation may be related to the ease with which a Theme can be reanalyzed as the subject of a sentential complement in ambiguous sentences lacking *that*, such as *John knows Bill likes Sue*.

i.e., not implemented as a neural network). However, the proposed operations are neurobiologically plausible. Because the merge, bind, and unbind functions operate on corresponding bits across two vectors, they could be implemented within a neural network using one-to-one connections between the areas over which the input vectors are represented. The unmerge operation requires an auto-associative memory; recurrent networks employing distributed representations have long been touted as a natural framework for this kind of memory.

## 15.2 Temporal WM

The goal of the following simulation is to demonstrate the basic functionality of a WM list, which allows a serial representation of the order of items activated at different times. This is a network simulation, with each list item being represented by a single node. A list node is always initially activated during the peak of the oscillatory cycle. Thus a mechanism that coordinates timing of activation with the oscillatory cycle is assumed. However, this mechanism is beyond the scope of the present work. Thus, it was not implemented as part of the network.

Following [Lis95], list nodes are modeled as units that undergo a sub-threshold oscillatory drive, exhibit an increase in excitability after firing (ADP), and send lateral inhibitory inputs to each other. We use  $i$  to denote the  $i$ th node to be Appended. The membrane potential,  $V$ , of a node is given by:

$$V(i, t) = O(t) + A(i, t) - I(t) + E(i, t)$$

where  $O$  denotes the oscillatory drive,  $A$  denotes ADP,  $I$  denotes inhibitory input, and  $E$  denotes excitatory external input. A node fires when  $V$  exceeds a threshold,  $TH$ .  $TH$  is specified relative to resting potential, and set to 10mV. Firing causes the node's ADP component to be reset, and inhibition to be sent to the other

nodes.

The oscillatory function  $O$  has a cycle length of 200ms, and linearly increases from -5mV to 5mV during the first half of the cycle, and decreases back to -5mV during the second half.

The ADP and inhibition are modeled by functions of the form:

$$F(t; M, T) = M * (t/T)^{1.5} * \exp(1 - t/T)$$

which increases to a maximal value (controlled by  $M$ ) and then decreases (on a time scale controlled by  $T$ ). The ADP is given by:

$$A(i, t) = F(t - t_i; M_A, T_A)$$

where  $t_i$  denotes the time at which the  $i$ th node last fired. ( $A(i, t)$  is 0 if the node has not yet fired.) The inhibition is given by:

$$I(i, t) = \sum_{j=1}^n F(t - t_j; M_I, T_I)$$

where  $n$  gives the number of nodes, and  $F$  is 0 if node  $j$  has not yet fired, or if  $i = j$ . The following values were used:  $T_A = 230$ ms,  $M_A = 13$ mV,  $T_I = 5$ ms,  $M_I = 3$ mV.

The external input,  $E$ , is such that node  $i$  receives an input of amount  $M_A$  commencing at time  $F + i * P$ , where  $F$  and  $P$  are constants. Node  $i$  continues to receive this input at each time step until it fires.  $F$  is set to 100ms (the peak of the first cycle), and  $P$  is assumed to be a multiple of 200ms, so that items to be Appended are activated at the peak of a cycle.

A simulation using the above equations, with  $P = 200$  and 8 nodes, yielded all nodes firing in the correct sequence after 9 cycles. Nodes 1-8 fired at times 31, 48, 61, 71, 80, 88, 95, 103, respectively (times are given relative to the start of



the cycle). Thus, when a new item was activated at each successive peak of the oscillatory cycle, all items came to fire sequentially within a cycle, as desired.

If, however, external input is applied out of sync with the oscillatory cycle, incorrect orderings result. For example,  $P = 173$  yielded the following after 9 cycles: nodes 5, 4, 1, 2, 3, 6, 7, and 8 fired at times 30, 47, 60, 70, 70, 85, 85, and 85, respectively, while  $P = 227$  yielded nodes 5, 6, 4, 7, 1, 2, 3, and 8 firing at times 30, 47, 60, 60, 76, 76, 89, and 89.

### 15.3 Parsing Algorithm

In the present work, the aim is to show that the proposed algorithm is viable at the computational level, by demonstrating that the algorithm is powerful enough to handle the complex structures found in natural language. Therefore, the algorithm was implemented at the symbolic level.

#### 15.3.1 Implementation

The full parsing algorithm is given in Figures 15.3.1 and 15.3.1. This algorithm was implemented using a positional variable taking one of four values (before subject, before verb, before object, or after object) to determine the branching direction of an embedded clause. Each input sentence was represented by a sequence of two-character symbols, representing phrases formed by a first stage of RR encoding. The first character was alphabetic, specifying syntactic type, and the second character was numeric, distinguishing different instances of the same syntactic type. For example, the input representation of a sentence with a sentential complement containing a preposed adverbial clause having a right-branching relative clause, like:

47. John said that after Mary dropped the vase that Jim bought, Jane got a new vase.

is:

N1 V1 C1 A1 N2 V2 N3 R1 N6 V3 N4 V4 N5

where 'N' specifies an NP, 'V' a transitive verb, 'C' a complementizer, 'A' an adverb, and 'R' an relative pronoun. The output for each sentence is a string specifying the RR encoding of the sentence. For the above sentence, the desired output is the string:

$$\begin{aligned} &N1 + V1@A1@(N2 + V2@N3) + V1@A1@C@(N6 + V3 @ N3) + \\ &V1@(N4 + V4@N5) \end{aligned}$$

The model was tested on a variety of sentences containing multiple embeddings of relative, sentential, adverbial, and noun-complement clauses. These inputs are given in section 15.3.2. The correct output was generated for all of the sentences, except for RC/RCs and an NC/RC, consistent with human performance. The reason that the algorithm failed on these structures is discussed in the following chapter.

### 15.3.2 Stimuli

The following lists the stimuli used for the parsing simulation. For ease of comprehension, an example sentence is presented for each input sequence. The correct output was generated for all sentences, except 12 and 13 (RC/RCs) and 27 (NC/RC).

### Chunk Current Clause

```
/* Remove current clause from lists */
if (Dtag is empty) Empty lists
else Partial delete starting at Dtag; Dtag = empty

/* Integrate current clause */
if (part of a center-embedded clause)

    Append CntrSc @ CurRR to lists
    if (starting right branch) CntrSc = CntrSc @ CurSc
    else WM-RR encode

else /* right branching only */

    TotRR = TotRR + TotSc @ CurRR
    TotSc = TotSc @ CurSc

end if
```

### Branch on predicate x

```
/* Record start of clause */
if (lists not empty) Dtag will get tag field of clause

/* integrate new scope into clausal scope */
if (starting a center-embedded clause)
    CntrSc = x
elseif (starting a right branch inside a center-embedded clause)
    CntrSc = CntrSc @ x
else /* right branching only */
    TotSc = TotSc @ x
end if

/* reset encoding of current clause */
CurSc, CurRR = empty
```

Figure 15.1: Chunking and branching procedures for the full RR encoding algorithm

```

/* Initialize */
set WM variables and lists to empty

/*Process input */
for each item x

    if (x starts an embedded clause)

        if (x is an adverb) CurSc = empty
        if (no incomplete dependencies) Chunk current clause
        Branch on x

    else if (x resumes a higher clause)

        Chunk current clause

    end if

    if (x is a verb)      CurSc = x
    else if (x is a PP)   CurSc = Vmod

    if (x is a subject NP) CurSc = empty

    if (x is an NP or PP) CurRR = CurRR + CurSc @ x
    if (x is an NP, PP, or verb) Append x to lists

end for

```

Figure 15.2: Full RR encoding algorithm, using Chunk and Branch operations specified in Figure 3.

N1 V1 N2

1. The cat chased the rat.

\*Two clauses\*

N1 V1 C2 N2 V2 N3

2. The man knows that the cat chased the rat.

N1 R1 I2 V3 N3

3. The cat which was chased ate the fish.

N1 R1 N2 V2 V1 N3

4. The cat which the dog chased ate the fish.

N1 R1 V2 N2 V1 N3

5. The cat which chased the rat ate the fish.

N1 V1 N2 R1 V2 N3

6. The cat chased the rat which ate the cheese.

N1 V1 N2 R1 N3 V2

7. The cat chased the rat which the dog bit.

A1 N1 V1 N2 N3 V2 N4

8. After the cat chased the rat, the dog ate the meat.

N1 V1 N2 A1 N3 V2 N4

9. The dog ate the meat after the cat chased the rat.

C1 N1 V1 N2 V2 N3

10. That the dog ate the chocolate bothered Bill.

N4 C1 N1 V1 N2 V2 N3

11. The fact that the dog ate the chocolate bothered Bill.

.

\*Three clauses\*

N1 R1 N2 R2 N3 V3 V2

12. The rat which the cat which the dog hates chased

V1 N4

ate the cheese.

N1 R1 N2 R2 V3 N3 V2

13. The rat which the cat which hates the dog chased

V1 N4

ate the cheese.

N1 R1 V2 N2 R2 N3 V3

14. The dog which chased the cat which the rat feared

V1 N4

ate the meat.

N1 R1 V2 N2 R2 V3 N3

15. The dog which chased the cat which chased the rat

V1 N4

ate the meat.

N1 V1 N2 R1 V2 N3 R2

16. The dog chased the cat which chased the rat which

V3 N4

ate the cheese.

N1 V1 N2 R1 V2 N3

17. The dog chased the cat which chased the rat

R2 N4 V3

which the lion liked.

N1 V1 N2 R1 N3

18. The dog chased the cat which the rat

R2 V4 N4 V3

which ate the cheese feared.

N1 V1 N2 R1 N3

19. The dog chased the cat which the rat

R2 N4 V4 V3

which the lion liked feared.

N1 R1 V2 C1 N3 V3 N4

20. The man who thinks that the cat chased the rat

V1 N5

ate the cheese.

N1 R1 N2 V2 V1 C3 N3

21. The man who the lion chased thinks that the rat

V3 N4

ate the cheese.



N1 V1 N2 R1 V2 C3 N3

22. The lion chased the man who thinks that the rat

V3 N4

ate the cheese.

N1 V1 C2 N2 V2 C3

23. The man knows that the girl thinks that

N3 V3 N4

the cat chased the rat.

N1 V1 C2 N2 V2 N3

24. The man thinks that the rat ate the cheese

A1 N4 V4 N5

after the dog bit the cat.

N1 V1 C1 A0 N2 V2 N3

25. The man knows that when the cat chases the rat,

N4 V4 N5

the lion chases the dog.

N4 C1 N5 R1 N1 V1

26. The fact that the dog which Sue adopted

V2 N2 V3 N3

ate the chocolate bothered Bill.

N4 R1 N5 C1 N1 V1 N5

27. The woman who the fact that the dog ate the chocolate

V2 V3 N3

bothered hit Bill.

\*four clauses\*

N1 R1 V2 C2 N2 V3 N3

28. The man who knows that the cat chased the rat

A1 N4 V4 N5 V1 N6

after the dog ate the meat ate the pie.

## Chapter 16

### Complexity

Next we consider how the proposed processing accounts for complexity phenomena. Recall that we want to explain the following.

- RC/RC
  - Very difficult
  - Effect of NP type
    - \* Easier if N3 is a pronoun.
    - \* N3 pronoun is easier than an N1 or N2 pronoun.
  - V2-drop
    - \* Felicitous for double center-embedding.
    - \* Not felicitous for center-embedded RC within a right-branching RC.
  - N3-type x V2-drop interaction - V2-drop is not felicitous for pronoun N3
- Noun Complements
  - NC/RC easier than RC/RC.

- RC/NC as hard as RC/RC.
- Crossed-serial Dependencies
  - Easier than double center-embeddings.
  - Pronoun N3 has no effect.
- Similarity-based interference
  - Increase in difficulty as number of similar items increases.
  - For a fixed number of similar items, increase in difficulty as their proximity to each other increases.

First we will see how the proposed processing of center-embedded clauses accounts for the RC and NC phenomena. Then the relative ease of crossed-serial dependencies will be addressed, followed by a discussion of similarity-based interference.

## 16.1 Center Embedding

### 16.1.1 RC/RC

Consider processing of the sentence:

48. The vase that the man that Sue dated bought fell.

At the first *that*, center-embedded processing is invoked. At the second *that*, center-embedded processing is again invoked, overwriting the information in the WM variables pertaining to the first RC. When *bought* is encountered, the lists are:

NP	NP, Cl, GapReq	NP, Cl, GapReq	Gap
<b>the + vase</b>	<b>the + man</b>	<b>sue</b>	<b>Gap</b>
<b>E</b>	<b>E</b>	<b>E</b>	<b>dated</b>
E	E	E	V

and *Dtag* is set to the values of the tag field of the inner RC (*NP, Cl, GapReq*). At this point, deletion of the inner RC from temporal WM is required. Recall that explicit read out of a list is always initiated at the trough of the oscillatory cycle. Thus, the “Deletion Required” node is activated at that point. During the first temporal slot, inhibition is prevented because the tag field does not match *Dtag*. However, a match does occur during the second slot, and inhibition is triggered. However, this inhibition is premature; it really should have been initiated at N3. Therefore, N2 is erroneously deleted from the lists, giving:

NP
<b>the + vase</b>
<b>E</b>
E

Thus, because list items are read out in order, and N2’s syntactic features match N3’s, information about the outer RC is erroneously deleted from WM. I propose that this premature deletion is the fundamental cause of the difficulty of an RC/RC.

Note that the the V2-drop effect [Gib99] falls out naturally from this account. During WM-RR encoding following deletion, processing of the *main clause* is re-instantiated, because only the main-clause subject remains on the lists. Thus only the main-clause verb is expected. However, in the case of a center-embedded RC within an *right-branching* RC, such as (49), incorrect deletion does not arise.

49. I like the vase that the man that Sue dated bought.

Following *the vase*, the word *that* signals the start of a right-branching clause. All items on the lists (i.e. the temporal encoding of the main clause) are deleted. In this case, the subject of the outer RC is not tagged with the *Cl* feature, because the **C** predicate is permanently stored in **TotSc**. Therefore the lists up to *bought* are:

NP, GapReq	NP, Cl, GapReq	Gap
<b>the + man</b>	<b>sue</b>	<b>Gap</b>
<b>E</b>	<b>E</b>	<b>dated</b>
E	E	V

and *Dtag* is  $(NP, Cl, GapReq)$ . During deletion of the inner RC, a match does not occur at the first temporal slot because the required *Cl* feature is not active. Therefore the sentence is processed correctly, explaining why V2-drop is not felicitous in this case [Gib99].

An RC/RC could be processed correctly if N2 could be distinguished from N3. I propose that this is why a pronoun N3 makes an RC/RC seem easier [Gib98, War02a]. When N3 is a pronoun, this additional syntactic information is reflected in its tag field. Thus, N3 will have the syntactic features  $(NP, Pr, Cl, GapReq)$ . This tag field will overwrite N2's features in *Dtag*. During deletion of the inner RC, matching is performed against N3's syntactic features. In this case, a match does not occur at N2 because it does not possess the *Pr* feature. Thus deletion of the inner RC proceeds correctly. During WM-RR encoding, center-embedded processing of the outer RC is re-instantiated. The outer RC is processed, and then processing of the main clause is re-instantiated. Thus the sentence is processed correctly. This implies that V2-drop should not be felicitous for a pronoun N3, as we have demonstrated [Whi04d].

However, this analysis only applies when  $N3$  is a pronoun. Recall that a match occurs when all of  $Dtag$ 's features are active in the tag field. If one of the other subjects is a pronoun (while  $N3$  is not) a premature match *will* occur at  $N2$ , because  $N2$  would possess all of  $N3$ 's features. Thus the match asymmetry arising from the disinhibition mechanism explains why an  $N3$  pronoun is easier than an  $N1$  or  $N2$  pronoun [War02a].

### 16.1.2 Noun Complements

A clause that must be a relative clause contains the *GapReq* feature, while a potential noun complement does not. Note that this syntactic feature is independently motivated. Consideration of the following examples demonstrates that the parser needs to keep track of some important differences between these constructions.

50. a. The fact that John read magazines surprised Sue.  
b. The fact that John read in the newspaper surprised Sue.  
c. \*The fact which John read magazines surprised Sue.  
d. The fact which John read in the newspaper surprised Sue  
e. The item that John read in the newspaper surprised Sue.  
f.\* The item that John read magazines surprised Sue.

*Read* is a verb that can be used in either transitive or intransitive constructions. This is why (a) is grammatical as a noun complement, with or without the presence of the direct object *magazines*. This structure has a third reading as a relative clause, as shown in (b). The contrast between (b) and (c) shows that the complementizer *which* can only introduce a relative clause. The fact that

the ungrammaticality of (c) is easily detectable shows that the parser must have some device to indicate that the requirement of a gap to be bound to the head of the relative clause is not satisfied. The easily detected contrast between (a) and (f) shows that the parser must directly mark whether a noun complement is permissible; it is not sufficient to indicate the lexical choice of the complementizer. The *GapReq* feature fulfills these requirements, specifying an additional constraint on an embedded clause. It is necessary to store this feature in the tag field to allow retrieval of this information (during WM-RR encoding) if a center-embedded clause interrupts the processing of an RC or a potential NC.

Thus the temporal encoding of (51) up to *fell* is:

NP	NP, Cl	NP, Cl, GapReq	Gap
<b>the + fact</b>	<b>the + vase</b>	<b>sue</b>	<b>Gap</b>
<b>E</b>	<b>E</b>	<b>E</b>	<b>bought</b>
E	E	E	V

51. The fact that the vase that Sue bought fell upset her.

Deletion of the RC will proceed correctly because N2 does not possess the *GapReq* feature, explaining the ease of an NC/RC [Gib98].

However, in the case of an RC/NC, N2 would contain all of N3's features. Again, due to the matching asymmetry, *Dtag* (containing N3's features) would match at N2, triggering premature inhibition and the erroneous removal of the RC from temporal WM. This explains the difficulty of an RC/NC [Gib00]. Therefore, an RC/NC should show the same pattern of N3-type and V2-drop effects as an RC/RC. Intuitively, this seems to be the case, although experimental studies have not been done to confirm this.



This analysis implies that an RC/RC that could have been an NC/RC, such as (52), should be processed correctly.

52. The proposal that the student who Amy advises made at the meeting intrigued us.

This is because the lack of a *GapReq* feature is determined at the start of the clause. Thus, N2 would not have the *GapReq* feature, allowing the inner RC to be correctly deleted. For such an RC/RC, N3-type should not have an effect, in contrast to unambiguous RC/RC. This prediction is unique to the TPARRSE model, and will be experimentally tested in the future. (However, note that perceived overall complexity may still remain high, due to the reanalysis triggered by *made at*. The specific prediction is that N3-type should have no effect at the processing of the final verb phrase in a self-paced reading study, in contrast to the effect previously observed for the unambiguous case [Whi04d]. )

### 16.1.3 Summary

The above complexity phenomena are explained by the following key assumptions:

- Subject NPs are stored serially in working memory, with accompanying syntactic features.
- Items are deleted from working memory by sequentially “searching” for target syntactic features; the search is initiated at the first item.
- There is an asymmetry to the search process due to the disinhibition mechanism; for inhibition to be triggered, all target features must be active in the list item, but not necessarily vice versa.

Premature deletion (starting at N2 rather than N3) follows from the first two assumptions, and explains RC/RC difficulty and the V2-drop effect. The blocking of this premature deletion via syntactic differences between N2 and N3 explains the relative ease of an NC/RC and a pronoun N3. Because deletion of the inner RC proceeds correctly, V2-drop is not felicitous for a pronoun N3, or for a center-embedded RC within a right-branching RC. The third assumption explains the difficulty of an RC/NC and a pronoun N2.

Thus, the proposed account is based on the nature of WM representations. This approach differs from previous accounts, which depend on capacity in some way. The DLT proposes capacity limitations in re-exciting previously processed constituents [Gib98]. The interference account is based on a maximal number of unattached subjects [Lew96]. In the Vosse & Kempen model [Vos00], correct attachment depends on relative inhibitory strength. We have seen in Chapter 13 that none of these proposals can fully explain the complexity phenomena. In contrast, the novel account offered by the TPARRSE model covers all of these phenomena. Next we see how the model also explains crossed-serial dependencies.

## 16.2 Crossed-Serial Dependencies

The processing of a center-embedding requires deletion of the embedded clause from WM, in order to correctly associate the separated higher-level subjects and predicates. However, processing of cross-serial dependencies does not require such deletion, because the subjects and verbs occur in the same order. Thus the verbs can be slotted into WM, and then the RR encoding can be read off. For example, consider processing of the English gloss of (9):

53. Joanna has the men Hans the horses helped to-teach to-feed.

The NPs are processed by Appending them to the noun list, giving <sup>1</sup>:

NP	NP	NP	NP
<b>joanna</b>	<b>the + men</b>	<b>hans</b>	<b>the + horses</b>
<b>E</b>			
E			

Then the verbs are Appended, giving:

NP	NP	NP	NP
<b>joanna</b>	<b>the + men</b>	<b>hans</b>	<b>the + horses</b>
<b>E</b>	<b>has-helped</b>	<b>to-teach</b>	<b>to-feed</b>
E	V	V	V

Now the verbs line up with their objects, and the information in WM can be RR-encoded in the manner of a right-branching sentence, giving:

**Joanna + has-helped@(the + men) + has-helped@to-teach@hans  
+ has-helped@to-teach@to-feed@(the + horses)**

which is equivalent to:

**joanna + has-helped@(the + man + to-teach@(hans + to-  
feed@(the + horses))**

Thus the sentence can be processed without deleting individual embedded clauses from WM. <sup>2</sup> Therefore, processing of crossed-serial dependencies is more efficient

---

<sup>1</sup>It is assumed that the auxiliary between the first and second subjects is saved, to be joined with the first verb.

<sup>2</sup>Partial RR encodings could be created by performing WM-RR encoding after each verb.

that center-embeddings, accounting for their reduced complexity[Bac86]. Because partial deletion is not required, making N3 a pronoun should not influence processing. This is exactly the result observed in [Kaa04].

We have now seen how the WM lists can be used like stacks for processing center-embeddings, and like queues to process crossed-serial dependencies. Because the lists are not actually stacks (i.e., there is no *pop* operation that removes the last element on a list), processing can break down, explaining complexity phenomena associated with double center-embeddings. In contrast, the dual seriality of the lists can be used directly to represent crossed-serial dependencies, enabling more reliable processing.

### 16.3 Interference in Working Memory

Recall that a WM list is comprised of banks of cells, and each activated list item draws on a subset of each of those banks. A new list item can therefore “steal” cells from already activated items. In this way, adding new items to WM can degrade the representation of previous items. The more similar the new item is to a previous item (i.e. the greater the number of matching bits), the more the opportunity for degradation of the previous item. Thus WM representations will become more degraded as the number of items increases, and as their similarity to each other increases. In addition to these factors, WM representations are also likely to be degraded over time. As an item continues to fire across oscillatory cycles, its activation may decay, the synchronization across positional banks may decrease, and/or constituent cells may fire in the wrong subcycle.

Recall that fast within-position inhibition and fast, weak across-position excitation are proposed to stabilize the representation of a single item (within a oscil-

latory subcycle), whereas slow inhibition serves to separate items. The amount of fast excitation would have to be rather narrowly tuned in order to avoid causing cells that should fire at upcoming subcycles to fire prematurely, but yet promote almost synchronous firing within a subcycle. Within each position, it is easiest to maintain separation between subcycles  $n$  and  $n+1$  when the polarity switches (0 to 1 or 1 to 0) across subcycles, due to the additional support of the fast inhibition. When the polarity doesn't switch between subcycles, it is more likely that cells belonging to subcycle  $n+1$  will fire prematurely in subcycle  $n$ . If the subcycles remain separated, this is not a problem. However, if this occurs on a large scale,  $n$  and  $n+1$  may collapse into a single subcycle. As the number of positions in which the polarity stays the same increases, this merging of subcycles becomes more likely, as there are fewer non-switching positions to drive strong separation between subcycles. Thus, when two similar items fire in consecutive subcycles, it is more difficult to maintain the separation between those items, accounting for the observed increase in perceived complexity as proximity between similar items increases [Lew02, Lee03, Kaa04].

In the DLT [Gib98, Gib00], capacity constraints emerge from the general assumption of a fixed pool of resources. The TPARRSE model provides a more detailed proposal for the nature of such capacity limitations. However, we have seen that capacity limitations per se cannot explain all of the complexity phenomena. Rather, the underlying structure of WM in the TPARRSE explains interference effects and general distance effects, while the proposed manipulations over those representations explain the specific pattern of complexity effects observed for doubly center-embedded clauses.

## Chapter 17

### Conclusion

I first speculate briefly on future directions of research related to the TPARRSE model, and then summarize the most important points of this dissertation.

#### 17.1 Future TPARRSE Research

As for explaining behavioral data, I have concentrated on complexity effects in the present work. I believe that the proposed TPARRSE representations could also explain reanalysis phenomena - namely why some ambiguous sentences are easy to reanalyze, and some are not. Current explanations assume that reanalysis operations are carried out a representation corresponding to a syntactic tree. In contrast, I propose that reanalysis operations are carried out over the WM lists. When an unexpected word or phrase occurs, the information encoded on the WM lists is reinterpreted to generate a new RR encoding of the previously processed material. If the established WM representation is incompatible with the correct interpretation, the sentence will be difficult to reanalyze. One avenue of future work will focus on specifying the nature of these reanalysis operations in detail, to allow a comprehensive account of the standard reanalysis cases.

Thus far, I have concentrated on the representations that encode hierarchical

structure, and have left the specification of the neural implementation of the parsing algorithm for future work. In general, I assume that the parsing rules are implemented by gating nodes which appropriately direct the flow of activation to neural areas which implement the WM variables, the WM lists, and the merge, bind, Append and Delete operations. I assume that the ability to perform these basic operations is innate. Thus language acquisition entails learning to store the relevant information in WM variables, and to invoke the appropriate primitive operations. A recent model of sequential-task learning, in which the basal ganglia gate individual “stripes” of prefrontal cortex [Ore03], seems ideal for this task. A stripe could correspond to a WM variable. Perhaps such an implemented model could develop the functionality of **CurSc**, **CurRR**, etc., as well as triggering of the required control operations. Therefore, future work on a neurally plausible implementation of the proposed parsing rules will focus on the application of this learning algorithm.

I would also like to pursue experimental investigations into the TPARRSE model. The model predicts that N3-type should not have an effect at the final verb for an RC/RC that could have been an NC/RC. This will be tested in a self-paced reading study. An EEG study has shown that theta-band amplitude increases as a sentence is processed [Bas02]. This amplitude may index syntactic working memory load [Bas02]. If so, this amplitude should be sensitive to the syntactic structure of a sentence. For example, working memory load should decrease following the completion of a center-embedded clause, implying that theta-band amplitude should also decrease at that point. This prediction will be tested in an EEG study

## 17.2 Conclusion

The goals of this work have been three-fold: (1) to advocate a particular approach to computational modeling; (2) to apply this approach to the problem of letter-position encoding; (3) to apply this approach to the problem of sentence parsing and the representation of hierarchical structure.

The approach places an emphasis upon developing computational theories, rather than on implementation of models. It emphasizes first understanding mature neural systems, rather than developing learning algorithms. Understanding what the mature system is doing would then provide strong constraints for investigations into how the system develops, because the endpoint is known.

The approach is truly interdisciplinary. Strong emphasis is placed on explaining the details of a wide range of relevant behavioral data. Such data provides clues as to what algorithms the brain is using. Although brain imaging is amazing, revolutionary, etc., I believe that behavioral data actually reveal more about *how* the brain is doing what it is doing. Consideration of neural architecture also provides information/constraints on what algorithms the brain is using. Knowledge of computational theories of neural processing provides the building blocks for formulating a model that meets the behavioral and neurobiological constraints.

The resulting model of a particular task specifies how information is mapped onto neural representations, and how one type of neural representation is transformed into another type. This abstract approach allows consideration of the big picture, without being limited by implementational constraints. Ideally, it leads to novel, verifiable predictions.

I first applied this approach to understanding how the brain encodes letter order during visual word recognition. The architecture of the visual system de-



terminated the lowest level of the model. Behavioral data on letter perceptibility, word priming, error patterns, and visual-field effects provided information about how this initial representation is transformed into a lexical representation. The goal of explaining this behavioral data led to the SERIOL model. The fact that the model led to an experiment which identified the source of the asymmetry of the length effect, which had been a subject of debate for half a century [Mel57, Bou73, Ell88, Jor03, Naz03], confirms the viability of the model and of the overall approach. These results, in conjunction with the experimental results on the N effect, demonstrate that although the SERIOL model is abstract, it is highly specific.

I have also applied this approach to the question of how hierarchical information is encoded during sentence processing. This is a more difficult task because (1) the problem is much harder and (2) there is much less relevant data available. Neural constraints were limited to generalities - fixed connectivity and local processing, although imaging data on oscillatory phenomena associated with verbal working memory were suggestive. Behavioral data was primarily in the form of complexity effects. The consideration of these factors, the computational demands of the task, and insights from the SERIOL model have led to the TPARRSE model. This model is unique in several ways: (1) the use of a neurobiologically - motivated sequential representation [Lis95] for stack-like and queue-like processing; (2) the dichotomy of representations used in temporal WM versus the thematic tree; (3) an account of complexity effects based on the specifics of WM representations and manipulations. It is hoped that this model too will lead to informative experimental results.

## Bibliography

- [Abb91] Abbott, L.F. (1991) Firing-Rate Models for Neural Populations. In Benhar, O., Bosio, C., Del Giudice, P. and Tabet, E., (Eds). *Neural Networks: From Biology to High-Energy Physics*. ETS Editrice: Pisa.
- [Abn89] Abney, S. (1989) A computational model of human parsing. *Journal of Psycholinguistic Research*, 18, 129-144.
- [Abn95] Abney, S. (1995) Chunks and dependencies: Bringing processing evidence to bear on syntax. In J. Cole, G. Green, and J. Morgan (Eds.), *Computational Linguistics and the Foundations of Linguistic Theory*. CSLI.
- [Abn91] Abney, S. & Johnson, M. (1991) Memory requirements and local ambiguities of parsing strategies. *Journal of Psycholinguistic Research*, 20, 233-250.
- [And91] Andrade, R. (1991) Cell excitation enhances muscarinic cholinergic responses in rat association cortex. *Brain Research*, 548, 81-93.
- [And89] Andrews, S. (1989) Frequency and neighborhood effects on lexical access: Activation or search? *Journal of Experimental Psychology: Learning, Memory and Cognition*, 15, 802-814.

- [And96] Andrews, S. (1996) Lexical retrieval and selection processes: Effects of transposed-letter confusability. *Journal of Memory and Language*, 35, 775-800.
- [And97] Andrews, S. (1997) The effect of orthographic similarity on lexical retrieval: Resolving neighborhood conflicts. *Psychonomic Bulletin and Review*, 4, 439-461.
- [Auc01] Auclair, L. & Chokron, S. (2001) Is the optimal viewing position in reading influenced by familiarity of the letter string? *Brain and Cognition*, 46, 20-24.
- [Bab99] Babyonyshev, M. & Gibson, E. (1999) The complexity of nested structures in Japanese. *Language*, 75, 423-450.
- [Bac86] Bach, E., Brown, C., & Marslen-Wilson, W. (1986) Crossed and nested dependencies in German and Dutch: A psycholinguistic study. *Language and Cognitive Processes*, 1, 249-262.
- [Bal95] Balota, D.A., & Abrams, R.A. (1995) Mental Chronometry: Beyond onset latencies in the lexical decision task. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 21, 1289-1302.
- [Bal94] Balota, D.A., Cortese, M.J., Sergent-Marshall, S.D., Spieler, D.H. & Yap, M.J. (2004) The English Lexicon Project: A web-based repository of descriptive and behavioral measures for 40,481 English words and nonwords. <http://elexicon.wustl.edu>, Washington University.

- [Bas02] Bastiaansen, M., van Berkum, J. & Hagoort P. (2002) Event-related theta power increases in the human EEG during online sentence processing. *Neuroscience Letters*, 323, 13-16.
- [Beh98] Behrman, M. *et al.* (1998) Visual complexity in letter-by-letter reading: “pure” alexia is not pure. *Neuropsychologia*, 36, 1115-1132.
- [Ber97] Berry, M.J., Warland, D.K. & Meister, M. (1997) The structure and precision of retinal spike trains. *Proceedings of the National Academy of Science*, 94, 5411-5416.
- [Ber84] Berwick R. & Weinberg, A. (1984) *The Grammatical Basis of Linguistic Performance*, Cambridge, MA: MIT Press.
- [Bie87] I. Biederman, I. (1987) Recognition-By-Components: a theory of human image understanding. *Psychological Review*, 94, 115-147.
- [Bin92] Binder, J. & Mohr, J. (1992) The topography of callosal reading pathways, a case-control analysis. *Brain*, 115, 1807-1826.
- [Bli73] Bliss, T. V. & Lomo, T. (1973) Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *Journal of Physiology*, 2, 331-356.
- [Bro93] Browosky, R. & Besner, D. (1993) Visual word recognition: A multi-stage activation model. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 19, 813-840.
- [Bou73] Bouma, H. (1973) Visual interference in the parafoveal recognition of initial and final letters of words. *Vision Research*, 13, 767-782.

- [Bol91] Boland, J.E. & Tanenhaus, M.K. (1991) The role of lexical representations in sentence processing. In G.B. Simpson (Ed.) *Understanding Word and Sentence*. Amsterdam: North-Holland. 331-366.
- [Bra04] *Brain and Language* (2004), 88(3).
- [Bur88] Burgess, C. & Simpson G. B. (1988) Cerebral hemispheric mechanisms in the retrieval of ambiguous word meanings. *Brain and Language*, 33, 86-103.
- [Bur02] Burt, J.S. & Tate, H. (2002) Does a reading lexicon provide orthographic representations for spelling? *Journal of Memory and Language*, 46, 518-543.
- [Bri68] Brindley, G. & Lewin, S. (1968) The sensations produced by electrical stimulation of the visual cortex. *Journal of Physiology*, 196, 479-493.
- [Bro72] Brown, J.W. (1972) *Aphasia, Apraxia, and Agnosia*. Charles C. Thomas: Springfield, Ill.
- [Bry94] Brysbaert, M. (1994) Interhemispheric transfer and the processing of foveally presented stimuli. *Behavioural Brain Research*, 64, 151-161.
- [Bry96] Brysbaert, M., Vitu, F. & Shroyens, W. (1996) The right visual field advantage and the optimal viewing position effect: On the relation between foveal and parafoveal word recognition. *Neuropsychology*, 10, 385-395.
- [Bry04] Brysbaert, M. (2004) The importance of interhemispheric transfer for foveal vision: A factor that has been overlooked in theories of visual

word recognition and object perception. *Brain and Language*, 88, 259-267.

- [Buc04] Buckmaster, P.S., Alonso, A., Canfield, D. R. & Amaral, D. G. (2004) Dendritic morphology, local circuitry, and intrinsic electrophysiology of principal neurons in the entorhinal cortex of macaque monkeys. *Journal of Comparative Neurology*, 470, 317-329.
- [Bur00] Burle, B. & Bonnet, M. (2000) High-speed memory scanning: A behavioral argument for a serial oscillatory model. *Cognitive Brain Research*, 9, 327-337.
- [Cap96] Caplan, D., Hildebrandt, N. & Makris, N. (1996) Location of lesions in *stroke* patients with deficits in syntactic processing in sentence comprehension. *Brain*, 119, 933-949
- [Car87] Carpenter, G.A. & Grossberg, S. (1987) A massively parallel architecture for a self-organizing neural pattern recognition machine, *Computer Vision, Graphics, and Image Processing*, 37, 54-115.
- [Cho59] Chomsky, N. (1959) On certain formal properties of grammars. *Information and Control*, 1, 91-112.
- [Cho59b] Chomsky, N. (1959) Review of B. F. Skinner, *Verbal Behavior*. *Language*, 35, 26-58.
- [Cho80] Chomsky, N. (1980) *Rules and Representations*. Columbia University Press: New York.
- [Chr99] Christiansen, M.H. & Chater, N. (1999) Connectionist natural language processing: The state of the art. *Cognitive Science*, 23, 417-437.

- [Cis03] Cisse, Y., Grenier, F., Timofeev, I. & Steriade, M. (2003) Electrophysiological properties and input-output organization of callosal neurons in cat association cortex. *Journal of Neurophysiology*, 89, 1402-1413.
- [Coh00] Cohen, L. *et al.* (2000) The visual word form area: spatial and temporal characterization of an initial stage of reading in normal subjects and posterior split-brain patients. *Brain*, 123, 291-307.
- [Coh02] Cohen, L. *et al.* (2002) Language-specific tuning of visual cortex? Functional properties of the Visual Word Form Area. *Brain*, 125, 1054-1069.
- [Coh03] Cohen, L. *et al.* (2003) Visual word recognition in the left and right hemispheres: anatomical and functional correlates of peripheral alexias. *Cerebral Cortex*, 13, 1313-1333.
- [Col77] Coltheart, M., Davelaar, E., Jonasson, J.T., & Besner, D. (1977) Access to the internal lexicon. In S. Dornic (Ed.) *Attention and Performance VI: The Psychology of Reading*. Academic Press.
- [Cor98] Cornelissen, P. *et al.* (1998) Coherent motion detection and letter position encoding. *Vision Research*, 38, 2181-2191.
- [Dav99] Davis, C. (1999) *The Self-Organising Lexical Acquisition and Recognition (SOLAR) model of visual word recognition*. Unpublished Doctoral Dissertation, University of South Wales.
- [Deh87] Dehaene, S., Changeux, J.P. & Nadal J.P. (1987) Neural networks that learn temporal sequences by selection. *Proceedings of the National Academy of Science*, 84, 2727-2731.

- [Deh02] Dehaene, S. *et al.* (2002) The visual word form area: A prelexical representation of visual words in fusiform gyrus. *Neuroreport*, 13, 321-325.
- [Deh04] Dehaene, S., Jobert, A., Naccache, L., Ciuciu, P., Poline, J.B., Le Bihan, D. & Cohen, L. (2004) Letter binding and invariant recognition of masked words: behavioral and neuroimaging evidence. *Psychological Science*, 15, 307-313.
- [Del00] Delorme, A., Richard, G. & Fabre-Thorpe, M. (2000) Ultra-rapid categorisation of natural scenes does not rely on colour cues: a study in monkeys & humans. *Vision Research*, 40, 2187-2200.
- [Dem92] Demonet, J. *et al.* (1992) The anatomy of phonological and semantic processing in normal subjects. *Brain*, 115, 1753-1768.
- [Duc03] Ducrot, S., Lete, B., Sprenger-Charolles, L., Pynte, J. & Billard, C (2003) The Optimal Viewing Position Effect in Beginning and Dyslexic Readers. *Current Psychology Letters*, 10, <http://cpl.revues.org/document99.html>.
- [Eng01] Engel, A. K. & Singer, W. (2001) Temporal binding and the neural correlates of sensory awareness. *Trends in Cognitive Science*, 5, 16-25.
- [Evi99] Eviatar, Z. (1999) Cross-language tests of hemispheric strategies in reading nonwords. *Neuropsychology*, 13, 498-515.
- [Ell88] Ellis, A. W., Young, A.W. & Anderson, C. (1988) Modes of word recognition in the left and right cerebral hemispheres. *Brain and Language*, 35, 254-273.



- [Elm90] Elman, J.L. (1990) Finding structure in time. *Cognitive Science*, 14, 179-211.
- [Est76] Estes, W.K., Altemeyer, D.H. & Reder, S.M. (1976) Serial position functions for letter identification at brief and extended exposure durations. *Perception & Psychophysics*, 19, 1-15.
- [Eve81] Evett, L. J. & Humphreys, G. W. (1981) The use of abstract graphemic information in lexical access. *Quarterly journal of Experimental Psychology*, 30, 569-575.
- [Far96] Farid, M. & Grainger, J. (1996) How initial fixation position influences visual word recognition: a comparison of French and Arabic. *Brain and Language*, 53, 351-368.
- [Fac01] Facoetti, A. & Molteni, M. (2001) The gradient of visual attention in developmental dyslexia. *Neuropsychologia*, 39, 352-357.
- [Fel01] Fellous, J.M., Houweling, A.R., Modi, R.H., Rao, R.P., Tiesinga, P.H. & Sejnowski T. J. (2001) Frequency dependence of spike timing reliability in cortical pyramidal cells and interneurons. *Journal of Neurophysiology*, 85, 1782-1787.
- [Fer92] Ferrara, V.P., Nealey, T.A. & Maunsell, J.H. (1992) Mixed parvocellular and magnocellular geniculate signals in visual area V4. *Nature*, 358, 756-761.
- [Fie02] Fiebach, C. *et al.* (2002) fMRI evidence for dual routes to the mental lexicon in visual word recognition. *Journal of Cognitive Neuroscience*, 14, 11-23.

- [Fit04] Fitch, T. & Hauser, M. (2004) Computational constraints on syntactic processing in a nonhuman primate, *Science*, 377-380.
- [Fos02] Foster, D. H. & Gilson, S. J. (2002) Recognizing novel three-dimensional objects by summing signals from parts and views. *Proceedings of Royal Society of London. Series B Biological Science*, 269, 1939-1947.
- [Fre76] Frederiksen, J.R. & Kroll, J.F. (1976) Spelling and sound: Approaches to the internal lexicon. *Journal of Experimental Psychology: Human Perception and Performance*. 2, 361-379.
- [Fri01] Friedmann, N. & Gvion, A. (2001) Letter position dyslexia. *Cognitive Neuropsychology*, 18, 673-696.
- [Fuk88] Fukushima, K. (1988) Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1, 119-130.
- [Ger01] Gers, F. A. & J. Schmidhuber, J. (2001) LSTM Recurrent Networks Learn Simple Context Free and Context Sensitive Languages. *IEEE Transactions on Neural Networks* 12, 1333-1340.
- [Ges95] Geschwind, N. (1965) Disconnection syndromes in animals and man. *Brain*, 88, 237-294 and 585-644.
- [Gib98] Gibson, E. (1998) Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68, 1-75.
- [Gib99] Gibson, E., & Thomas, J. (1999) Memory limitations and structural forgetting: The perception of complex ungrammatical sentences as grammatical. *Language and Cognitive Processes*, 14, 225-248.

- [Gib00] Gibson E. (2000). The dependency locality theory. In Marantz, Miyashita & O'Neil (Eds.), *Image, Language, Brain*. Cambridge, MA: MIT Press. 95-126.
- [Gib04] Gibson, E., Desmer, T., Watson, D., Grodner, D., & Ko, K. (2004) Reading relative clauses in English. Submitted.
- [Gra89] Grainger, J., O'Regan, J.K., Jacobs, A.M. & Segui, J. (1989) On the role of competing word units in visual word recognition: The neighborhood frequency effect. *Perception and Psychophysics*, 5, 189-195.
- [Gra96] Grainger, J. & Jacobs, A. (1996) Orthographic processing in visual word recognition: A multiple readout model. *Psychological Review*, 103, 518-565.
- [Gra04a] Granier, J.P. & Grainger, J. (2004) Letter position information and printed word perception: The relative-position priming constraint. Submitted.
- [Gra04b] Grainger, J. & Whitney, C. (2004) Does the human mind read words as a whole? *Trends in Cognitive Sciences*, 8, 58-59.
- [Ham82] Hammond, E.J. & Green, D.W. (1982) Detecting targets in letter and non-letter arrays. *Canadian Journal of Psychology*, 36, 67-82.
- [Har75] Harcum, E.R. & Nice, D.S. (1975) Serial processing shown by mutual masking of icons. *Perceptual and Motor Skills*, 40, 399-408.
- [Har01] Hari, R., Renvall, H. & Tanskanen, T. (2001) Left hemisphere neglect in dyslexic adults. *Brain*, 124, 1373-1380.

- [Hau04] Hauk, O. & Pulvermuller F. (2004) Effects of word length and frequency on the human event-related potential. *Clinical Neurophysiology*, 115, 1090-1103.
- [Hay03] Hayward, G. (2003) After the viewpoint debate: where next in object recognition? *Trends in Cognitive Sciences*, 7, 425-427.
- [Hel95] Hellige, J.B., Cowin, E.L. & Eng, T.L. (1995) Recognition of CVC syllables from LVF, RVF, and central locations: Hemispheric differences and interhemispheric interactions. *Journal of Cognitive Neuroscience*, 7, 258-266.
- [Hle97] Hellige, J.B. & Scott, G.B. (1997) Effects of output order on hemispheric asymmetry for processing letter trigrams. *Brain and Language*, 59, 523-30.
- [Hell99] Hellige, J.B. & Yamauchi, M. (1999) Quantitative and qualitative hemispheric asymmetry for processing Japanese kana. *Brain & Cognition*, 40, 453-463.
- [Hel99] Helenius, P., Tarkiainen, A., Cornelissen, P. L., Hansen, P. L., & Salmelin (1999) Dissociation of normal feature analysis and deficient processing of letter-strings in dyslexic adults. *Cerebral Cortex*, 9, 476-483.
- [Hin90] Hinton, G.E. (1990) Mapping part-whole hierarchies into connectionist networks. *Artificial Intelligence*, 46, 47-75.
- [Hoc97] S. Hochreiter, S. & Schmidhuber, J. (1997) Long Short-Term Memory. *Neural Computation*, 9, 1735-1780.

- [Hop95] Hopfield, J.J. (1995) Pattern recognition computation using action potential timing for stimulus representation. *Nature*, 376, 33-36.
- [Hum90] Humphreys, G.W., Evett, L.J. & Quinlan, P.T. (1990) Orthographic processing in visual word identification. *Cognitive Psychology*, 22, 517-560.
- [Hum92] Hummel, J. & Biederman, I. (1992) Dynamic binding in a network for shape recognition. *Psychological Review*, 99, 487-517.
- [Hum97] Hummel, J.E., & Holyoak, K.J. (1997) Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review*, 104, 427-466.
- [Ino09] Inouye, T. (1909) Die sehstörungen bei schussverletzungen der kortikalen sehspähre nach beobachtungen an versündeten der letzten japanische kriege. W. Engelmann.
- [Jac02] Jackendoff, R. (2002) *Foundations of Language*. Oxford University Press.
- [Jen02] Jensen, O & Tesche, C. (2002) Frontal theta activity in humans increases with memory load in a working memory task. *European Journal of Neuroscience*, 15, 1-6.
- [Jor03] Jordan, T.R., Patching, G. R. & Thomas, S. M. (2003) Assessing the role of hemispheric specialisation, serial-position processing, and retinal eccentricity in lateralised word recognition. *Cognitive Neuropsychology*, 20, 49-71.

- [Kaa04] Kaan, E. & Vasic, N. (2004) Cross-serial dependencies in Dutch: Testing the influence of NP type on processing load. *Memory and Cognition*, 32, 175-184.
- [Kan95] Kanerva, P. (1995) A family of binary spatter codes. In F. Fogelman-Soulie and P. Gallineri (eds.), *ICANN '95, Proceedings International Conference on Artificial Neural Networks*, 1, 517-522.
- [Kha04] Khader, P & Rosler, F. (2004) EEG power and coherence analysis of visually presented nouns and verbs reveals left frontal processing difference. *Neuroscience Letters*, 354, 111-114.
- [Kli96] Klimesch, W. (1996) Memory processes, brain oscillations and EEG synchronization. *International Journal of Psychophysiology*, 24, 61-100.
- [Kli99] Klimesch, W. (1999) EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. *Brain Research Reviews*, 29, 169-195.
- [Kli01] Klimesch, W., Doppelmayr, M., Wimmer, H., Schwaiger, J., Rohm, D., Gruber, W. & Hutzler, F. (2001) Theta band power changes in normal and dyslexic children. *Clinical Neurophysiology*, 112, 1174-1185.
- [Kor85] Koriat, A. & Norman, J. (1985) Reading rotated words. *Journal of Experimental Psychology: Human Perception and Performance*, 11, 490-508.
- [Kwa99a] Kwantes, P.J. & Mewhort, D.J. (1999) Modeling lexical decision and word naming as a retrieval process. *Canadian Journal of Experimental Psychology*, 53, 306-315.

- [Kwa99b] Kwanten, P.J. & Mewhort, D.J. (1999) Evidence for sequential processing in visual word recognition. *Journal of Experimental Psychology: Human Perception and Performance*, 25, 276-231.
- [Lan01] Lane, P. & Henderson, J. (2001) Incremental Syntactic Parsing of Natural Language Corpora with Simple Synchrony Networks. *IEEE Transactions on Knowledge and Data Engineering*, 13(2).
- [Lav01a] Lavidor, M., Ellis, A., Shillcock, R. & Bland, T. (2001) Evaluating a split processing model of visual word recognition: Effects of word length. *Cognitive Brain Research*, 12, 265-272.
- [Lav01b] Lavidor, M., Babkoff, H. & Faust, M. (2001) Analysis of standard and non-standard visual format in the two hemispheres. *Neuropsychologia*, 39, 430-439.
- [Lav02a] Lavidor, M. & Ellis, A. (2002) Word length and orthographic neighborhood size effects in the left and right cerebral hemispheres. *Brain and Language*, 80, 45-62.
- [Lav02b] Lavidor, M. & Ellis, A. (2002) Orthographic Neighborhood effects in the right but not in the left cerebral hemisphere. *Brain and Language*, 80, 63-76.
- [Lav02c] Lavidor, M., Ellis, A.W. & Pansky, A. (2002) Case alternation and length effects in lateralized word recognition: Studies of English and Hebrew. *Brain and Cognition*, 50, 257-271.

- [Lav03] Lavidor, M., & Walsh, V. (2003) A magnetic stimulation examination of orthographic neighborhood effects in visual word recognition. *Journal of Cognitive Neuroscience*. 2003, 15, 354-363.
- [Lav04a] Lavidor, M., Hayes, A., Shillcock, R., & Ellis, A.W. (2004) Evaluating a split processing model of visual word recognition: effects of orthographic neighborhood size. *Brain and Language*, 88, 312-320.
- [Lav04b] Lavidor, M. & Walsh, V. (2004) The nature of foveal representation. *Nature Reviews Neuroscience*, 5, 729-735.
- [Lee03] Lee, S. & Nakayama, M. (2003) Effects of syntactic and phonological similarity in Korean center-embedding constructions. Poster presented at the 16th Annual CUNY Conference on Sentence Processing, Cambridge, MA.
- [Lef04] Leff, A. (1994) A historical review of the representation of the visual field in primary visual cortex with special reference to the neural mechanisms underlying macular sparing. *Brain and Language*, 88, 268-278.
- [Leg01] Legge, G.E., Mansfield, J.S. & Chung S.T. (2001) Psychophysics of reading. XX. Linking letter recognition to reading speed in central and peripheral vision. *Vision Research*, 41, 725-743.
- [Lef78] Lefton, L.A., Fisher, D.F. & Kuhn, D.M. (1978) Left-to-right processing of alphabetic material is independent of retinal location. *Bulletin of the Psychonomic Society*, 112, 171-174.



- [Lew00] Levitan, S. & Reggia J.A. (2000) A computational model of lateralization and asymmetries in cortical maps. *Neural Computation*, 12, 2037-2062.
- [Lew96] Lewis, R. L. (1996). Interference in short-term memory: The magical number two (or three) in sentence processing. *The Journal of Psycholinguistic Research*, 25, 93-115.
- [Lew95] Lewis, R. L. (1998) Renanalysis and limited repair parsing: leaping off the garden path. In J. Fodor, & F. Fereirra (Eds.), *Reanalysis in sentence processing*. Dordrecht: Kluwer, 247-285.
- [Lew02] Lewis, R. L. & Nakayama, M. (2002). Syntactic and positional similarity effects in the processing of Japanese embeddings. In Nakayama, M. (Ed.) *Sentence Processing in East Asian Languages*. Stanford: CSLI Publications.
- [Lie03] Liederman, J., McGraw-Fisher, J., Schulz, M., Maxwell, C., Theoret, H. & Pascual-Leone, A. The role of motion direction selective extrastriate regions in reading: a transcranial magnetic stimulation study. *Brain and Language*, 85, 140-155.
- [Lis95] Lisman, J.E., Idiart, M.A.P. (1995) Storage of  $7 \pm 2$  short-term memories in oscillatory subcycles. *Science*, 267, 1512-1515.
- [Lor04] Lorusso, M.L., Facoetti, A. & Molteni, M. (2004) Hemispheric, attentional, and processing speed factors in the treatment of developmental dyslexia. *Brain and Cognition*, 55, 341-348.

- [Lov93] Lovegrove, W. (1993) Wolf's weakness in the transient visual system: A causal factor in dyslexia? *Annals of the the New York Academy of Sciences*, 682, 57-69.
- [Mac04] MacKeben, M., Trauzettel-Klosinski, S., Reinhard, J., Durrwachter, U., Adler, M., & Klosinski, G. (2004). Eye movement control during single-word reading in dyslexics. *Journal of Vision*, 4, 388-402, <http://journalofvision.org/4/5/4/>, doi:10.1167/4.5.4.
- [Mag01] Magee, J.C. (2001) Dendritic mechanisms of phase precession in hippocampal CA1 pyramidal neurons. *Journal of Neurophysiology*, 86, 528-532.
- [Mar80] Marcus, M. (1980) *A theory of syntactic recognition for natural language*. MIT Press: Cambridge, MA.
- [Mar01] Marcus, G. (2001) *The Algebraic Mind*. MIT Press: Cambridge, MA.
- [Mau83] Maunsell, J.H.R. & Van Essen, D.C. (1983) Functional properties of neurons in the middle temporal visual area of the macaque monkey. I. Selectivity for stimulus direction, speed, and orientation. *Journal of Neurophysiology*, 19, 1127-1147.
- [Mau90] Maunsell, J.H., Nealey, T.A. & De Priest, D.D. (1990) Magnocellular and parvocellular contributions to responses in the middle temporal visual area (MT) of the macaque monkey. *Journal of Neuroscience*, 10, 3323-3334.

- [Mas82] Mason, M. (1982) Recognition time for letters and nonletters: Effects of serial position, array size, and processing order. *Journal of Experimental Psychology*, 8, 724-738.
- [McC03] McCandliss, B., Cohen, L. & Dehaene, S. (2003) The visual word form area: expertise for reading in the fusiform gyrus. *Trends in Cognitive Sciences*, 7, 293-299.
- [McC81] McClelland, J.L. & Rumelhart, D.E. (1981) An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review*, 88, 375-407.
- [Mel57] Melville, J.P., (1957) Word-length as a factor in differential recognition. *American Journal of Psychology*, 70, 316-318.
- [Mew69] Mewhort, D.J.K., Merikle, P.M. & Bryden, M.P. (1969) On the transfer from iconic to short-term memory. *Journal of Experimental Psychology*, 81, 89-94.
- [Mon04] Monaghan, P., Shillcock, R. & McDonald, S. (2004) Hemispheric asymmetries in the split-fovea model of semantic processing. *Brain and Language*, 88, 339-354.
- [Mon98] Montant, M. Nazir, T.A., Poncet, M. (1998) Pure alexia and the viewing position effect in printed words. *Cognitive Neuropsychology*, 15, 93-140.
- [Moz91] Mozer, M. (1991) *The Perception of Multiple Objects: A Connectionist Approach*. MIT Press.

- [Naz03] Nazir, T. A. (2003) On hemispheric specialization and visual field effects in the perception of print: A comment on Jordan, Patching, and Thomas. *Cognitive Neuropsychology*, 20, 73-80.
- [Naz04a] Nazir, T.A. (2004) Reading habits, perceptual learning, and recognition of printed words. *Brain and Language*, 88, 294-311.
- [Naz04b] Nazir, T.A., Kajii, N., Frost, R & Osaka, N. (2004) Script characteristics modify the way we perceive isolated words: Visual field effects in the perception of French, Hebrew, Kanji and Hiragana words. In preparation.
- [New04] New, B., Ferrand, L., Pallier, C. & Brysbaert, M. (2004) Re-examining word length effects in visual word recognition: New evidence from the English Lexicon Project. Submitted.
- [Nic76] Nice, D.E, & Harcum, E.R. (1976) Evidence from mutual masking for serial processing of tachistoscopic letter patterns. *Perceptual and Motor Skills*, 42, 991-1003.
- [Nig93] Nigrin, A. (1993) *Neural Networks for Pattern Recognition*. MIT Press.
- [Nob94] Nobre, A., Allison, T. & McCarthy, G. (1994) Word recognition in the human inferior temporal lobe. *Nature*, 372, 260-263.
- [Ore84] O'Regan, J.K., Levy-Schoen, A., Pynte, J. & Brugailere, B. (1984) Convenient fixation location within isolated words of different length and structure. *Journal of Experimental Psychology: Human Perception and Performance*, 18, 185-197.

- [Ore03] O'Reilly, R.C. & Frank, M. J. (2003) Making Working Memory Work: A Computational Model of Learning in the Prefrontal Cortex and Basal Ganglia. ICS Technical Report 03-03, University of Colorado, Boulder.
- [Pea98] Pearlmutter, N. & Mendelsohn, A. (1998) Serial versus parallel sentence processing. Paper presented at the 11th Annual CUNY Conference on Human Sentence Processing. Rutgers University, New Jersey.
- [Per98] Perea, M. (1998) Orthographic neighbors are not all equal: Evidence using an identification technique. *Language and Cognitive Processes*, 13, 77-90.
- [Per03] Perea, M. & Lupker, S. J. (2003) Transposed-letter confusability effects in masked form priming. In: S. Kinoshita and S.J. Lupker, Eds., *Masked Priming: State of the Art*. Psychology Press. 97-120.
- [Per04] Perea, M. & Lupker, S. J. (2004) Can CANISO activate CASINO? Transposed-letter similarity effects with nonadjacent letter positions. *Journal of Memory and Language*, 51, 231-246.
- [Per95] Peressotti, F. & Grainger, J. (1995) Letter-position coding in random consonant arrays. *Perception & Psychophysics*, 57, 875-890.
- [Per99] Peressotti, F. & Grainger, J. (1999) The role of letter identity and letter position in orthographic priming. *Perception & Psychophysics*, 61, 691-706.
- [Plsa93] Plaut, D.C. & McClelland, J.L. (1993) Generalization with componential attractors: Word and nonword reading in an attractor network.

In Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society, 824-829, Erlbaum.

- [Pla95] Plate, T.A. (1995), Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6, 623-641.
- [Pog90] Poggio, T. & Edelman, S. (1990) A network that learns to recognize three-dimensional objects. *Nature*, 343, 263-266.
- [Pol90] Pollack, J. (1990) Recursive distributed representations. *Artificial Intelligence*, 46, 77-105.
- [Pol02] Polk, T. & Farah, M. (2002) Functional MRI evidence for an abstract, not perceptual, word-form area. *Journal of Experimental Psychology: General*, 131, 65-72.
- [Pri03] Price, C. & Devlin, J. (2003) The myth of the visual word form area. *Neuroimage*, 19, 473-481.
- [Pul03] Pulvermuller, F. (2003) *The Neuroscience of Language : On Brain Circuits of Words and Serial Order*. Cambridge University Press.
- [Ore00] O'Reilly, R.C & Munakata, Y. (2000) *Computational Explorations in Cognitive Neuroscience*. MIT Press.
- [Rag01] Raghavachari S., Kahana M., Rizzuto D., Caplan J., Kirschen M., Bourgeois B., Madsen J. & Lisman J. (2001) Gating of human theta oscillations by a working memory task. *Journal of Neuroscience*, 21, 3175-3183.
- [Ray75] Rayner, K. (1975) Parafoveal identification during a fixation in reading. *Acta Psychologica*, 4, 271-82.

- [Ray76] Rayner, K. & McConkie, G. (1976) What guides a reader's eye movements? *Vision Research*, 16, 829-837.
- [Reg01] Reggia, J.A., Goodall, S.M., Shkuro, Y. & Glezer, M. (2001) The callosal dilemma: explaining diaschisis in the context of hemispheric rivalry via a neural network model. *Neurological Research*, 23, 465-471.
- [Rie97] Rieke, F., Warland, D., De Ruyter van Steveninck, R. & Bialek, W. (1997) *Spikes: Exploring the Neural Code*, MIT Press.
- [Rey04] Reynolds, M. & Besner, D. (2004) Neighborhood density, word frequency and spelling-sound regularity effects in naming: Similarities and differences between skilled readers and the Dual Route Cascaded computational model. *Canadian Journal of Experimental Psychology*, 13-31.
- [Roh02] Rohde, D.L.T. (2002). A connectionist model of sentence comprehension and production. Unpublished PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.
- [Roh01] Rohm, D., Klimesch, W., Haider, H. & Doppelmayr, M. (2001) The role of theta and alpha oscillations for language comprehension in the human electroencephalogram. *Neuroscience Letters*, 310, 137-40.
- [Sch04] Schoonbaert, S. & Grainger, J. (2004). Letter position coding in printed word perception: Effects of repeated and transposed letters. *Language and Cognitive Processes*. In press.

- [Sei89] Seidenberg, M.S. & McClelland, J.L. (1989) A distributed, developmental model of word recognition and naming. *Psychological Review*, 96, 523-568.
- [Sha93] Shastri, L., & Ajjanagade, V. (1993) From simple associations to systematic reasoning. *Behavioral and Brain Sciences*, 16, 417-494.
- [Sha99] Shastri, L. (1999) Advances in SHRUTI – A neurally motivated model of relational knowledge representation and rapid inference using temporal synchrony. *Applied Intelligence*. 11, 79-108.
- [She76] Sheil, B.A. (1976) Observations on context-free parsing. *Statistical Methods in Linguistics*, 6, 71-109.
- [Sik02] Sikaluk, P. D., Sears, C. R., and Lupker, S. J. (2002). Orthographic neighborhood effects in lexical decision: The effects of nonword orthographic neighborhood size. *Journal of Experimental Psychology: Human Perception and Performance*, 28, 661-681.
- [Sta75] Stanners, R.F., Jastrzembski, J., E., & Westbrook, A. (1975) Frequency and visual quality in a word-nonword discrimination task. *Journal of Verbal Learning and Verbal Behaviour*. 14, 259-264.
- [Ste97] Stein, J. & Walsh, V. (1997) To see but not to read; the magnocellular theory of dyslexia. *Trends in Neuroscience*, 20, 147-152.
- [Ste03] Stevens M. & Grainger J. (2003) Letter visibility and the viewing position effect in visual word recognition. *Perception & Psychophysics*, 65, 133-151.



- [Sus01] Sussman, R.S. & Sedivy, J.C. (2001) The time-course of processing syntactic dependencies: Evidence from eye movements during spoken narratives. In J.S. Magnuson and K. M. Crosswhite (Eds.) University of Rochester Working Papers in the Language Sciences, 2, 52-70.
- [Tar99] Tarkiainen, A., Helenius, P., Hansen, P. C., Cornelissen, P. L., & Salmelin (1999) Dynamics of letter string perception in the human occipitotemporal cortex . *Brain*, 11, 2119-2132.
- [Tal00] Talcott, J.B., Witton, C., McLean, M.F., Hansen, P.C., Rees, A., Green, G.G. & Stein JF (2000). Dynamic sensory sensitivity and children's word decoding skills. *Proceedings of the National Academy of Sciences*, 97, 2952-2957.
- [Tou88] Touretzky, D. S. & Hinton, G. E. (1988) A distributed connectionist production system. *Cognitive Science*, 12, 423-466.
- [Van02] Van Rullen, R. & Thorpe, S.J. (2002) Surfing a spike wave down the ventral stream. *Vision Research* , 42, 2593-2615.
- [Vic96] Victor, J.D. & Purpura, K.P. (1996) Nature and precision of temporal coding in visual cortex: a metric-space analysis. *Journal of Neurophysiology*, 76, 1310-1326.
- [Vid01] Vidyasagar, T.R.(2001) From attentional gating in macaque primary visual cortex to dyslexia in humans. *Progress in Brain Research*, 134, 297-312.
- [Vid04] Vidyasagar, T.R. (2004) Neural underpinnings of dyslexia as a disorder of visuo-spatial attention. *Clinical Experimental Optometry*, 87, 4-10.

- [Vos00] Vosse, T. & Kempen, G. (2000) Syntactic structure in human parsing: A computational model based on competitive inhibition and a lexicalist grammar. *Cognition*, 75, 105-143.
- [War02a] Warren, T. & Gibson, E. (2002) The influence of referential processing on sentence complexity. *Cognition*, 85, 79-112.
- [War02b] Warren, T. & Gibson, E. (2002) Evidence for a constituent-based distance metric in distance-based complexity theories. Poster presented at the CUNY Conference on Human Sentence Processing.
- [War80] Warrington, E. & Shallice, T. (1980) Word-form dyslexia. *Brain*, 103, 99-112.
- [Was95] Wassle, H., Grunert, U., Rohrenbeck, J. & Boycott, B. (1989) Cortical magnification factor and the ganglion cell density of the primate retina. *Nature*, 341, 643 - 646.
- [Wes87] Westheimer G., (1987) Visual Acuity. Chapter 17. In Moses, R. A. and Hart, W. M. (eds) *Adler's Physiology of the Eye, Clinical Application*. St. Louis: The C. V. Mosby Company.
- [Whi01a] Whitney, C. (2001) How the brain encodes the order of letters in a printed word: The SERIOL model and selective literature review. *Psychonomic Bulletin and Review*, 8, 221-243.
- [Whi01b] Whitney, C. (2001) Position-specific effects within the SERIOL framework of letter-position coding. *Connection Science*, 13, 235-255.
- [Whi02] Whitney, C. (2002) An explanation of the length effect for rotated words. *Cognitive Systems Research*, 3, 113-119.

- [Whi04a] Whitney, C. (2004) Hemisphere-specific effects in word recognition do not require hemisphere-specific modes of access. *Brain and Language*, 88, 279-293.
- [Whi99] Whitney, C. & Berndt, R.S. (1999) A new model of letter string encoding: Simulating right neglect dyslexia. *Progress in Brain Research*, 121, 143-163.
- [Whi04b] Whitney, C. & Lavidor, M. (2004) Orthographic neighborhood effects: The SERIOL model account. Submitted.
- [Whi04c] Whitney, C. & Lavidor, M. (2004) Why word length only matters in the left visual field. *Neuropsychologia*. In press.
- [Whi04d] Whitney & Weinberg (2004) Interaction between Subject Type and Ungrammaticality in Doubly Center-Embedded Relative Clauses. Poster presented at the 17th Annual CUNY Sentence Processing Conference, University of Maryland.
- [Wol74] Wolford, G. & Hollingsworth S. (1974) Retinal location and string position as important variables in visual information processing. *Perception & Psychophysics*, 16, 437-442.
- [You85] Young, A. W. & Ellis A.W. (1985) Different methods of lexical access for words presented to the left and right visual hemifields. *Brain and Language*, 24, 326-358.
- [Zie98] Ziegler, J. & Perry, C. (1998) No more problems in Coltheart's neighborhood: resolving neighborhood conflicts in the lexical decision task. *Cognition*, 68, B53-B62.