The Influence of Semantic Number Representations
on Arithmetic Fact Retrieval

by

John William Whalen

A dissertation submitted to The Johns Hopkins University
in conformity with the requirements for the degree of
Doctor of Philosophy

July 30, 2000
The Influence of Semantic Number Representations on Arithmetic Fact Retrieval

John William Whalen

Advisor: Dr. Michael McCloskey

ABSTRACT

This dissertation explores how arithmetic facts such as $6 \times 7 = 42$ are stored and retrieved in memory. The present work examines the existing literature on retrieval of arithmetic facts, identifying the major findings from studies of normal arithmetic, acquired deficits after brain damage, and artificial arithmetic operations. Several theories of arithmetic fact retrieval are summarized, each of which either attempts to account for arithmetic fact retrieval performance, or the form in which arithmetic facts are stored. Each theory is evaluated in terms of how well it accounts for the observed phenomena, and issues which remain to be resolved are identified. Two theoretical positions are considered viable and testable: one suggests that arithmetic facts are stored in a phonological form. This theory is evaluated using the performance of a brain-damaged patient. The patient's pattern of performance is found to be incompatible with this phonological storage hypothesis. The second viable theoretical position proposes that arithmetic facts are stored in an abstract semantic form. This notion is incorporated into a new theory of arithmetic fact retrieval which suggests that arithmetic facts are retrieved from a network of facts using a magnitude representation of the problem operands. This Semantic Network Retrieval Theory is described in detail and simulated using a distributed network representation, demonstrating that the theory is able to account for the major arithmetic phenomena. According to this theory, the main source of competition during retrieval are answers with operands which are close in magnitude to the correct operands. An experimental evaluation of this hypothesis using an artificial arithmetic operation revealed performance benefits when the number of facts with operands close in magnitude to the actual operands was
reduced. This finding provides some additional support for the Semantic Network Retrieval Theory.
ACKNOWLEDGMENTS

This dissertation was made possible by the assistance, mentoring, and support of many people. My advisor Michael McCloskey offered keen insights and suggestions while allowing me the freedom to pursue my own ideas and interests. Perhaps most valuable is his keen ability to look deep into the problems and theoretical questions and pull out the fundamental challenges and issues. I hope to someday emulate his willingness to challenge his own ideas and to produce clear thinking in every aspect of research. I am grateful to Brenda Rapp for her thought provoking ideas, warmth, and encouragement. I also want to thank Paul Smolensky for introducing me to new approaches and perspectives.

Of great support to me during my dissertation were my fellow graduate students and friends in Cognitive Science. Elwyn Leek, in particular was a great friend --- willing to listen to seemingly endless questions about arithmetic (and other unassociated ramblings) whenever I asked. Mary Rupp and Debbie Grover were always terrific in supporting me and putting my various stresses in perspective.

I want to offer a special thanks to Mark Ashcraft who helped me continue my research while in Cleveland. Under no obligations Mark instantly fell into the role of advisor and colleague, providing unlimited discussion time, lab space, and subjects.

Finally, I must acknowledge the tremendous support from my family and my wife Patricia. In the back of my mind I could always hear you all cheering me on.
TABLE OF CONTENTS

I: Introduction ................................................................. 1

II: Basic Arithmetic Phenomena ........................................... 3
    Normal Arithmetic Fact Retrieval ................................... 3
    Problem Size Effect .................................................... 3
    Error Patterns .......................................................... 4
    Error Priming ............................................................ 6

Dyscalculia ................................................................. 8
    Acquired Dyscalculia .................................................. 8
    Developmental Dyscalculia ......................................... 10

Artificial Arithmetic Operations ....................................... 11
    Graham and Campbell: Alphaplication ......................... 12
    Zbrodoff & Logan: Alphabet Arithmetic ....................... 14
    Harley: Pelification ................................................... 15
    Whalen, Lindemann, Goldschlager & McCloskey: Pelification ... 19
    Summary of Artificial Arithmetic Findings ................... 22

III: Theories of Arithmetic Fact Retrieval ............................ 23
    Theories Accounting for the Problem Size Effect and Error Patterns 23
    Table Search Theories .............................................. 23
    Ashcraft's Network Retrieval Theory ............................ 25
    Siegler's Distribution of Associations Theory ............... 29
    Non-Retrieval Strategies During Adult Arithmetic Performance 33
    In What Form Are Arithmetic Facts Stored? .................... 36
    Dehaene's Phonological Storage Hypothesis ................... 37
    McCloskey's Abstract Retrieval Position ....................... 40
    Campbell's Multi-Format Position ............................. 41

Simulations of Arithmetic Fact Retrieval ........................... 44
    Anderson, Viscuso & Spoer (1989) .............................. 44
    Campbell's Network Interference Simulation .................. 46
    McCloskey and Lindemann: MATHNET ........................... 48

Unresolved Issues in Simple Arithmetic ............................ 52
    What are the Sources of the Problem Size Effect? .......... 52
What are the Sources of Errors in Arithmetic? ................. 52
How Often are Non-Retrieval Strategies Used? ............... 53
How Are Arithmetic Facts Represented in Memory?....... 53
IV: Evaluation of the Phonological Storage Hypothesis ........ 55
Case History .......................................................... 57
Numerical Comprehension and Production .............. 57
Arithmetic Fact Retrieval ........................................ 58
Evaluation of the Phonological Storage Hypothesis ....... 59
Method .............................................................. 64
Scoring Procedure ................................................. 65
Results .............................................................. 65
Additional Evaluation of Phonological Conversion Ability... 71
Additional Evaluation of the Phonological Storage Hypothesis 72
Method .............................................................. 74
Scoring Procedure ................................................. 74
Results .............................................................. 74
Discussion ................................................................... 75
Alternative Hypotheses ........................................... 76
V: The Semantic Network Retrieval Theory......................... 81
Semantic Representations of Magnitude ...................... 81
Evidence From Non-Arithmetic Numerical Tasks .......... 81
Representing Numerical Magnitude in the Semantic Network
Simulation .......................................................... 87
Cognitive Architecture ................................................. 89
Basic Structure of Semantic Network Retrieval Theory .... 89
Simulation Framework ............................................. 91
Retrieval of Arithmetic Facts from Memory .................. 94
Simulating Arithmetic Fact Retrieval .......................... 96
Interpretation of Basic Phenomena .............................. 97
Problem Size Effect ................................................... 98
Error Types and Frequencies ..................................... 101
Priming ............................................................. 107
Dyscalculia ......................................................... 108
Simulation Results and the Semantic Network Retrieval Theory .... 110
VI: Evaluation of Semantic Network Retrieval Theory ............. 112
LIST OF TABLES

Table 1: Error types and percentages for multiplication ....................... 6
Table 2: Variations in frequencies of error types across brain-damaged patients. ................................................................. 10
Table 3: Sample pelification table .......................................................... 16
Table 4: Error types in Whalen et al. and Harley experiments .................. 21
Table 5: Error types and frequencies produced by MATHNET. ................. 50
Table 6: Examples of KSR’s spoken and written responses to arithmetic problems .................................................................................. 66
Table 7: Examples of possible perseverations ............................................ 67
Table 8: Example of perseveration analysis of reordered data ..................... 68
Table 9: Example of possible perseveration at a depth of 2 ......................... 69
Tables 10 & 11: KSR’s perseveration rate relative to chance for addition, subtraction, multiplication and division .................................................. 70
Table 12: Examples of KSR’s spoken and written answers to simple arithmetic problems ................................................................................ 75
Table 13: Comparison of lexical stack positions of three, thirteen, nine, and nineteen ......................................................................... 84
Table 14: Errors in simulated and normal multiplication ............................. 102
Table 15: Operand errors separated according to operand distance .......... 103
Table 16: Table errors separated by operand distances ............................... 105
Table 17: Reported frequencies of errors sorted according to the difference between the actual operands, and the operands of the erroneous response ........................................................................... 119
Table 18: The amounts of competition problem 4 x 6 receives from each of its competitors .............................................................. 121
Table 19: The competition problem node 4 x 6 receives when 3 x 6, 5 x 6, 4 x 5 and 4 x 7 are removed ......................................................... 122
Table 20: The competition problem node 6 x 7 when 3 x 6, 5 x 6, 4 x 5 and 4 x 7 are removed ................................................................. 122
Table 21: The competition weighting scale for any problem ....................... 123
Table 22: Competition values for normal arithmetic ................................. 124
Table 23: Display of 16 problems removed from training and testing for one diamond arithmetic fact set .................................................. 124
Table 24: Competition values for reduced problem set presented in Table 23. ................................................................. 125

Table 25: Sample of diamond arithmetic answers. .............................. 126

Table 26: Proportions of different error types in the diamond arithmetic experiment ......................................................... 131
LIST OF ILLUSTRATIONS

Figure 1: Ashcraft's Table Search Model ................................................ 24
Figure 2: Ashcraft's Network Retrieval Theory....................................... 26
Figure 3: Siegler's Distribution of Associations Theory ......................... 30
Figure 4: Dehaene's phonological storage hypothesis............................. 38
Figure 5: Retrieval processes according to the phonological storage hypothesis ................................................ 55
Figure 6: Results of incorrectly encoding the problem in phonological form. 56
Figure 7: Processes involved in saying the problem aloud and writing the answer. ................................................................. 60
Figure 8: Results of incorrectly transcoding the problem ......................... 61
Figure 9: Implications of a speech production error for arithmetic fact retrieval. ................................................................. 62
Figure 10: Implications of a perseverative error for arithmetic fact retrieval. 63
Figure 11: Implications of a minor phonological conversion error .......... 64
Figure 12: According to the phonological storage hypothesis written and spoken answers are based on an .................................................. 73
Figure 13: KSR's deficit according to McCloskey's Semantic Retrieval Position ................................................................. 77
Figure 14: Example of a compressive number line with activation pattern for 4. .............................................................................. 85
Figure 15: Representative Semantic Magnitude Representations .......... 88
Figure 16: Basic organization of the semantic network retrieval theory ...... 90
Figure 17: Connection weights between the last operand nodes and problem nodes 8 x 5 and 9 x 5. ......................................................... 93
Figure 18: Basic semantic network simulation architecture.................... 93
Figure 19: Simulation iteration counts according to problem family......... 99
Figure 20: Simulation error rates by problem family ............................ 100
Figure 21: Simulation solution times before and after the removal of the problems 4 x 5, 4 x 7, 3 x 6 and 5 x 6................................. 114
Figure 22: Relation between RT and problem family............................ 130
I: INTRODUCTION

Recently there has been renewed interest in the cognitive processes involved in numerical processing. The ability to manipulate numerals and perform calculations are central to many aspects of everyday life. In addition, these abilities constitute a clearly delineated domain in which to study the larger issues of memory, cognitive representations, and cognitive processes. Therefore, understanding how we represent and manipulate numerals could have implications both for numerical processing, and more generally for theories of memory and language.

Despite significant progress in the area of numerical cognition, fundamental questions remain about how we represent numbers and perform simple calculations. Some of the current issues include: How are numbers represented internally? How do we remember and represent simple arithmetic facts like $5 + 4 = 9$? Why are larger problems such as $8 \times 9$ harder than smaller problems like $2 \times 3$? This dissertation is designed to address some of these questions.

I begin by describing some of the basic findings from studies of arithmetic fact retrieval and introduce the current theories of arithmetic fact retrieval, evaluating their ability to account for the available findings, and illustrating some of the issues in arithmetic processing which have yet to be accounted for. The dissertation research begins with a study of a brain-damaged patient which evaluates one of the primary theories of arithmetic fact retrieval. I argue that the pattern performance of patient KSR is incompatible with the notion that arithmetic facts are retrieved in a phonological form. Next I develop an alternative to the current models of arithmetic fact retrieval, and provide a computer simulation of the theory demonstrating how it accounts for the major arithmetic phenomena. Finally, I provide an evaluation of this new theory of arithmetic fact retrieval with an experiment in which normal subjects are taught an artificial arithmetic operation.
My main argument will be that a representation of numerical magnitude plays an important role in retrieving arithmetic facts from memory. This position is based on the robust evidence from other non-arithmetic tasks which suggests that humans possess an internal representation of numerical magnitude. Given that a representation of this nature influences performance on other numerical tasks, I propose that this same internal representation of magnitude also influences arithmetic fact retrieval, and propose how this might account for the major arithmetic phenomena.
II: BASIC ARITHMETIC PHENOMENA

In order to review the literature as succinctly as possible, I focus on the findings which have been most influential in shaping current theories of arithmetic fact retrieval. Three main areas of research will be encompassed in the literature review. First, I describe the error and reaction time phenomena reported in normal adult arithmetic fact retrieval. Next, I turn to studies of acquired and developmental dyscalculia and suggest how arithmetic impairments can influence theories of normal arithmetic fact retrieval. Finally, I review the findings from artificial arithmetic studies.

Normal Arithmetic Fact Retrieval

Problem Size Effect

Perhaps the most pervasive finding from simple arithmetic research that is responses to larger problems such as 9 x 8 are on average, slower and more error prone than responses to smaller problems such as 2 x 3. This problem size effect has been robustly reported in studies of both addition (e.g., Ashcraft & Battaglia, 1978; Parkman & Groen, 1971), and multiplication (Campbell, 1985; Miller et al., 1984; Harley, 1990), and has been reported across cultures (Geary, 1996). Several possible sources of the problem size effect have been proposed, including the size of the operands (Gallistel & Gelman, 1992), the frequency of problem presentation (Ashcraft, 1982), the frequency with which non-retrieval strategies are used (e.g., successive addition; Siegler and Shrager, 1984; Lefevre et al., in press), and the order in which the problems were acquired (Campbell & Clark, 1992). None of these theoretical positions have been widely accepted, in part because each of these factors are highly intercorrelated in normal arithmetic and little independent evidence supporting these positions has been acquired (see Ashcraft, 1992).

Exceptions to the Problem Size Effect
Notably, there are two sets of problems for which response times and error rates systematically depart from the typical increase in latency (and error rate) relative to problem size. First, problems with identical operands (e.g., 6 x 6), known as tie problems, are answered more quickly than other problems of similar operand and answer size. Compared with non-tie problems, tie problems also reveal a smaller increase in reaction time relative to problem size (Campbell & Graham, 1985; Miller et al., 1985).

The second set of exceptions to the problem size effect are the multiplication problems with a 0 or 1 operand (e.g., 6 x 0, 1 x 4), and the addition problems with a 0 operand (e.g., 3 + 0). Reaction times on these problems do not increase with problem size (as measured by sum of operands, for instance). Some have proposed (e.g., Ashcraft, 1982) that RTs are uncorrelated with problem size because these problems are solved using a general rule (e.g., N x 0 = 0, N + 0 = N), rather than retrieving individual arithmetic facts from memory (e.g., 2 x 0 = 0; 3 x 0 = 0). The patterns of impaired performance resulting from brain-damage have provided important evidence consistent with this position (Sokol et al., 1991).

Error Patterns

While educated adults enjoy a high rate of accuracy on simple arithmetic problems, when subjected to moderate time pressure, subjects typically make errors on approximately 5-10% of problems (e.g., Campbell & Graham, 1985). Several studies have documented that the errors made in producing answers to arithmetic problems are highly systematic and consistent across studies (e.g., Ashcraft, 1992; Harley, 1990; McCloskey et al., 1991; Miller et al., 1984). This consistency is thought to reflect the underlying structure of the arithmetic fact retrieval system, and can constrain theories of arithmetic fact retrieval. For this reason, each type of error and its frequency of occurrence is described below.

The description of errors focuses on multiplication errors because these errors can often provide more information than errors from other operations. Multiplication problems typically do not share an answer with any problem other than its complement (e.g., 8 x 9 = 72), whereas
addition problems (for example) often have many problems which share answers with one another (e.g., all problems with a sum of 9). This property of multiplication permits comparisons not only of the difference between the error and the correct answer, but also comparisons of the correct operands with the operands corresponding to the response (if the error is another multiplication answer). Operand comparisons are not generally possible in studies of addition or other operations because it can not be determined which operands may be related to a particular response.

The most common error that subjects make when producing answers to simple arithmetic problems is to produce the correct answer for another problem which shares one operand with the presented problem. For example, if the problem 7 x 6 was presented, a typical error might be to produce the answer "forty eight". Reports indicate that operand errors account for approximately 75-80% of all errors in production tasks (Campbell & Clark, 1985; Harley, 1990).

An interesting characteristic of operand errors is that the erroneous responses are usually correct for a problem that shares one operand with the presented problem and is also close in magnitude with respect to the other operand (McCloskey et al., 1991). For example, it is more likely that "forty eight" would be an error for the problem 7 x 6 (presented problem: 7 x 6, response was correct for problem: 8 x 6, difference in operands: 8 - 7 = 1) than "twelve" (presented problem: 6 x 7, response correct for problem: 6 x 2, difference in operands: 7 - 2 = 5). Most errors have been found to have an operand distance of two or less; as operand distance increases, the frequency of operand errors systematically decreases.
Table 1: Error types and percentages for multiplication (Campbell, 1985).

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operand</td>
<td>77</td>
</tr>
<tr>
<td>Table</td>
<td>12</td>
</tr>
<tr>
<td>Non-Table</td>
<td>8</td>
</tr>
<tr>
<td>Operation</td>
<td>3</td>
</tr>
</tbody>
</table>

The second most common type of error is one in which the answer is correct for another problem in that operation, but which shares no operands with the problem presented (accounting for approximately 15% of errors). For example, if the problem 6 x 7 were presented, the response "seventy two" would be considered a table error.

The third type of error reported in studies of simple arithmetic is the non-table error. Subjects occasionally produce a response which is not the correct answer for any problem in the intended or any other operation (e.g., 6 x 8 = 47). These errors occur infrequently and constitute about 7% of errors (Campbell, 1985; Harley, 1990).

Lastly, the rarest type of error is the operation error in which the erroneous response corresponds to the correct answer for those operands in a different operation (e.g., 6 x 7 = 13). This type of error is quite uncommon, encompassing approximately 3% of errors (Campbell, 1985; Harley, 1990).

Error Priming

Campbell and colleagues have reported that the likelihood of producing an error on a particular trial increases when related arithmetic problems have been recently tested. For example, the probability of producing a specific error (e.g., 24) for a related problem (e.g., 4 x 5) increases significantly if that answer was retrieved from memory in an earlier trial (e.g., 4 x
6; termed positive error priming). Error rates have been found to be influenced by roughly the ten immediately preceding trials (Campbell & Clark, 1989).

Campbell proposes that error priming is the result of residual answer activation remaining in the fact retrieval system after retrieval, which affects subsequent fact retrieval attempts. Support for this position comes from an experiment in which multiplication trials (6 x 8 = ?) were alternated with number reading trials (stimulus: 28; response: "twenty eight"; Campbell, 1991). Relative to chance, answers to previous multiplication trials were found to be 30% more likely to be errors on subsequent multiplication trials, while number reading trials produced no measurable influence on errors. This suggests that positive error priming involves arithmetic fact retrieval rather than spoken numeral production. However, Tarling (1993) found only modest (4% above chance) positive error priming effect when subjects performed a multiplication verification task (e.g., True/False: 4 x 6 = 28) which presumably also requires arithmetic fact retrieval. One possible explanation for the discrepancy between the production and verification tasks is that arithmetic verification tasks may induce more non-retrieval strategies (Zbrodoff & Logan, 1986). Despite the inconsistency between the two task types, the priming effect has been robustly reported in studies of arithmetic fact retrieval production tasks.

In contrast to positive error priming, the response to the trial immediately preceding the current trial has been found to be less likely as an error than would be expected by chance. Termed negative error priming, this effect has been reported when immediately preceding trials involved either reading a numeral aloud or producing multiplication answers. For this reason, Campbell suggests that the negative error priming effect originates in numeral production processes and does not involve arithmetic fact retrieval (Campbell & Clark, 1989; Campbell, 1991).

Dyscalculia
Acquired Dyscalculia

In addition to studies of cognitive arithmetic processes in normal subjects, researchers also study how brain damage impairs the ability to retrieve arithmetic facts from memory. The focus of this research is to determine how the normal system may be constructed such that acquired damage may result in the observed deficits.

One of the fundamental findings from acquired dyscalculia is that arithmetic fact retrieval is functionally independent of both numeral comprehension, and production processes. Several cases have revealed that impairments may be restricted to either arithmetic fact retrieval or numeral comprehension and production processes (e.g., Benson & Denckla, 1969; Sokol et al., 1991; Warrington, 1982). For example, patient DRC was found to be impaired on arithmetic tasks (e.g., addition, multiplication) despite having no significant impairments in either comprehension or production of numerals (Warrington, 1982). In contrast, other cases reveal impairments restricted to a particular comprehension or production process. For example, Benson and Denckla (1969) report two patients who were severely impaired on numerical tasks which required spoken numeral production, yet who were able to retrieve arithmetic facts from memory when the arithmetic tasks did not require spoken responses (e.g., arabic numeral, or multiple choice responses).

Arithmetic fact retrieval has also been reported to dissociate from the performance of multidigit calculation procedures such as carrying and borrowing. Several patients (e.g., PS, GE, Sokol et al., 1991) suffered impairments in retrieving arithmetic facts from memory despite retaining the ability to perform multidigit calculation procedures. In contrast other patients reveal the opposite dissociation: an impairment in performing multi-digit calculation procedures, while being unimpaired at arithmetic fact retrieval. These dissociations indicate that arithmetic fact retrieval is independent of other processes such as numeral comprehension and production processes, and multidigit calculation procedures.

Characteristics of Arithmetic Fact Retrieval Deficits
Studies of the arithmetic fact retrieval abilities of brain-damaged patients have also revealed several interesting error patterns which place important constraints on theories of normal arithmetic fact retrieval. The discussion will consider separately performance for the problems 2 x 2 to 9 x 9, and performance for the 0 x N and 1 x N problems for reasons to be described below.

For the problems 2 x 2 to 9 x 9, patients are not equally impaired on all problems, but rather exhibit considerable non-uniformity of impairment across problems. For example, patient MD was virtually unimpaired at 3 x 7, but erred on 91% of 3 x 8 problems, and on 39% of 3 x 9 problems (different patients exhibit significantly different patterns of impairment across problems). Despite the seemingly scattershot damage, there is a consistent trend for larger problems to reveal higher error rates than smaller problems (McCloskey et al., 1991).

Most patients also tend to produce the same types of errors as normal subjects do, with approximately the same relative frequencies (while the absolute error rates are typically much higher). For example, McCloskey et al. (1991) report that subjects CM, IE, MD, PS, SB most frequently produced operand errors, followed in frequency by table errors, operation errors, and non-table errors (as is found for normal subjects). However, two patients (FW and TM) appear somewhat exceptional as they produced many more non-table errors than is typically reported for normal subjects.
<table>
<thead>
<tr>
<th>Error Type</th>
<th>Approximate Percentages of Error Types</th>
<th>Patients: CM, MD, IE, SB, PS</th>
<th>Patients: FW, TM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operand</td>
<td>75</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>Table</td>
<td>15</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Non-Table</td>
<td>10-15</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>&lt;5</td>
<td>&lt;5</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Variations in frequencies of error types across brain-damaged patients.

In contrast to the highly irregular patterns of impairment for the problems 2 x 2 - 9 x 9, for the problems 0 x N and 0 x N brain-damaged patients reveal either uniformly high or uniformly low error rates. For example, patient CM was unimpaired on all 0's problems, whereas GE consistently erred on 0's problems. Almost all errors on 0's problems involve producing the answer 'N x 0 = N'. The uniformity of performance across 0's problems suggests that these problems may be solved by reference to a general rule (i.e., 0 times any number is 0). If one rule is responsible for all 0's problems then it would be expected that subjects are either able to answer all problems correctly, or are unable to answer any problems correctly, a pattern which has been repeatedly reported. While less striking, comparable results have also been found for 1 x N and N x 1 problems (McCloskey et al., 1991), suggesting they these problems are also solved by rule.

**Developmental Dyscalculia**

Temple (1991, 1995) has studied several cases of developmental dyscalculia. Interestingly, when developmental cases are studied individually, the patterns of impairment appear quite similar to those found after acquired impairment. For example, several cases (e.g., SW, HM, AW, RW) have revealed a pattern of performance very similar to that of brain-
damaged patient DRC: a clear dissociation between impairment to arithmetic fact retrieval, and sparing of numeral comprehension and production processes. For example, AW was found to be highly accurate at tests of arabic numeral comprehension, and spoken numeral production (95-100% correct), but revealed significant impairments when asked to say aloud the answer to problems written in arabic form (e.g., 3 x 4 = ?). Cases such as AW’s provide converging evidence that fact retrieval processes are functionally separate from both the procedures used for non-retrieval solutions and numeral comprehension and production processes.

In general, many of the patterns of impairment revealed in acquired dyscalculia have also been observed in developmental dyscalculias, providing additional evidence that the arithmetic impairments reported in acquired and developmental dyscalculia are not exceptional cases, but rather are likely to reflect the architecture of the normal arithmetic fact retrieval system.

**Artificial Arithmetic Operations**

In addition to studying normal and impaired performance of normal arithmetic operations, researchers have also used artificial arithmetic operations to study arithmetic fact retrieval. The advantage of using an artificial arithmetic operation is that several variables such as amount of practice, and the ability to solve problems using non-retrieval strategies, may be controlled and systematically varied.

Artificial arithmetic operations have been most useful in assessing the different possible sources of the problem size effect. Possible sources of the effect such as problem presentation frequency, order of problem acquisition, and frequency of non-retrieval strategy use, are highly intercorrelated in normal arithmetic, but may be independently manipulated in an artificial operation. For example, in order to control the effects of non-retrieval strategies, many artificial arithmetic operations have no systematicity to the answers forcing subjects to retrieve answers from memory rather than working the answers out.
Artificial arithmetic research is useful to understanding normal arithmetic to the extent to which the findings can be related to normal arithmetic. Therefore, the strongest evidence comes from studies which have the same cognitive requirements as normal arithmetic (e.g., subjects must learn several arithmetic facts, two numerals as operands in the range of 2 to 9, the answer is numerical, subjects are trained to retrieve the artificial facts as quickly and accurately as normal arithmetic facts). In this way, the similarity between the artificial and normal operations may be sufficient to draw parallels between the cognitive processes which underlie the artificial and normal arithmetic operations.

**Graham and Campbell: Alphaplication**

Perhaps the first study of arithmetic fact retrieval using an artificial operation was that of Graham (1989), reported and extended in Graham and Campbell (1992). Third and fourth grade students were taught an artificial operation termed 'alphaplication' in which two vowels were used as operands, and a randomly selected consonant was the answer (e.g., A,U=m). Using the five vowels A,E,I,O,U, Graham constructed 25 different problems which were taught to children over a period of three days.

By constructing alphaplication as a new operation, several difficulties involved in studying normal arithmetic were avoided. First, because the answer may not be reconstructed (e.g., 3 x 4 also equals 4+4+4), subjects must retrieve the facts from memory and therefore the results may be interpreted as retrieval based phenomenon. Second, factors such as the amount of training, and the order of problem acquisition may be independently controlled. In this experiment, the order of presentation (and presumably acquisition) of the problems was systematically varied.

Several phenomena reported in studies of normal arithmetic were also observed in the alphaplication study. For example, the most common error type was the operand error. Seventy-eight percent of alphaplication errors were operand errors. Further, as is typically found in normal arithmetic, the next most common type of error was the table error (20%), and
then non-table errors (1.5%). Finally, Graham and Campbell report an effect of the order of presentation of the stimuli. Problems presented first were more accurate and faster than those studied second. Graham and Campbell suggest that these similarities between normal arithmetic and alphaplication were observed because in both cases a network of inter-related facts was used to retrieve facts from memory.

However, in order to ensure that similar retrieval processes are being used in normal arithmetic and alphaplication, the characteristics of alphaplication problems and training must mimic normal arithmetic as closely as possible. In the case of alphaplication, there are some important shortcomings. First, the set of alphaplication facts involved letters of the alphabet, rather than numerals. If some effects in normal arithmetic involved the nature of the numerals themselves, then these effects would not be expected in alphaplication. Second, normal arithmetic facts can be retrieved with error rates which are less than 10%, even when responding within a time limitation of less than one second. In contrast, alphaplication error rates averaged 30% when subjects were provided with 4 seconds to produce a response. Third, normal arithmetic problems are studied and learned through extensive training, while alphaplication was learned within 3 sessions over 3 days. Finally, the number of facts learned, 25, was much smaller than the 64 facts acquired in normal arithmetic. Together, these shortcomings are significant, and although some parallels may be drawn between alphaplication and normal arithmetic, the results of the alphaplication study may not reflect the processes underlying normal arithmetic.

**Zbrodoff & Logan: Alphabet Arithmetic**

One of the most extensive uses of artificial arithmetic operations has come from Zbrodoff and her colleagues (Zbrodoff, in press; Logan & Klapp, 1991; Klapp et al., 1991; Zbrodoff & Logan, 1986). Subjects are trained on 'alphabet arithmetic' problems in which a combination of letters and numerals compose the problem, with a letter answer (e.g., $A + 2 = C$). The value of the letters are equal to their position in the alphabet. Thus subjects may
originally verify the problems by either converting the problem into its numerical equivalent (e.g., \(1 + 2 = 3\)), or reciting the appropriate number of letters of the alphabet (e.g., \([A], B, C\)). This design was primarily intended to study the transition from solving problems using an algorithm (i.e., counting up) to solving problems by retrieving the answers from memory.

Zbrodoff (in press) recently designed a series of alphabet arithmetic experiments intended to study the source of the problem size effect by manipulating problem presentation frequency, and order of acquisition. Unfortunately, there are serious shortcomings in using alphabet arithmetic to study the problem size effect. First, by providing subjects with a method of converting alphabet arithmetic problems into normal addition problems, Zbrodoff has severely limited her ability to manipulate both how the problem is being solved, and the amount of experience subjects have with a particular problem. If the problems are being converted into their numerical equivalents (e.g., \(A + 2 \rightarrow 1 + 2\)) and addition facts are being retrieved from memory, all of the prior experience with normal arithmetic will affect the subject's performance, and variables such as frequency of problem presentation, can not be independently manipulated. Second, because of the design of the problems, we cannot determine if the numerical operands play a role in the problem size effect as was intended. Each problem used one of 6 consecutive letters (e.g., \(A-F\)) as the first operand, and one of three digits (2-4) as the second operand and a letter as an answer (e.g., \(A + 2 = C\)). However, each operand letter was always paired with only one digit (e.g., \(A + 2, B + 3\)), permitting the subject to ignore the digit operand and simply memorize a letter-letter association (e.g. if \(A\) is an operand, \(C\) is the answer). Finally, because of the small number of problems learned (6), subjects may not reveal the same types of associative interference which is found in normal arithmetic (which has 64 problems). In summary, it appears that alphabet arithmetic has significant weaknesses both in its ability to independently manipulate the various sources of the problem size effect, and in its ability to relate performance to normal arithmetic.

**Harley:** *Pelification*
The design of the artificial arithmetic experiments performed by Harley (1990) appear to overcome many of the shortcomings of the alphabet arithmetic and alphaplication studies. The set of 'pelification' facts were constructed to mimic normal arithmetic problems as closely as possible. Pelification problems were constructed using arabic numerals as operands, and answers. Like normal arithmetic, there was an operation sign put between the operands (represented here as 'à'). A 10 x 10 matrix was constructed in which 0's and 1's problems could be solved by reference to a rule. 0 à N and N à 0 problems were equal to 10, and 1 à N and N à 1 problems equaled 10+N. All problems were commutative (i.e., 3 à 4 = 4 à 3).

<table>
<thead>
<tr>
<th>Operand 1</th>
<th>Operand 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9</td>
<td></td>
</tr>
<tr>
<td>0 10 10 10 10 10 10 10 10 10</td>
<td></td>
</tr>
<tr>
<td>1 10 11 12 13 14 15 16 17 18 19</td>
<td></td>
</tr>
<tr>
<td>2 10 12 67 79 86 38 41 59 57 71</td>
<td></td>
</tr>
<tr>
<td>3 10 13 79 52 26 89 58 62 94 47</td>
<td></td>
</tr>
<tr>
<td>4 10 14 86 26 87 39 83 53 29 73</td>
<td></td>
</tr>
<tr>
<td>5 10 15 38 89 39 37 91 68 76 82</td>
<td></td>
</tr>
<tr>
<td>6 10 16 41 58 83 91 98 31 97 34</td>
<td></td>
</tr>
<tr>
<td>7 10 17 59 62 53 68 31 23 93 46</td>
<td></td>
</tr>
<tr>
<td>8 10 18 57 94 29 76 97 93 43 51</td>
<td></td>
</tr>
<tr>
<td>9 10 19 71 47 73 82 34 46 51 78</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Sample pelification table (Harley, 1990)

There are two important differences between pelification and normal arithmetic. First, the answers to the problems 2 à 2 to 9 à 9 were randomly associated with the problems, and so could not be solved using a non-retrieval strategy. Second, the answers had values between 31 and 97 which were uncorrelated with the size of the operands and were not answers to real arithmetic problems. This design has two important advantages. First, the random pairing of operands and errors forces subjects to retrieve these facts from memory, no non-retrieval strategies can be used to solve the problems. Second, by uncorrelating problem and answer
size, the effects of operand size and answer size allows the effects of each of these variables to be evaluated separately.

Subjects were trained on pelification facts row by row, beginning with the 0's problems. When a subject could correctly answer all problems in the row being learned, the subject was then tested on the whole set of facts learned thus far, and if able to solve all the problems, proceeded to learn the next row of facts. In the final phase of the experiment, subjects were tested on speeded production of all pelification facts several times.

By specifically uncorrelating operand size with answer size, Harley was able to separate out two possible sources of the problem size effect: the size of the operands (e.g., Gallistel and Gelman, 1992), and the size of the answers (e.g., Campbell & Clark, 1992). Harley was specifically interested in three different aspects of performance. First, are there differences in performance between rule based and fact problems? Second, how similar are the types and rates of errors in pelification to those found in normal arithmetic? Third, is there an effect of either answer magnitude or operand magnitude on RT and error rates?

Results from final testing revealed several interesting phenomena. First, subjects were virtually always correct on the rule based problems (99.6% correct), but were only able to correctly respond to 83% of fact based problems (2 \( \div \) 2 to 9 \( \div \) 9). Second, the types and frequencies of errors followed the pattern observed in normal arithmetic: the most common errors during learning were operand errors (57%), followed in frequency by table errors (22%), non-table errors (14%), and operation errors (4%). These results are quite comparable to normal arithmetic, suggesting that many of the same cognitive processes may be involved in both normal arithmetic and pelification.

Third, the operand size was found to influence RTs and error rates. Problems with larger operands had higher error rates and longer reaction times than problems with smaller operands (independent of the size of the answers). This is consistent with the notion that a representation of operand magnitude was involved in arithmetic fact retrieval. To further
explore this possibility, Harley examined the operand errors found in the study to see if there was a relation between the magnitude of the correct operands, and the magnitude of the operands corresponding to the erroneous response. Harley reported that errors tended to have operands which were closer in magnitude to the correct operands than might be expected by chance, again consistent with the notion that operand magnitude plays a role in arithmetic fact retrieval. In contrast, no effect was found of answer magnitude on RTs and error rates. Further analysis of the relation between the magnitude of erroneous answers and correct answers revealed that the errors were no closer in magnitude to the correct problem than might be expected by chance.

However, other factors were correlated with operand magnitude which may have contributed to fact that RT and error rates related to operand magnitude. These include the frequency of problem presentation and order of acquisition. In order to determine if these factors may have been involved in the operand size effect, Harley performed another pelification experiment designed to avoid these correlations.

The problems $2 \times 2$ to $9 \times 9$ were divided into 4 groups (with approximately equal overall problem size among groups) so that one group of problems could be learned, and then the next group and so on. In order to determine the effect of problem frequency, the problems within each group were divided in half and assigned to either a high presentation frequency, or low presentation frequency group. Problems in the high frequency group were presented approximately three times as often as the problems in the low frequency group.

Harley reports a significant effect of frequency of problem presentation, and no reliable effect of order of acquisition. Harley also replicated other experimental findings: reaction times and error rates were found to relate to the size of the operands (e.g., the sum of the operands), but were not related to the size of the answers.

Harley believed that there were several important findings from this set of experiments. First, he suggested that both the magnitude of the operands and the frequency of problem
presentation contributed to the problem size effect in pelification, and that these factors may also play a role in normal arithmetic. Second, Harley noted that in both experiments that errors tended to be close in terms of the operands of the problem and error, rather than the magnitude of the answer and erroneous response. This suggests that in pelification, subjects were retrieving the answers based on a representation of the magnitude of the operands. Finally, other potential problem size correlates such as the order of problem acquisition, and answer size were not found to influence reaction times or error rates.

However, these experiments do have some apparent weaknesses. First, subjects in Harley’s first experiment had an overall error rate which was 10% higher than is found in normal arithmetic, and average RTs which were approximately 400 ms longer than is usually observed in normal arithmetic. This suggests the facts were not learned as well as normal arithmetic facts, and therefore pelification fact retrieval performance may not be as representative as possible of normal arithmetic fact retrieval. Second, these experiments introduced several different manipulations simultaneously, including varying problem presentation frequency, order of acquisition, and rule based problems combined with retrieval based problems. One possibility is that some of the interactions between these manipulations may have influenced the findings reported, such as the problem size effect relative to the size of the operands. In a replication of the Harley study, Whalen, Lindemann, Goldschlager & McCloskey (in preparation) designed an experiment in which many of the manipulations present in the Harley experiments were simplified in order to better examine specific performance variables, such as the magnitude of the problem operands, the frequency of problem presentation.

**Whalen, Lindemann, Goldschlager & McCloskey: Pelification**

Whalen, Lindemann, Goldschlager and McCloskey (in preparation) adopted Harley’s general design, with a few key modifications. First, rather than having a fixed amount of practice on the problems, Whalen et al. required subjects to reach a specific performance
criterion before entering the final test phase (100% correct; average RT less than 1100 ms).
This ensured that each subject's performance approximated the response accuracy and latency
found in normal arithmetic. Second, the pelification table was constructed more simply. Rather
than designing the table around problem families (the details of which were not described
above), the answers and problems were paired by allowing a random pairing of problems and
answers such that there was (1) no correlation between the size of the answers and problem
operands, (2) no differences between tie and non-tie answer characteristics, and (3) no
systematic non-retrieval solutions. Pelification problems were also limited to the \(2 \times 2 - 9 \times 9\)
range, eliminating rule based associations between problems and answers.

Whalen et al.'s first experiment was designed to determine if Harley's main findings
(i.e., error pattern, problem size effect relative to operand size) might be replicated if the various
learning characteristics such as problem presentation frequency, and order of acquisition were
held constant across problems. Eight subjects were trained over several sessions (range: 12-25)
on their own individual set of 64 pelification facts. Each training session presented all 64
pelification facts, and each fact was presented equally frequently.

Results reveal a replication of several of Harley's (1990) findings. Perhaps most
strikingly, despite the fact that answer size and operand size were uncorrelated, performance
was both faster and more accurate for problems with smaller operands than those with larger
operands. This effect was observed using several different measures of operand size including:
(1) dividing the problems into those with small and large operands, and (2) problem family
(computing an average RT for 2's problems, 3's problems, and so on), and (3) the sum of
operands. In contrast, final reaction times and error rates during training did not correlate with
the size of the answers.

The error types and frequencies were consistent with both the findings from the Harley
experiments and those reported in normal arithmetic. First, operand errors were by far the most
common type of error, followed by table errors, non-table errors, and operation errors.
Second, an operand distance effect was also found in this experiment, replicating Harley's finding. In this case, the magnitude of the operands corresponding to the response were found to be closer in magnitude to the actual operands than expected by chance, even when the answers were very different (e.g., stimulus: \(2 \div 3\); response: 98; even if \(2 \div 3 = 31\) and \(2 \div 3 = 98\)).

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Whalen et al.</th>
<th>Harley</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operand</td>
<td>68</td>
<td>57</td>
</tr>
<tr>
<td>Table</td>
<td>17</td>
<td>22</td>
</tr>
<tr>
<td>Non-Table</td>
<td>12</td>
<td>14</td>
</tr>
<tr>
<td>Operation</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 4: Error types in Whalen et al. and Harley experiments.

Whalen et al. performed an additional experiment to evaluate the effect of varying the amount of practice subjects receive on different problems. In this experiment half of the problems were assigned to a high frequency group and the remainder were assigned to the low frequency group. There was no systematic relation between the problems chosen to be high frequency and correlates of problem size, such as operand size, and answer size. Problems assigned to the high frequency group were presented and trained three times as often as those assigned to the low frequency group. All other characteristics of the stimuli and design remained the same as the previous experiment. As might be expected, responses to problems which received more training were both faster and more accurate than responses for problems which received less training. Whalen et al. also report a replication of several important findings from previous pelification experiments, including a problem size effect relative to the problem operands, comparable error types and frequencies, and an operand size effect.

**Summary of Artificial Arithmetic Findings**
The pelification studies produced four main findings which are relevant to theories of arithmetic fact retrieval. First, the problem size effect was found to correlate with the size of the operands: as operand size increased so did the latency of responses, and the frequency of errors. This effect, which was replicated over 4 separate experiments, indicates that a representation of the magnitude of the operand plays a role in arithmetic fact retrieval. Second, the types and relative frequencies of errors closely mimicked the pattern observed for normal arithmetic. Operand errors were most frequent, followed by table errors, non-table errors and operation errors. The similarity in performance between these artificial operations and normal arithmetic suggests that both operations may have utilized similar underlying cognitive processes. Third, an operand distance effect was reported in the pelification experiments relative to operand size. If an operand error occurred, the response's operands were typically close in magnitude to the correct operands. This again suggests that operand magnitudes play a role in arithmetic fact retrieval. Finally, the frequency with which problems were presented and trained was found to correlate with latencies and error rates. Problems trained more frequently had lower error rates and faster response times, suggesting that practice strengthens the problem-answer associations.
III: THEORIES OF ARITHMETIC FACT RETRIEVAL

Despite the subjective simplicity of remembering that two plus two is four, comprehensive theories of arithmetic fact retrieval must account for several diverse findings. However, most theories of arithmetic fact retrieval have focused on specific aspects of fact retrieval rather than providing a theory capable of handling all major phenomena. Some theories account for only the problem size effect and errors (e.g., Ashcraft, 1982; Siegler, 1987), while others focus on the form in which the arithmetic facts may be stored in memory (Dehaene, 1992; McCloskey, 1992). Through this discussion each type of arithmetic theory will be discussed, and the relative strengths and weakness of each theory will be considered.

Theories Accounting for the Problem Size Effect and Error Patterns

Table Search Theories

Early theories of arithmetic fact retrieval suggested that arithmetic facts are retrieved from a metaphorical lookup table much like a times table used by children to learn the elementary arithmetic facts. In such a model, activation must spread across the table and traverse a certain distance in order to retrieve the appropriate arithmetic fact from memory (e.g., Ashcraft, 1978; Widaman, 1989). This class of theories was designed to account for the problem size effect, assuming that the spread of activation from node to node takes a certain amount of time. The larger the problem, the greater the distance that activation must spread across to reach the answer, and the longer it will take to retrieve an arithmetic fact from memory.

Ashcraft’s Table Search Theory

Ashcraft & Battaglia (1978) proposed that arithmetic facts are organized into the equivalent of a two-dimensional table. The table is entered at the row and column corresponding to the first and second operands, and activation spreads inward. The answer
which receives the most activation is the one retrieved from memory. Because only the correct answer receives activation from both the row and column sites, that node should receive the most activation and be retrieved.

Figure 1: Ashcraft's Table Search Model

Accounting for the problem size effect was the primary motivation behind the development of this theory. According this table search theory, time is required for activation to spread along the row and column of the problem table. Therefore larger problems will require more time for activation to spread to the correct answer than will smaller problems. This theory, however has significant difficulty in accounting for the systematic departures from the problem size effect, such as the tie problems.
Perhaps the most significant shortcoming of the table search model is its inability to account for the effects of brain damage. If it is assumed that an inability to retrieve a particular fact results from the destruction or weakening of the associative links between nodes in the table, then presumably no activation may travel beyond the damaged site. For example, if connections to the 2 x 3 node were impaired, little or no activation would pass to the other 2 x N and N x 3 problems beyond this node. However, brain-damage often results in impairments to smaller problems (e.g., 4 x 5) yet spares larger problems which require activation to spread beyond this node (e.g., 4 x 9). More generally, table search models suffer from the inability to account for the error types and priming effects reported for normal and brain-damaged patients. For this reason we turn to theories of arithmetic fact retrieval which are based on a non-tabular network of stored arithmetic facts.

**Ashcraft's Network Retrieval Theory**

Ashcraft's network retrieval model (Ashcraft, 1982, 1987) represents one of the first explicit theories of arithmetic fact retrieval to propose that simple arithmetic problems are solved by retrieving the answer from a network of stored facts. According to the network retrieval theory, the arithmetic fact retrieval system includes a set of nodes representing each operand in the problem, and a set of nodes representing each answer. Each problem has a separate answer node, even if problems share the same answer (e.g., 8 x 3, and 4 x 6 have separate answer nodes). Activation spreads from operand nodes to answer nodes, and from answer nodes to other specific answer nodes. Each operand node is associated with its corresponding answer nodes. For example, the 4 node for the first operand is linked to all the 4 x N answer nodes (such as 20, 24, 28, etc.). In a similar fashion, all of the operand nodes for the second problem operand are associated with their corresponding answer nodes. In addition, answer nodes which have operands in common with other answer nodes (e.g., 20 and 25) are hypothesized to be connected to one another. The strength of the connection between answer nodes depends on
the 'operand distance' between answers (i.e., the answer node 20 (4 x 5) will activate 24 (4 x 6) more strongly than 36 (4 x 9)).

Figure 2: Ashcraft's Network Retrieval Theory

According to Ashcraft's theory, fact retrieval involves the spreading of activation from the operand nodes to answer nodes and from answer nodes to other answer nodes. For example, when the problem 4 x 6 is presented, activation spreads from the 4 operand node to all the 4 x N answer nodes and from the 6 operand node all the N x 6 answer nodes. Only the 24 (4 x 6) answer node will receive direct activation from both the 4 and 6 nodes. Activation from this answer node will in turn activate to some degree the answer nodes for problems which share an operand and have a second operand close in magnitude to the presented operand (e.g., the answer nodes 20, and 28).

Ashcraft proposes that the strength of the associations between the problems and answers are primarily determined by the amount of experience a subject has with that problem.
The more a problem is presented, the stronger the association between the problem and answer. He suggests that the problem size effect is primarily the result of frequency differences between small and large problems. Smaller problems are hypothesized to be more frequently presented than larger problems, leading to stronger associations between problems and answers for smaller problems. Support for this notion comes from counts of problem frequencies in textbooks (e.g., Ashcraft & Christy, in press). Smaller problems typically appear more frequently than larger problems in primary school and college texts. However, it is worth noting that there are numerous opportunities in which simple arithmetic facts may be experienced outside of the classroom, and counts of problems in textbooks may not be an accurate reflection of actual experience with these problems over a lifetime.

As the network retrieval theory is currently developed, it appears able to account for some, but not all of the different types of phenomena reported in the literature. For example, the artificial arithmetic experiments (e.g., Whalen et al., 1996) suggest that while problem frequency plays a role in solution RTs and error rates, other factors such as the magnitude of the operands also appear to play a role.

This theory also has trouble accounting for the different types of errors. Ashcraft postulates that on each trial, the correct answer should receive more activation than other answers, yet that under some circumstances the 'pandemonium like' answer selection procedure (a mechanism subsequently added to account for errors) may result in the selection of another highly activated answer (Ashcraft, 1987). The most highly activated answers next to the correct answer will be those which share an operand with the correct problem (which therefore have full activation from one operand node), and have a second operand which is close in magnitude to the presented problem’s second operand (allowing activation to spread from the correct answer node to this answer node). This formulation correctly predicts that the most frequent error type will be the operand error, and that these errors will tend to have operands which are close in magnitude to the correct operands, resulting in an operand distance effect.
Unfortunately, these predictions are largely the result of the post-hoc postulation of specific connections between the correct answer and certain answer nodes. Ashcraft assumes that activation will spread from the correct answer node to other answer nodes which share an operand with the correct answer, and have a second operand which is close in magnitude to the problem's second operand (stimulus: 8 x 7; activates 56 answer node, which activates 48, 64, 49, and 63). While these assumptions lead to the correct predictions, there seems to be little theoretical motivation for such connections. Why these particular associations should exist and not connections between all answers is not clear, nor is there evidence directly supporting this assumption.

The network retrieval theory also has some difficulty accounting for non-table errors, and table errors. The arithmetic fact retrieval system is only composed of the problems and their correct answers and therefore when a specific fact is retrieved, there is no specific mechanism within the Ashcraft theory capable of producing a non-table answer. Table errors pose a theoretical challenge. In order to account for table errors (e.g., 5 x 6 = 28) one must assume that: (1) activation from each of the operands activates their corresponding answer nodes (e.g., 5 x N answers and N x 6 answers); (2) through spreading activation from specific answer nodes (e.g., 5 x 7, 4 x 6) activation is transferred to other nodes which share an operand (e.g., 5 x 7 and 4 x 7, 4 x 6 and 4 x 7); (3) even though these nodes would only receive 'secondary' spreading of activation, one of their answers is nevertheless retrieved. This process both appears unlikely to occur with sufficient frequency, and relies on the existence of the answer node connections, whose motivation was questioned above.

Recently, Ashcraft (1992) has proposed an extension of his theory which incorporates two aspects of other models in order to avoid some of the shortcomings of earlier versions. Specifically, Ashcraft notes that his theoretical position is unable at present to account either for the manner in which children acquire arithmetic facts, or provide a motivated mechanism under which errors are likely to occur, or non-retrieval strategies may be employed. To account for
non-retrieval solutions of arithmetic problems, Ashcraft has largely adopted Siegler's position (presented in detail below) that simple arithmetic problems may be solved either by retrieving the correct arithmetic fact from memory, or by using a non-retrieval strategy. Ashcraft postulates that both retrieval, and non-retrieval strategies begin in parallel, and the first solution reached is the one used. To account for error generation, Ashcraft has also adopted the notion that individual problems may be associated with more than one answer. For example, the problem 8 x 9 might be most strongly associated with 72, but also have associations with 81, 63, 64, and so on. This extension of the network retrieval theory provides a mechanism whereby errors may be generated. By adding the notion that several answers may be activated by the problem presented, there is an additional mechanism which might be likely to generate errors.

Unfortunately, Ashcraft’s current formulation has not been described with sufficient detail to determine how the different aspects of the theory will interact. The theory in its present formulation does not produce tightly constrained predictions about performance. Therefore, a further detailed description will be required to determine the success of the most recent theoretical modifications.

**Siegler's Distribution of Associations Theory**

Siegler's research has focused primarily on the acquisition of the basic arithmetic facts. However, his theory does provide predictions about both childhood and adult solution times, and error patterns. During early schooling children show overt signs both of counting up during addition, and performing repeated addition to solve multiplication problems. Siegler suggests that each time a problem is solved using one of these strategies (or arithmetic fact retrieval), a link between the response generated and the presented problem is strengthened (eventually strengthening the association enough to allow answers to be consistently retrieved from memory). For example, if the problem 2 + 3 is solved by counting up (e.g., 3, 4, 5 ->
answer = 5), the association between the problem node '2 + 3' and the answer node 5 is strengthened.

However, solution strategies do not always lead to a correct solution. In particular, Siegler considers two types of errors which commonly occur when using the repeated addition strategy. First, when solving a multiplication problem (e.g., 3 x 4) using repeated addition, one might add too few (4+4=8), or too many times (4 + 4 + 4 + 4 = 16). Second, an error could be made when adding (e.g., 4 + 4 + 4 = 11). These errors will result in several answers becoming associated with a problem. The larger the problem, the greater the number of multiple additions, and therefore the greater the likelihood that an error will be committed, and multiple answers will be associated with the problem.

![Diagram of Siegler's Distribution of Associations Theory](image)

Siegler also proposes that a criterion is used to determine if the answer will be retrieved from memory, or solved using a non-retrieval strategy. Each time a problem is presented, a confidence criterion is set (which determines how certain the subject must be to state a retrieved answer), and a search length (which determines how many attempts will be made to retrieve an answer before turning to a non-retrieval strategy to solve the problem). When a problem is presented, the problem node corresponding to that problem is activated. Activation then spreads to the various answers which are associated with that problem. The likelihood of selecting each of the answers is proportional to the strength of the connection between the
problem and answer relative to the other answers. For example, if the answer 10 were associated with the problem 7 + 3 with a strength value of .4, and the total strength of all connections summed to 1, then the probability of selecting that answer would be 40%. If the activation of the answer exceeds a preset confidence criterion, then the answer generated by retrieval process is used. Otherwise, the same retrieval process is repeated, or if the system determines if the maximum search time has been reached, non-retrieval strategies are employed.

The distribution of associations theory accounts for several common findings in arithmetic, such as the problem size effect. Because non-retrieval strategies are more error prone for larger problems, the strength of the association between the problem and correct answer will be weaker for larger problems than that expected for smaller problems. Larger problems would therefore require more retrieval attempts and more frequent usage of non-retrieval strategies, which together will result in higher error rates and longer reaction times for larger problems.

This theory suggests that when errors occur in arithmetic, they are either the result of making an error in a non-retrieval strategy, or retrieving one of the incorrect answers associated with a problem (an association formed due to errors in non-retrieval strategies). Therefore, the frequency and type of errors in normal arithmetic should correspond to the errors reported in non-retrieval strategies. For example, the most common error committed when using the multiple addition strategy (i.e., 4 x 6 = 6 + 6 + 6 +6) is to add too many multiples (4 x 6 = 28), or too few (4 x 6 = 18). Given that errors will tend to have a number of repeated additions which is close to correct, this is also consistent with the operand distance effect. It is much more likely that the repeated addition strategy would result in a number of repeated additions which is close to correct. Non-table errors may be accounted for by suggesting they are the result of erroneous additions during multiple addition. For normal multiplication, numbers close to the correct answer are typically not answers to other multiplication problems, and errors which are close to the correct amount will typically be classified as non-table errors.
However, this theory appears unable to account for two other types of errors: operation errors and table errors. Operation errors simply are not accounted for in this theory. It would not be expected that the problem representation (e.g., 7 x 3) would become associated with the answer (10) due to errors in multiple additions, or some other non-retrieval strategy. While it is possible that children might use erroneously use an addition strategy to solve a multiplication problem when learning the facts, it is unlikely that this error would be frequent enough to still play a role in adult performance.

Table errors are another type of error which this theory cannot currently account for. It is not clear how errors in retrieval strategies might result in an answer which is related to different problem operands (e.g., 8 x 5 = 54). One possibility is that table errors arise from errors in addition in non-retrieval strategies (e.g., 3 x 5 -> 5 + 5 + 5 -> 16). However, as mentioned earlier, errors which are close to the correct answer tend to be non-table errors (e.g., problem: 8x5; close non-table errors: 37,38,39,41,43; close table error: 42). Therefore this theory predicts (contrary to the reported findings) that there should be more non-table errors than table errors.

The distribution of associations theory also has difficulty accounting for error priming phenomena. Presumably, positive error priming is due to the influence of residual activation of an answer after retrieval on subsequent arithmetic fact retrieval attempts. However, as currently formulated, each problem has a unique set of problem-answer associations, and associative strengths are determined solely by past responses to that problem and are unaffected by responses to other problems. For example, retrieval of the answer to 4 x 6 would affect the activation level of 24 for the problem 4 x 6 (because that is one of 4 x 6's associated answers), but this would not affect the association between 4 x 7 and its representation of 24. Because each problem has unique associations to answers which do not interact, there is no mechanisms that would allow responding to one problem to affect performance of a different problem.
Finally, the artificial arithmetic studies, particularly those by Harley (1990) and Whalen et al. (1996) pose a considerable challenge to Siegler's theory. These studies did not allow subjects to use non-retrieval strategies to solve arithmetic problems, but nevertheless report (a) a problem size effect relative to the size of the operands, and (b) the types and frequencies of errors normally reported in arithmetic. Because these artificial arithmetic operations were set up so that non-retrieval strategies were unavailable, the RT and error phenomena must be attributed to retrieval based phenomena. This suggests that normal arithmetic performance may also be attributed to retrieval based processes, which are not necessarily related to performance of non-retrieval strategies during acquisition of the arithmetic facts.

**Non-Retrieval Strategies During Adult Arithmetic Performance**

While the Siegler theory appears to have some significant shortcomings, it raises an important possibility: adults may sometimes use non-retrieval strategies to solve arithmetic problems. It has generally been assumed that virtually all arithmetic problems are solved by retrieving the answer from memory (e.g., Ashcraft, 1992; McCloskey, 1992; Siegler, 1988). However, this assumption has recently been challenged (Baroody, 1995; Lefevre et al., in press; Siegler & Shipley, 1995). For example, Lefevre and colleagues suggest that while most problems are solved by arithmetic fact retrieval, not all are retrieved from memory. Lefevre proposes that problems may sometimes be solved using non-retrieval strategies (e.g., solving $6 \times 7$ by retrieving $6 \times 6$ and adding 6).

Others such as Baroody (1995) suggest that no arithmetic facts are stored in memory. Instead, he proposes that the systematicity of arithmetic operations such as multiplication allow answers to be quickly recomputed each time an answer is required (a detailed proposal suggesting how this might occur has not yet been proposed). In light of these challenges to the prevailing view that adults virtually always retrieve arithmetic facts from memory, the relevant evidence will be examined.

*Reaction Time Results*
Several researchers have studied both children and adults as they solve simple arithmetic problems (e.g., Ashcraft et al., 1982; Campbell, 1985; Miller et al., 1984; Siegler). As discussed above, solution latencies for larger problems (e.g., 7 x 8) are generally longer than those for smaller problems (4 x 3). However, these studies have also revealed that children's performance differs dramatically from adult performance in terms of the magnitude of the problem size effect. Groen & Parkman (1972) found that children's reaction times for addition problems may be predicted using a MIN model, in which reaction times linearly increase by approximately 400 ms for each increase in value of the smaller operand, suggesting a counting-on-from-the-largest-operand strategy (e.g., problem: 2 + 8; solution: "eight" "nine" "ten"). However, for adults, the step function was found to be very small (on the order of 20 ms per step). Because the 20 ms step is much faster than the speed with which silent counting may be performed, a counting model could not account for adult performance and was rejected in favor of the notion that most problems were solved using direct retrieval of the arithmetic facts from memory (Groen & Parkman, 1972). While this evidence suggests that not all arithmetic problems with strategies (i.e., Baroody, 1995), this does not preclude the possibility that arithmetic problems are sometimes solved using a non-retrieval strategy. For this reason I turn to studies of self-reported strategy use during arithmetic problem solving.

Self-Reports of Strategy Use

Three studies have been performed to determine the frequency with which subjects retrieve arithmetic facts from memory. Each study required subjects to solve simple arithmetic problems (e.g., 4 x 3 = ___), and immediately report how they solved the problem (e.g., retrieved the answer from memory; retrieved another fact and modified the answer: e.g., 7 x 6 = 7 x 7 - 7; etc.). These studies relied on self reports as an accurate report of the cognitive processes which were performed to solve the problem. For example, Geary and Wiley (1991) studied how college age adults and elderly adults solve simple addition problems. They found that elderly adults virtually always report that they retrieved the fact from memory (reporting
non-retrieval strategies for only 1% of trials), while college age adults reported using non-retrieval strategies for approximately 10% of trials. However, the instructions to subjects emphasized accuracy rather than speed, thereby potentially biasing the subjects towards using a non-retrieval strategy. Although exact details are unavailable, in a similar experiment by Healy et al. (1993) subjects reported that approximately 16% of problems were solved using non-retrieval strategies. Finally, Lefevre et al. (in press), found that for simple multiplication problems subjects report non-retrieval strategies for 10-15% of trials. However, Lefevre also reports that individual variation was considerable, ranging from 0% reported strategy use to as high as 75% strategy use. In summary, subjects reported on average that they solved between 1% and 15% of problems using non-retrieval strategies.

While the results of these studies are intriguing, the validity of the self reports has not been strongly supported by converging evidence from either latencies or error patterns. For example, Lefevre has reported that average RTs for trials reported to be solved by non-retrieval strategies are typically slower than those in which subjects reported that the retrieved the answer. This finding is consistent with either the notion that subjects were accurately reporting their solution method, or that subjects believed that non-retrieval strategies take longer to execute and generally reported retrieval when responses were fastest. Thus it is not yet clear how accurate the self reports may be. Unlike studies of children's arithmetic solution strategies, adults do not provide overt signs of the strategy they are using.

Studies of artificial arithmetic operations may help to determine the frequency of adult non-retrieval strategy use. While artificial arithmetic operations such as pelification do not allow for non-retrieval strategies, they do reveal error and RT patterns which are very similar to those found in normal arithmetic. This similarity suggests that most arithmetic errors are likely retrieval based, rather than non-retrieval strategy based errors.

However, until further evidence is gathered, the frequency of non-retrieval strategies will be a issue which is largely unresolved. While it is likely that adults occasionally use non-
retrieval strategies, the degree to which these strategies may contribute to the patterns observed in normal arithmetic is currently uncertain. Thus, an important future consideration is to determine the extent to which individuals might use non-retrieval strategies, and how the use of these strategies influences arithmetic performance.

**In What Form Are Arithmetic Facts Stored?**

While the theories described thus far focus on the reaction time and error rate patterns observed in arithmetic, the next three theoretical perspectives examine a different aspect of arithmetic performance: the form in which arithmetic facts are stored in memory. The most significant distinction between the theories described below is the notion that arithmetic facts are either stored in one particular form (Dehaene, 1992; McCloskey, 1992), or that arithmetic facts are stored in many different forms (Campbell, 1992).

**Dehaene's Phonological Storage Hypothesis**

Dehaene has proposed a triple-code theory which suggests that all numerical processes are performed using one of three types of numerical codes: an analog magnitude representation, an auditory-verbal code, or a visual code. He proposes that arithmetic facts are stored and retrieved exclusively in an auditory-verbal form. In order to retrieve an arithmetic fact from memory, a problem presented in arabic numerals (e.g., 4 x 6) must be converted into an auditory-verbal code (see Figure 4 below). Once converted into an auditory-verbal code, the arithmetic problem (e.g., /four times six/) can be used to retrieve the appropriate arithmetic fact from memory (e.g., /four times six is twenty four/). The arithmetic fact gives rise to the answer in an auditory-verbal form (/twenty four/). In order to write the answer, it must first be converted from its auditory-verbal form into an arabic form (24).
The notion that arithmetic facts are stored in a phonological form (coined the *phonological storage hypothesis*) has been motivated by several observations which have not yet been strongly supported experimentally. Perhaps the strongest evidence in support of the phonological storage hypothesis comes from the observation is that multilingual individuals appear to perform calculations in the language they spoke when learning arithmetic (e.g., Kolers, 1968; Marsh & Maki, 1976; Shanon, 1984). For example, even if fluent in English, a native French speaker who learned the arithmetic facts in French might perform calculations fastest in French (e.g., "trois et cinq"). This phenomenon has been taken to suggest that arithmetic facts are stored phonologically in the language of acquisition, and that the phonological representation of the problem must be generated in the language in which the facts were learned in order to retrieve the stored fact (e.g., Kolers, 1968; but see Noel & Seron, 1992).
Other casual observations appear potentially consistent with the phonological stored hypothesis. For example, children learning arithmetic facts typically engage in substantial amounts of overt and covert oral rehearsal, perhaps leading to the creation of phonological representations of the facts. In addition, both children and adults often report that they say simple arithmetic problems to themselves (aloud or silently) when trying to remember the answers. This introspection could suggest that a phonological problem representation must be computed in order to retrieve the arithmetic fact from memory.

Recent evidence from bilingual arithmetic appears potentially consistent with Dehaene's position. For example, Frenck-Mestre and Vaid (1993) report that when bilingual subjects verify problems in their second language (e.g., "deux et trois font huit"), subjects respond more slowly than if the problem is presented in their first language. It has also been reported that during verification tasks, incorrect answers which are correct for another operation (e.g., 2 + 3 = 6) produced less interference when presented in their non-preferred language than when presented in their preferred language. While these findings are consistent with the notion that problems in the secondary language had to be converted into the subjects primary language, it could also be the case that problems presented in either the subject's first and second language must both be converted into another representation (e.g., a semantic representation not tied to any particular form of presentation). If the first language conversion process were more efficient than that of the second language, the same pattern of results would be expected. At present then, despite intriguing introspections consistent with the phonological storage hypothesis, there is little evidence which uniquely supports this theory. This phonological storage hypothesis therefore remains a viable candidate theory that needs to be further evaluated and perhaps developed to account for RT and error phenomena.

McCloskey's Abstract Retrieval Position

McCloskey and colleagues have not proposed a specific theory of arithmetic fact retrieval, but they have proposed a general cognitive architecture for numerical processing
which makes assumptions about the form in which arithmetic facts are retrieved and answers are produced. McCloskey et al. (McCloskey, 1992; McCloskey et al., 1985) propose that numerical processing is composed of three main components: numeral comprehension, calculation and numeral production. According to this theory, all stimuli regardless of the manner in which they were presented, are converted to an abstract semantic code in order to perform calculations. Arithmetic facts are retrieved using an abstract representation of the problem operands and operation, and the arithmetic fact is retrieved in an abstract semantic form, which is subsequently converted into an appropriate form for output (e.g., spoken numerals).

Some evidence consistent with the notion that arithmetic facts are retrieved using an abstract operand representation comes from the operand distance effect. These errors are related to the original problem in terms of the magnitude of their operands. The operand distance effect might be explained by postulating that in addition to the correct fact, other facts are activated which have semantically related operands (related in terms of magnitude as opposed to phonologically related operands, or those with similar visual features).

A second possible source of support for the semantic theory comes from the problem size effect (related to the magnitude of the problems operands) found in the artificial arithmetic operations. RTs and error rates varied according to the magnitude of the operands and not other features such as their visual shape or phonological form. This finding might be accommodated within McCloskey's theory by suggesting that the arithmetic facts are retrieved using a semantic magnitude representation of the problem operands.

However, until McCloskey and colleagues further developed their theoretical perspective, it cannot generate clear predictions regarding the primary arithmetic fact retrieval phenomena. Thus like the phonological storage hypothesis, McCloskey's abstract retrieval position remains a viable mechanism for arithmetic fact retrieval which has yet to be fully developed.
Campbell's Multi-Format Position

Campbell's theories of arithmetic fact retrieval (Campbell, 1985; Campbell and Clark, 1992; Campbell, 1995) have undergone an evolution from their inception, resulting not in an explicit and testable theory, but rather in a general theoretical position: humans possess the ability to store numerical information in many different ways including but not limited to visual, semantic, auditory, and written forms. According to this position all numerical cognition (including arithmetic fact retrieval) involves a complex interaction of many different format specific codes. Campbell (1995) suggests that arithmetic facts are represented in every form in which arithmetic expressions may be described (e.g., visually, auditorily, written words, imaginary number lines, colors, finger counts, etc.). Unfortunately, Campbell has not detailed how the various codes may interact and his position provides no specific predictions about arithmetic performance.

Recently, Campbell has used the positive error priming paradigm to determine if there are format specific arithmetic effects which would not be expected by the single-format theories of Dehaene and McCloskey. For example, Campbell (1994) reports stronger error priming effects when the primed and priming problems are presented in the same format than if they are presented in different formats. For example, if a previous trial involved the presentation of the arabic problem $4 \times 7$, likelihood of the error 28 in response to the arabic problem $4 \times 8$ is greater than the likelihood of the same error in response to the written problem $four \times eight$. Campbell has suggested that this effect constitutes evidence that different presentation formats will invoke different interactions between the various format specific codes. He suggests that if arithmetic facts stored in a single form, there should be no difference between the priming effects of same and different format primes since in both cases (e.g., $4 \times 8$ or $four \times eight$) the same arithmetic facts are being retrieved.

Clearly this finding is potentially troublesome for theories of arithmetic fact retrieval which postulate a single underlying representation of the arithmetic facts. However, in this
experiment problems were presented in alternating formats (arabic, written words, arabic, and so on). One possibility is that residual activation within each input system is playing a role in retrieval performance. It might be the case that only one type of input is provided to the fact retrieval system, and that other types of input are inhibited during this time. For example, there may be residual activation of the <6> and <4> nodes in the arabic input system, but this input is inhibited during word trials, and therefore only plays a role in arabic trials. Thus, an effect which is specific to the numeral comprehension mechanisms could result in such a 'format specific' effect, even if there were only a single underlying form of arithmetic facts.

As more data become available, perhaps multi-format proposals may be preferred. However, Campbell's perspective will only be adopted if able to generate firm predictions and be evaluated empirically.

**Performance Across Presentation and Answer Formats After Brain Damage**

While no fully developed format-specific theory of arithmetic fact retrieval has been proposed, some studies of brain-damaged patients have attempted to explore this possibility. For example, if it is assumed that problems presented in one format (e.g., arabic digits) preferentially access arithmetic facts in the same form of specific code, then arithmetic fact retrieval performance after acquired brain injury might vary according to the specific presentation and response formats used in that particular task.

Sokol et al. (1991) studied the performance of their subject PS to determine if there were any format specific effects. Patient PS's performance was found to be highly consistent across presentation and response formats. When presented with arabic, written verbal, or dots stimuli, PS's error rates were 12%, 13%, 13% respectively. When responding to arithmetic problems by producing responses in arabic numerals, written verbal numerals, or dots error rates were 13%, 14%, 12% respectively. Sokol et al. computed a pairwise correlation of PS' error rates across each stimulus and response format for each problem. Correlations were quite high, ranging from .83 to .90, indicating that PS tended to err on the same problems in each
stimulus and response format. While this result is not necessarily unreconcilable with some format-specific hypotheses, this result is highly consistent with the notion that arithmetic facts are stored in a single form.

Retraining Specific Arithmetic Facts After Acquired Impairment

Some researchers have sought to determine how retraining arithmetic facts after brain-damage might influence other non-trained arithmetic facts. The primary question addressed was: If problems are retrained in one specific presentation and response form, how will that affect performance when the problem is presented or answered in a novel form? Presumably performance should be the same across presentation and response forms if the facts are stored in a single underlying form (assuming no training benefit to comprehension and production processes)

One study reports that when problems were trained in a specific format (e.g., problems presented in arabic numerals and requiring a keypad response), responses were more accurate when the problem was tested in the trained format (i.e. arabic numeral responses) than for other formats, (e.g., when spoke responses were required; Kashiwaga et al., 1982). Unfortunately the patients (reported by group, not individually) did appear to have some spoken numeral production impairments. Therefore, it is possible that performance was less accurate when answers were spoken because of a spoken numeral production impairment, not because of greater difficulty with verbal format fact representations. In summary, there appears to be little data which at present strongly supports the general multi-format position supported by Campbell. Nevertheless this remains a possibility which deserves continued attention as arithmetic theories are further developed.

Simulations of Arithmetic Fact Retrieval

In addition to theoretical inquiries into the RT and error patterns in arithmetic, there have also been several network simulations of arithmetic which attempt to simulate arithmetic
performance based on their theoretical perspective. These simulations are important to consider because they can provide evaluations of specific representational claims relevant to theories of arithmetic.

**Anderson, Viscuso & Spoer (1989)**

Anderson, Viscuso, and Spoer (1989) published one of the first simulations of arithmetic fact retrieval. Their formulation was based on Anderson's (1983) Brain-State-in-a-Box algorithm. In this simulation, all units have weighted connected with one another as well as to themselves. Units may take on values from +1 to -1.

The Viscuso et al. simulation consisted of 3 groups of units representing the first problem operand, the second problem operand, and the answer, respectively. Each group of units provided a distributed representation of each operand and the answer. The units for the first and second operands were set to the activation pattern corresponding to the operands in the problem and the answer units were left to settle on an answer. The representation of the operands and answer was composed of two parts: a representation of the magnitude of the numeral, and a representation of the written form of the numeral (e.g., one). The magnitude representations were constructed so that numbers with similar magnitudes had similar representations (e.g., [++++----------] for 0, [-------++++++++] for 1, [----++++++++] for 2, and so on). Name representations were based on an 8 unit per letter pattern, derived from the letter's ASCII code.

Arithmetic fact retrieval involved a series of multiple cycles in which the units propagate activation to one another relative to their activation levels. Network size limitations constrained the network to approximate arithmetic facts, rather than exact arithmetic facts. Answers were rounded to the nearest 10, or if under 10 to the nearest 5. For example, the answer to 8 x 7 is 60, 8 x 8 is 60, 4 x 2 is 10, 3 x 3 is 10, and 6 x 1 is 5.

Unfortunately, the Viscuso simulation has some serious shortcomings as a simulation of arithmetic fact retrieval. Perhaps most importantly, the approximate answer representations are
unlikely to illicit the types of inter-problem competition that are normally present in arithmetic. Small differences in numerical magnitude appear to be influential in arithmetic fact retrieval, and this simulation fails to represent those differences (e.g., normal arithmetic: \(3 \times 4 = 12, 2 \times 5 = 10\); approximate arithmetic: \(3 \times 4 = 10, 2 \times 5 = 10\)). Another weakness in the design of the simulation is the incorporation of arithmetic facts representations which are stored in written numeral codes. While arithmetic facts could be stored with reference to the arabic form of the numeral, or perhaps the phonological form of the problem, it is unlikely that arithmetic facts are stored with reference to the written form of the problem. Only rarely are arithmetic problems encountered in a written form (e.g., nine times six is fifty four).

The simulation also fails to capture important aspects of normal arithmetic fact retrieval performance. For example, normal subjects make occasional errors on otherwise well learned facts. In contrast, the Brain-State-in-a-Box algorithm is deterministic, so it will either provide a correct answer, or repeatedly produce the same error for a particular problem. Further, overall performance was found to be poor. Even though the simulation used approximate arithmetic facts, the simulation was only able to solve 2-9’s problems with an overall accuracy of 42%, much below the normal accuracy rates of about 90-95% correct. In short, the BSB simulation appears to have several significant weaknesses which make it of little value in shaping theories of arithmetic fact retrieval.

**Campbell's Network Interference Simulation**

Campbell has recently produced a network interference simulation (1996) which attempts to simulate his general position that arithmetic facts are represented in several different format specific codes. He suggests that the arithmetic fact retrieval system is composed of a format specific representation of the operands, problem nodes representing each individual arithmetic fact (presumably also in a format specific form), and an answer representation for each individual fact, composed of a magnitude representation, and a format specific
representation of the answer. The current simulation represents arithmetic facts stored in only one form (arabic numeral input, with spoken output).

When a problem is presented (e.g., 3 x 5), the problem operands and the operation are represented in a format specific form (e.g., [3 5][x]; currently few details exist about the exact nature of format specific representations). The format specific representation activates all problem nodes which have one or more of the following characteristics: (a) an operand in common with the presented problem (e.g., the [3 6][x] problem node); (b) an answer with a tens value equal to the first operand, or a ones value equal to the second operand (e.g., stimulus: [3 5][x]; activates problem nodes [6 5][x] (because answer is 30, and [5 5][x] because answer is 25); (c) problem nodes with answers which are close in magnitude to the correct answer (based on their logarithmic difference using a variation of the Welford equation). In addition to the factors presented above which affect fact retrieval, the amount of activation each problem node receives is based on how recently the problem node was previously retrieved. Problem nodes which were recently retrieved receive more activation than other problem nodes.

During retrieval, problem nodes inhibit other problem nodes in order to produce a winning problem node (however 5’s problems and ties problems receive less inhibition than other nodes). Problem nodes with an activation level which is at least 95% of the most highly activated node activate the format specific form of their answer. Answer retrieval occurs when one problem node reaches the specified activation level, and at that time the activation levels for each format specific answer representation (e.g., "twenty", "thirty", "six", "eleven") from each problem node are summed, and the most highly activated tens and ones answers (unless a teens answer exceeds these) are selected as the answer.

This simulation appears in some ways too complex, and in others insufficiently specified. For example, the simulation has so many different influences on arithmetic fact retrieval performance it is very difficult (if possible) to determine which aspects of the simulation, might be contributing to produce a certain pattern of performance.
On the other hand, some aspects of the simulation either lack strong theoretical motivation, or remain underspecified. For example, while the pattern of performance is somewhat different for ties and 5's problems, within Campbell's framework it is not clear why these problems should retrieve special treatment. Similarly, the motivation for having the most highly activated nodes provide answer activation (as opposed to all nodes) is also uncertain. Finally, and perhaps most importantly, Campbell's simulation remains unspecified in important ways pertaining to his general viewpoint that arithmetic facts are stored in many different format specific codes, which interact to produce the normal pattern of performance. First, Campbell provides only one format specific code in his simulation. If the interactions between codes are relevant, then it is not clear how a simulation without other codes should perform. Second, Campbell does not provided a detailed description about the attributes of the format specific codes. For example, does the arabic code representation of 3 partially activate the representation of 8 due to their similar visual characteristics (whereas four and five interact in spoken codes due to their phonological similarities)? Simply put, the Campbell simulation does not provide a clear opportunity to evaluate his multi-format position, due to the simulation's complexities and under specifications.

**McCloskey and Lindemann: MATHNET**

McCloskey and Lindemann (1992) devised a network simulation which attempts to model several aspects of arithmetic fact retrieval, including occasional error generation (as opposed to deterministic answer production), normal arithmetic error types and frequencies, and patterns of performance of brain-damaged subjects.

The arithmetic fact retrieval simulation is composed of problem units, hidden units, and answer units. The problem units are composed of two fields representing the problem operands, and a third field representing the arithmetic operation. The answer units are composed of 2 fields representing the tens and ones answer quantities. The operand and answer quantities are represented by a distributed representation of quantity in which similar
quantities have overlapping representations. For example, 7 is represented by [ - - - - - - + + + - - -], 8 is represented by [ - - - - - - - - + + + - + -], and 9 is represented by [ - - - - - - - - - + + + + + + + ], where - equals -1 and + equals +1. The MATHNET simulation has several connections between units, all of which are bidirectional and symmetrical. All problem units are connected to all hidden units, and all hidden units are connected to all answer units. In addition, all answer units are connected to one another.

The main goal of the MATHNET simulation was to produce a simulation in which arithmetic fact retrieval is inherently stochastic. McCloskey and Lindemann attempt to do so by simulating arithmetic fact retrieval using the Mean Field Theory paradigm (Peterson & Anderson, 1987). This paradigm allows activation levels for individual units to be initially highly variable, but over time settle to a more stable state. The process of annealing (settling the network) allows for some variability in performance.

Weights between connections were initially set to small random values and the network was trained on the arithmetic facts by problem family (e.g., 2's problems, 3's problems and so on). While a new set of facts were being learned (e.g., 4's problems), the old facts were also retrained half as often as the new facts. Training involved comparing the activation patterns when only operand nodes were held constant with the activation patterns while both problem and answer units were held constant, and modifying the weights in the free state to better match the correct pattern. Three separate simulations were performed with identical parameters and training to evaluate the consistency of the results.

McCloskey and Lindemann report several findings of interest. To begin with, the MATHNET simulation provided accuracy levels which were comparable to normal arithmetic. Under a speeded settling procedure (or annealing schedule), the simulations were 97% correct over 30 test runs, and each problem was at least 65% correct, with at most 3 problems with accuracies under 85% correct. The types and frequencies of errors were generally comparable to those found in normal arithmetic (as presented in Table 5), although there were more non-
table errors than typically reported, and there were no operation errors (since only one operation was simulated).

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operand</td>
<td>79</td>
</tr>
<tr>
<td>Table</td>
<td>8</td>
</tr>
<tr>
<td>Operation</td>
<td>0</td>
</tr>
<tr>
<td>Non-Table</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 5: Error types and frequencies produced by MATHNET.

Of the operand errors, 91% were found to have an 'operand distance' of ±1 (e.g., 7 x 8 = 63), a finding which is comparable to the operand distance effect in normal arithmetic. Presumably this effect relates to the nature of the operand representations, which provided all adjacent numerals (e.g., 4, 5) with some similar features.

McCloskey and Lindemann also report a problem size effect. Error rates were found to be generally higher for large problems than smaller problems, and larger problems required more iterations of settling to adopt a stable state than smaller problems. There are several possible sources of this effect, including the order of problem acquisition, and frequency of problem presentation (smaller problems were both presented first, and trained more frequently). Identical network simulations were trained with independent manipulations of these factors to determine the source(s) of the problem size effect. Problem frequency manipulations were the only manipulation found to produce a problem size effect.

McCloskey and Lindemann also studied the effects of lesioning the network and compared the results to studies of brain-damaged patients. All connection weights were randomly lowered by a factor randomly chosen from a distribution with an average of .40 and a standard distribution of .10. This damage resulted in an average error rate across simulations of
approximately 20%. Several findings reported in studies of brain-damaged patients appear to be reproduced using this lesioning technique. For example, the simulations revealed non-uniform impairment across problems. Error rates for some problems were as high as 100% while other problems were unaffected by the reduction in connection weights.

Perhaps equally interesting is the finding that the proportions of different types of errors closely resembled the patterns reported in studies of acquired impairments. Most lesioned simulations revealed the predominate pattern of error frequencies reported in cases of brain-damage: most errors were operand errors, with much lower proportions of table and non-table errors. However, a few damaged simulations also revealed the pattern of performance of patients FW and TM, significantly more non-table errors than reported in other cases. For example, lesioned network A revealed 47% operand errors, 19% table errors, and 34% non-table errors. Interestingly, when the same intact simulations were equally damaged on different occasions, they revealed different patterns of impairment, suggesting that both error patterns may be the result of the same types of damage, rather than impairments of unique subsystems.

In summary there appear to be several relevant findings from this study. First, network retrieval theories appear to be capable of producing some of the major arithmetic phenomena, such as the errors patterns reported in normal arithmetic, and the patterns of impairment after brain damage. Second, operand errors and the operand distance effect may be related to similarities between the representations of numerically similar numbers. The numeral representations in this simulation, which included only quantity representations (and did not include either visual or phonological features of the numerals) did appear capable of eliciting operand errors, and an operand distance effect. However only problem frequency was found to affect retrieval accuracies and latencies. In sum, this simulation provided some support for the notion that arithmetic facts might be retrieved using a representation of numerical magnitude within a network of stored arithmetic facts.

Unresolved Issues in Simple Arithmetic
What are the Sources of the Problem Size Effect?

Despite a great deal of research focused on revealing the source or sources of the problem size effect, no single position has prevailed. Several different underlying sources have been considered including: the frequency of problem presentation (Ashcraft, 1987; Ashcraft and Christy, 1995), and the frequency of non-retrieval strategy use (Siegler and Shrager, 1984), the order in which the arithmetic facts were acquired (Campbell, 1985), and the size of the answers (Campbell, 1996). The recent evident from artificial arithmetic reveals two possible sources of the problem size effect: problem presentation frequency and operand magnitude. No current theory appears fully capable of accounting for these two sources of the problem size effect.

What are the Sources of Errors in Arithmetic?

While theories have attempted to account for the various types of errors made in arithmetic production tasks, no single theory has successfully dealt with all of the main findings. For example, Siegler's Distribution of Associations theory provides a clear account of operand errors, the operand distance effect, and non-table errors, but is unable to adequately account for table errors, operation errors, or the findings from the artificial arithmetic operations. Further, the artificial arithmetic suggest that errors may be retrieval based without reference to non-retrieval strategy use.

Ashcraft's Network Retrieval Theory can account for operand errors and the operand distance effect, but can not account for table errors, operation errors, or non-table errors. Perhaps more importantly, the Ashcraft theory assumption that specific answers are linked to one another remains largely unmotivated. In summary, these theories appear to fall short of handling all of the errors generated in normal arithmetic.

How Often are Non-Retrieval Strategies Used?

Another unresolved issue in the study of normal arithmetic is the frequency of non-retrieval strategy use. Beginning early grade school subjects increasingly rely on arithmetic fact
retrieval to solve simple single-digit problems (e.g., 6 x 7). However, it is not clear to what extent adults use non-retrieval strategies to solve arithmetic problems (e.g., solving 8 x 9 by retrieving 8 x 8 and adding 8). The frequency of non-retrieval strategy use is an important consideration because current arithmetic fact retrieval theories assume that: (1) virtually all problems are solved using arithmetic fact retrieval; and (2) reaction time and error phenomena in speeded arithmetic may be attributed to arithmetic fact retrieval. Determining the frequency of adult non-retrieval strategies use, and the influence of non-retrieval strategy use on overall reaction time and error rates, are crucial issues for theories of arithmetic fact retrieval.

How Are Arithmetic Facts Represented in Memory?

There are currently two explicit hypotheses about the form in which arithmetic facts are stored. One possibility is that arithmetic facts are stored in a phonological form. Some limited support for this position comes from introspective reports of how children, adults, and multi-lingual adults solve arithmetic.

A second possibility is that arithmetic facts are stored in an abstract semantic form. There appears to be somewhat stronger evidence consistent with this semantic fact representation position (e.g., operand distance effect and problem size effect relative to the magnitude of the operands in artificial arithmetic studies).

In this dissertation, I propose that the error and RT phenomena may be best accounted for by a theory of arithmetic fact retrieval based on the assumption that arithmetic facts are retrieved using a semantic representation (Chapter V: The Semantic Network Retrieval Theory). In order to substantiate this claim it is important to first evaluate the possibility that arithmetic facts may be stored in a phonological form. The next chapter provides an evaluation of the phonological storage hypothesis using a detailed study of a brain-damaged patient.
IV: EVALUATION OF THE PHONOLOGICAL STORAGE HYPOTHESIS

The first experimental investigation introduced in this dissertation is intended to evaluate the hypothesis that arithmetic facts are stored and retrieved in a phonological form (Dehaene, 1992; Kolers, 1968; Shannon, 1984). According to this phonological storage hypothesis, regardless of the format in which a problem is presented (e.g., arabic numerals; $4 \times 6$) it must be converted into a phonological form (e.g., /four times six/) in order to retrieve the arithmetic fact from memory (/four times six is twenty four/).

![Diagram of retrieval processes according to the phonological storage hypothesis]

Figure 5: Retrieval processes according to the phonological storage hypothesis

The phonological storage hypothesis makes the prediction that the correct arithmetic fact should only be retrieved from memory if the problem is correctly converted into its phonological representation. If an error is made in translating the problem into a phonological form, then the answer retrieved from memory should reflect this error.
As illustrated in Figure 6, if the problem 4 x 6 were mistakenly translated into the phonological representation /three times six/, then the arithmetic fact /three times six is eighteen/ should be retrieved from memory, resulting in the written answer 18.

![Diagram](image)

**Figure 6:** Results of incorrectly encoding the problem in phonological form.

This chapter reports the pattern of performance of brain-damaged patient KSR. KSR is able to retrieve arithmetic facts from memory despite being unable to correctly name aloud the arabic operands of the problem. For example, when presented the problem 7 x 4, KSR said the problem aloud as "nine times four", yet wrote the correct answer to the presented problem, 28. I will argue that this pattern of performance is incompatible with the phonological storage hypothesis because: (a) KSR can retrieve arithmetic facts from memory (as opposed to working the answers out); and (b) many of KSR's spoken errors may be attributed to impairment in phonological conversion of the problem, and not other processes such as spoken numeral production. If KSR's errors in naming the problem operands reflect impairment in converting
problem into its corresponding phonological form, then the phonological storage hypothesis predicts that KSR should not be able to retrieve the correct arithmetic facts from memory.

Case History

KSR, a 44 year old male, was a Ph.D. candidate in Chemical Engineering before suffering a CVA in 1994. KSR’s general language abilities were assessed with the Boston Diagnostic Aphasia Exam (Goodglass & Kaplan, 1983). To briefly summarize the tests of written and oral language, KSR's ability to comprehend and produce spoken language was severely impaired as a result of his CVA, while written language processes were much less impaired.

Numerical Comprehension and Production

Described below are several tests of KSR's ability to comprehend and produce numerals in different forms. Critical to the evaluation of the phonological storage hypothesis described below are KSR's abilities to comprehend and produce arabic numerals, and produce spoken numerals. KSR was given several transcoding tasks in which he was asked to translate a numeral from one form (e.g., 6) to another (e.g., six). Each transcoding task was composed of ten numerals from 10 to 99, and ten numerals from 100 to 99,999.

Tests of arabic numeral comprehension and production revealed no significant impairments. KSR made no errors either when writing the arabic form of written word numerals (20/20 correct), or when writing the written word form of arabic numerals (20/20), except for occasional spelling errors such as 90 -> ninty). When presented with two arabic numerals KSR was also able to correctly judge which of two numbers was the larger without error (20/20).

Pencil and paper arithmetic tasks also require arabic numeral comprehension and production. If performed accurately they can provide evidence that arabic numeral comprehension and production processes are intact. In a paper and pencil test of addition,
subtraction and multiplication using single and multidigit numerals presented in arabic form, KSR solved all problems quickly and accurately (60/60). Together these results suggest that KSR has no significant impairments either in comprehension or production of arabic numerals.

In contrast, KSR does appear to have a significant spoken numeral production impairment. On transcoding tasks constructed like those reported above, KSR was only able to correctly say aloud 3 of 20 arabic numerals, and 1 of 20 written numerals. Errors predominately involved producing the correct spoken form for another numeral (e.g., 13- >"fourteen", seven hundred one -> "six hundred one").

Spoken errors cannot be attributed to the loss of specific phonological representations, or an inability to produce certain numerals. Over the course of extensive testing, KSR was able to correctly say aloud each of the individual numeral names at least once. The fact that KSR's errors are whole numeral substitutions suggests that he is unable to convert arabic and written numerals into their appropriate phonological form for spoken production.

**Arithmetic Fact Retrieval**

KSR performed timed single digit arithmetic in which problems were presented in arabic form, and responses were typed on a numerical keypad. Collapsing over two runs of the 390 simple arithmetic problems from all four operations, KSR answered 92% of problems correctly (719/780). His average reaction times for addition, subtraction, multiplication and division were 1046, 1121, 1167, 1352 ms respectively (normal arithmetic RTs typically range from 800-1300 ms). KSR's RTs are significantly faster than expected if he was solving problems using a non-retrieval strategy such as multiple additions (adding seven 8s to solve 7 x 8) which would be expected to take several seconds. In summary, KSR's speed and accuracy suggest that KSR is able to retrieve arithmetic facts from memory.

**Evaluation of the Phonological Storage Hypothesis**
This section evaluates the phonological storage hypothesis' primary prediction: that a problem must be correctly converted into its corresponding phonological representation in order to retrieve the appropriate arithmetic fact from memory. To test this prediction, on each trial KSR was asked to say aloud the problem operands and write the answer in arabic digits. According to the phonological storage hypothesis arithmetic facts are associations between the output phonology of the problem (e.g., /four times six/) and the output phonology of the answer (e.g., /twenty four/). Because saying aloud the problem and arithmetic fact retrieval both require the problem to be converted into the same phonological representation (e.g., /four times six/), it is reasonable to assume that the output of the phonological conversion process is used both in arithmetic fact retrieval and spoken production of the problem. Saying aloud the problem can therefore be used as an indirect measure of the phonological form used in arithmetic fact retrieval. Written answers were used as a measure of KSR's ability to retrieve the appropriate arithmetic fact from memory.

![Diagram](Figure 7: Processes involved in saying the problem aloud and writing the answer.)
If KSR was able to say aloud the problem operands correctly, then it was assumed that
the problem was correctly converted into its phonological form. According to the phonological
storage hypothesis, a correct phonological representation of the problem should allow KSR to
retrieve the appropriate arithmetic fact from memory. However, if KSR was unable to say
aloud the problem operands correctly, this could reflect an error in converting the problem into
its phonological form.

According to the phonological storage hypothesis, if the phonological form of the
problem is incorrect, KSR should not retrieve the correct arithmetic fact from memory. The
arithmetic fact retrieved should reflect the error in the encoding of the problem operands. For
example, as depicted in Figure 8, if the problem 4 x 6 is incorrectly represented phonologically
as /three times six/, then the answer /three times six is eighteen/ should be retrieved from
memory.

Figure 8: Results of incorrectly transcoding the problem.

56
However, KSR’s spoken responses are only an indirect measure of his ability to convert the problems into phonological form. Therefore it is possible that some spoken errors may not be the result of errors in phonological conversion. For example, after correctly converting a problem into its phonological representation (e.g., 7 x 9 -> /seven times nine/) an error might be made in speech production such that one portion of the problem is incorrect (e.g., "seven times nime"; see Figure 9).

In this case, the phonological storage hypothesis would predict that the correct arithmetic fact should still be retrieved because the problem was correctly represented in phonological form (/seven times nine/) and can be used to retrieve the appropriate arithmetic fact from memory.

Alternatively, even if KSR said the incorrect number word (e.g., stimulus: 4 x 7; response: "four times nine") he may nevertheless have correctly encoded the problem in phonological form (/four times seven/). Some brain-damaged patients exhibit a pattern in which
they sometimes repeat previous responses (perseverations), or produce upcoming responses (anticipations; e.g., say answer to problem when attempting to name the operands). If the error in naming the problem was perseverative or anticipatory in nature, the spoken response might be unrepresentative of the underlying phonological representation of the problem. The mechanisms underlying perseverative errors are not well understood, therefore it is conceivable that such an error might occur after the generation of the (correct) phonological problem representation as depicted in Figure 10. If an utterance was based on a perseveration or anticipation, then the phonological storage hypothesis might still predict that the correct arithmetic fact should be retrieved provided the appropriate phonological form of the problem was submitted to the phonological fact retrieval system.

Finally, even if there is an error in the phonological conversion of the problem, the correct arithmetic fact might still be retrieved in some special circumstances. For example, if there was a minor error in the conversion of the problem into phonological form (e.g., 5 x 6 ->

Figure 10: Implications of a perseverative error for arithmetic fact retrieval.
/frive times six/), it might be possible to retrieve the correct arithmetic fact from memory even if the phonological representation is not exactly correct. Therefore, to be conservative, it will be assumed that when the phonological form is close to correct, the correct arithmetic fact may still be retrieved.

![Diagram](image.png)

Figure 11: Implications of a minor phonological conversion error.

In summary, the phonological storage hypothesis does not predict that all errors in naming aloud the problem operands should result in an inability to retrieve the correct arithmetic fact from memory. Rather, only major phonological conversion process errors (e.g., 2 x 3 -> /eight times three/) are certain to impair arithmetic fact retrieval. For this reason, we will remove from the analysis all trials in which the spoken error could conceivably allow the correct arithmetic fact to be retrieved, such as minor phonological errors (e.g., 5 -> "frive"), and trials in which it is determined that the subject might have either perseverated from a previous response or anticipated an upcoming response. Once these trials are removed, the remaining
spoken responses should reflect KSR’s ability to convert the problem into its corresponding phonological form.

**Method**

All 390 simple arithmetic table problems (e.g., $3 + 4$, $14 - 9$, $5 \times 4$, $36 / 6$) were presented in arabic form. KSR was asked to say aloud the problem and write the answer in arabic form. Several slightly different versions of this task were performed, each of which resulted in comparable performance and will therefore be reported together. The first version of this task required KSR to say the problem and then write the answer. In the second version KSR wrote the answer, then said the problem aloud. In the third version KSR said the problem aloud, said the answer aloud, and then wrote the answer. Problems were presented until KSR signaled he was ready to respond (typically 1-2 s) at which time the problem was removed in order to reduce the likelihood that he might re-encode the problem after making one response.

**Scoring Procedure**

Only errors in saying aloud the problem operands (not the operation) were considered. In order to provide the phonological storage hypothesis with the greatest possible benefit of the doubt, any trial with a spoken error which could conceivably be unrepresentative of the phonological representation of the problem, was removed from the analysis.

**Results**

Over the three tasks using all 390 arithmetic problems, KSR was able to produce the correct written answer for 98% of trials (1147/1170). However, he was only able to say aloud the problem operands correctly for 12% of trials (141/1170). Examples of errors are provided in Table 6.
Table 6: Examples of KSR's spoken and written responses to arithmetic problems.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Spoken Problem</th>
<th>Written Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 + 5</td>
<td>&quot;two plus four&quot;</td>
<td>13</td>
</tr>
<tr>
<td>7 - 4</td>
<td>&quot;sixteen minus four&quot;</td>
<td>3</td>
</tr>
<tr>
<td>1 x 7</td>
<td>&quot;six times nine&quot;</td>
<td>7</td>
</tr>
<tr>
<td>14 / 2</td>
<td>&quot;fifteen divided by five&quot;</td>
<td>7</td>
</tr>
</tbody>
</table>

To ensure that spoken errors in the analysis could not be attributed to mechanisms other than conversion of the problem into phonological form, all errors which could possibly be considered simple phonemic errors, anticipations or perseverations were removed from the analysis.

Eleven trials were removed from the analysis because of an error which was considered phonemic in nature and possibly originating from speech production (e.g., 4 -> "door"). An additional 6 trials were removed from the analysis because of possible anticipatory errors (e.g., stimulus: 2 + 3; spoken response: "two plus five").

*Perseveration Analysis*

If KSR was perseverating it would be expected that some errors would correspond to previous responses or stimuli\(^1\). However, even if KSR's errors were not perseverative, they would still occasionally correspond to previous responses simply by chance. In order to determine if KSR was sometimes perseverating on previous responses, it must be determined how often his spoken errors should correspond to previous stimuli and responses by chance. If the number of correspondences between his errors and previous trials exceeds the chance value,

---

\(^1\)The perseveration analysis sought to determine if an error corresponded with one of several different parts of previous trials including: the actual problem operands, the spoken problem operands, the correct answer, and the written answer.
it would indicate that some errors are perseverative. Below is a description of how chance levels of previous trial/spoken error correspondences were calculated.

To calculate the frequency of chance correspondences between KSR's spoken errors and previous responses, the data was randomly reordered several times and reexamined for spoken errors/previous trial correspondences. The number of chance correspondences between previous trials and spoken errors should not vary between the original order, and the random ordered data sets, since these errors simply occur by chance. However, for perseverative errors the order of the trials is crucial. A perseverative error is based on a previous stimuli or response. Relative to the original order of the trials, random reordering of the trials would dramatically reduce the number of error/previous trial correspondences if KSR's errors were sometimes perseverative. An example of the analysis to be performed is given in Tables 7 and 8. There are 4 error/previous trial correspondences in the original data (Table 7) but only 2 in the randomly reordered data (Table 8). The difference between analyses suggests that there may have been some perseverative responses in the original data set.

<table>
<thead>
<tr>
<th>Trial</th>
<th>Stimulus</th>
<th>Spoken Problem</th>
<th>Written Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4 + 3</td>
<td>&quot;nine plus three&quot;</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>2 + 4</td>
<td>&quot;two %= plus four&quot;</td>
<td>6 @</td>
</tr>
<tr>
<td>3</td>
<td>5 + 4</td>
<td>&quot;two %= plus six @&quot;</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>2 + 1</td>
<td>&quot;two $ plus two $&quot;</td>
<td>3 #</td>
</tr>
<tr>
<td>5</td>
<td>9 + 6</td>
<td>&quot;nine plus three $&quot;</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 7: Examples of possible perseverations  
(superscript symbols denote correspondences between errors and previous responses)

The original data were reordered and reanalyzed 1000 times to provide a measure of the likelihood that errors might correspond to a previous response by chance. If KSR was sometimes perseverating, the data in its original order should reveal more error/previous
response correspondences than the randomly reordered trial data. If KSR's original data has no more error/previous response correspondences than the randomly reordered trial data, then we can conclude that KSR's errors were rarely if ever perseverative.

<table>
<thead>
<tr>
<th>Trial</th>
<th>Stimulus</th>
<th>Spoken Problem</th>
<th>Written Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5 + 4</td>
<td>&quot;two plus six&quot;</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>9 + 6</td>
<td>&quot;nine &amp; plus three&quot;</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>2 + 1</td>
<td>&quot;two @ plus two @ &quot;</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>4 + 3</td>
<td>&quot;nine &amp; plus three&quot;</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>2 + 4</td>
<td>&quot;two plus four&quot;</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 8: Example of perseveration analysis of reordered data.

There is also one more level of complexity to the analysis. In addition to determining whether or not KSR's responses may have been perseverative, we may also ask how many previous trials were influencing his responses. For example, KSR may typically perseverate relative to the immediately preceding trial (trial N-1), or perhaps the last two preceding trials (trials N-1 and N-2). For this reason different levels of perseveration "depth" were considered separately. If KSR was found to perseverate at a depth of 1 (trial N-1), then we considered the level beyond that (i.e. trial N-2), and so on.

<table>
<thead>
<tr>
<th>Trial</th>
<th>Stimulus</th>
<th>Spoken Problem</th>
<th>Written Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>N-2</td>
<td>3 x 5</td>
<td>&quot;three times six&quot;</td>
<td>15</td>
</tr>
<tr>
<td>N-1</td>
<td>9 x 2</td>
<td>&quot;nine times two&quot;</td>
<td>18</td>
</tr>
<tr>
<td>N</td>
<td>4 x 7</td>
<td>&quot;four times six&quot;</td>
<td>28</td>
</tr>
</tbody>
</table>

Table 9: Example of possible perseveration at a depth of 2.
When it appears that KSR's responses correspond to even a slightly greater proportion of previous responses up to a certain depth (e.g., the two immediately preceding trials, N-1 and N-2), then all trials will be removed from the analysis in which an error corresponds to a stimulus or response in either of the previous two (N-1, N-2) trials. Note that even though many conceivably perseverative errors are unlikely to have arisen from perseverative origins, all of these potential perseverations are being removed from the analysis. This procedure ensures that even possibly perseverative responses do not play a role in the evaluation of the phonological storage hypothesis, providing the hypothesis with the greatest possible benefit of the doubt.

The perseveration analysis indicated that the correspondence between KSR’s actual errors and previous responses was slightly higher than the randomized correspondences for the immediately two preceding trials. As shown in Tables 10 and 11, the number of correspondences between previous trials and spoken errors is slightly higher than chance for perseveration depths of 1 and 2. For example, in addition 18 errors corresponded to stimuli or responses from immediately preceding trials (a depth of 1), whereas the 1000 random repairs of the data set produced an average of 13.2 error/previous trial correspondences. This suggests that about 5 of every 20 errors which corresponds to a response or stimulus from an immediately preceding trial may be a perseveration².

²The actual data appears to have fewer correspondences between errors and trials at longer perseveration depths (3 to 5). This effect can be attributed to the fact that many more errors in the actual data were scored as smaller perseveration depths (1 and 2), leaving fewer responses to be associated with the trials with larger perseveration depths.
This analysis suggests that KSR was sometimes perseverating on the 2 immediately preceding trials. For this reason, all trials in which an erroneous response corresponded to a response, stimulus, or answer from either of the two immediately preceding trials were removed from the analysis. This resulted in the removal of 674 trials, providing a total of 478 trials for final analysis. Even though only a modest proportion of the removed trials are likely

<table>
<thead>
<tr>
<th>Depth</th>
<th>Addition</th>
<th>Subtraction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actual</td>
<td>Reordered</td>
</tr>
<tr>
<td>1</td>
<td>18.0</td>
<td>13.2</td>
</tr>
<tr>
<td>2</td>
<td>5.0</td>
<td>6.1</td>
</tr>
<tr>
<td>3</td>
<td>1.7</td>
<td>2.8</td>
</tr>
<tr>
<td>4</td>
<td>1.3</td>
<td>1.5</td>
</tr>
<tr>
<td>5</td>
<td>0.3</td>
<td>0.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Multiplication</th>
<th>Division</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth</td>
<td>Actual</td>
</tr>
<tr>
<td>1</td>
<td>17.0</td>
</tr>
<tr>
<td>2</td>
<td>8.0</td>
</tr>
<tr>
<td>3</td>
<td>1.7</td>
</tr>
<tr>
<td>4</td>
<td>2.3</td>
</tr>
<tr>
<td>5</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Tables 10 & 11: KSR's perseveration rate relative to chance for addition, subtraction, multiplication and division.
to be perseverations, all of these trials were removed to provide the phonological storage hypothesis with the greatest possible benefit of the doubt.

**Final Results**

The trials remaining after the numerous removals provide a reasonable measure of KSR's ability to convert the problem into phonological form. The fundamental question for these trials is: Was KSR able to correctly answer arithmetic problems when the problem was *incorrectly* converted into phonological form? The answer is clearly yes. KSR was found to produce the correct written answer for 100% (132/132) of the trials in which the problem was correctly converted into phonological form, and for 98% (342/346) of trials in which the problem was not correctly converted into phonological form. Clearly, the prediction of the phonological storage hypothesis has been strongly contradicted. According to this hypothesis there should be no trials in which KSR is able to retrieve the arithmetic fact from memory when the problem was not correctly converted into phonological form. KSR's performance indicates that arithmetic facts are not stored exclusively in a phonological form.

**Additional Evaluation of Phonological Conversion Ability**

Critical to the conclusions drawn above is the notion that spoken responses accurately indicate that KSR is often unable to convert the stimulus problem into a phonological form. An additional task was used to determine if KSR might somehow have correctly converted numerals into an appropriate internal phonological form which may not have been available for spoken production. One way to evaluate the intactness of the phonological form without requiring spoken responses is to use a rhyme judgment task. KSR judged if a word and an arabic numeral rhymed (e.g., 4 pour), or did not rhyme (4 sour). Half of the rhyming and non-rhyming trials had word stimuli which were orthographically similar to the spelling of the numeral presented (e.g., 7 heaven), and half did not (e.g., 4 war).

KSR was visually presented examples of words and pictures which rhyme, and was asked to make similar judgments for the word/numeral pairs. KSR's performance largely
followed the orthographic similarity of the written form of the numeral and the word (even though the numerals were presented in arabic form). Almost all pairs which were orthographically similar were judged to rhyme (24/28 judged to rhyme; 14/28 correct), whereas approximately half of the non-orthographically similar pairs were judged to rhyme (15/28 judged to rhyme; 17/28 correct). KSR's overall rhyming performance of only 55% correct (31/56) revealed no signs of an intact ability to convert the stimulus numerals into the correct internal phonological form. This finding provides additional evidence consistent with the results from the arithmetic investigation reported above that KSR has a significant impairment in converting numerals into their appropriate phonological form.

**Additional Evaluation of the Phonological Storage Hypothesis**

While the previous section evaluated the notion that a phonological form of the problem must be used to retrieve arithmetic facts, this section evaluates the notion that answers to arithmetic problems are retrieved from memory in a phonological form. According to the phonological storage hypothesis, answers produced in any form (e.g., written, spoken) are based on an arithmetic fact which is originally retrieved in a phonological form. In order to write the answer, the retrieved answer must be converted from a phonological form into an arabic numeral form suitable for written production. In order to say the answer, no conversion is required, because the answer is retrieved in a phonological form which may be submitted directly to speech production mechanisms.

According to the phonological storage hypothesis, if KSR correctly produces the answer to a problem by writing the answer (e.g., stimulus: 5 x 6; response: 30), then the correct answer must have been retrieved in a phonological form.
Figure 12: According to the phonological storage hypothesis written and spoken answers are based on an arithmetic fact retrieved in phonological form.

Therefore, if KSR produces the correct written answer (indicating the answer was correctly retrieved in phonological form), he should also produce the correct spoken answer since the phonological answer representation may be submitted directly to speech production mechanisms (unless there is an error in speech production).

This hypothesis was evaluated by comparing KSR’s abilities to say the answer and write the answer. Since spoken responses are being used as an indirect measure of KSR’s phonological representation of the answer, we again consider the possibility that some spoken errors may possibly be attributed to perseverative or other speech production processes.

**Method**

All 390 simple arithmetic problems were presented in arabic form blocked by operation. KSR performed two versions of this task, one in which he said the answer and then wrote the
answer, and one in which he said the problem, said the answer and wrote the answer. KSR produced his responses after the problem was covered to avoid re-encoding of the problem after one response was made.

**Scoring Procedure**

The scoring procedure and perseveration analysis were modified for dealing with spoken answers rather than problems, but were otherwise the same as for the previous task. Spoken errors which could conceivably be considered to be speech production errors (e.g., stimulus: 3 + 4; response: "sevel"), or perseverations, were removed from the analysis. The remaining spoken responses should reflect KSR's ability to retrieve the phonological facts from memory.

**Results**

KSR's written answer performance continued to be highly accurate (99% correct; 772/780), while his spoken answers were only correct for 35% of trials (273/780). Table 12 provides examples of KSR's spoken and written answers to the arithmetic problems.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Spoken Answer</th>
<th>Written Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 + 7</td>
<td>&quot;twelve&quot;</td>
<td>14</td>
</tr>
<tr>
<td>10 - 1</td>
<td>&quot;eight&quot;</td>
<td>9</td>
</tr>
<tr>
<td>7 x 5</td>
<td>&quot;twenty six&quot;</td>
<td>35</td>
</tr>
<tr>
<td>16 / 4</td>
<td>&quot;eight&quot;</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 12: Examples of KSR's spoken and written answers to simple arithmetic problems.

After removal of all errors which could possibly be perseverations from the two previous trials, 442 error trials remained. These remaining trials should accurately reflect the phonological form in which the problem was retrieved. Contrary to the predictions of the
phonological storage hypothesis, there were a total of 167 trials in which KSR was able to produce the correct written answer, but yet was unable to produce the correct spoken answer. KSR was able to produce the correct written answer for 98% of trials whether or not his spoken answer was correct (267/273), or incorrect (167/169). This suggests that KSR was not retrieving arithmetic facts in a phonological form as predicted by the phonological storage hypothesis.

**Discussion**

KSR’s abilities were examined to evaluate the hypothesis that arithmetic facts are stored and retrieved exclusively in a phonological form. Contrary to the predictions of the phonological storage hypothesis, KSR’s ability to retrieve arithmetic facts from memory was not dependent upon correctly converting the stimulus problem into its phonological form. Rather, KSR’s ability to retrieve the answer to the problems (as measured by his written answers) was found to be highly accurate (more than 95% correct) whether or not he correctly converted the problem into its phonological form. This suggests that arithmetic facts are not stored exclusively in a phonological form.

The phonological storage hypothesis also assumes that arithmetic facts are retrieved in a phonological form. According to this hypothesis, spoken answer performance should be as accurate as written answer performance, since both answers are based on the same phonological answer form. However, KSR’s written answer performance was found to be much more accurate than spoken answer performance. KSR was highly accurate when producing written answers (~98% correct) whether or not the answer was correctly produced in spoken form. This indicates that answers were not retrieved exclusively in a phonological form.

Taken together, these results strongly contradict the predictions of the phonological storage hypothesis, suggesting that arithmetic facts are neither stored nor retrieved exclusively in a phonological form.
Alternative Hypotheses

While these results are highly problematic for the phonological storage hypothesis, other theoretical positions appear capable of handling these findings.

McCloskey's Semantic Retrieval Position

Within McCloskey's semantic retrieval framework we could interpret KSR's performance as consistent with an impairment in selecting the appropriate phonological form based on a semantic representation of that numeral.

Figure 13: KSR's deficit according to McCloskey's Semantic Retrieval Position

KSR's performance on tasks in which he transcoded numerals from one form to another (e.g., stimulus: 80; response: eighty) provided evidence that KSR was able to comprehend and produce arabic numerals. This is consistent within the McCloskey framework with the notion that KSR was able to convert arabic numerals into their corresponding semantic forms, and that KSR was able to convert semantic forms into forms suitable for arabic numeral production.
Given that there is also evidence that KSR was able to retrieve arithmetic facts from memory (e.g., timed arithmetic involving a keyboard response), within this framework it would expected that KSR should be able to accurately produce arabic numeral answers to arithmetic problems presented in arabic form.

However, KSR was also found to have severe difficulty in saying aloud arabic numerals. Within the McCloskey framework, one would interpret this deficit as an impairment in converting the semantic form of a numeral into its corresponding phonological form. This impairment should affect every task involving spoken numeral production, including numeral reading and arithmetic. Consistent with this prediction, KSR was found to have severe impairments in reading arabic numerals aloud, naming problem operands, and (as I have just described) saying arithmetic answers aloud. However, even when the problem is represented incorrectly in a phonological form, it may nevertheless be represented correctly in an abstract semantic form, allowing the appropriate arithmetic fact to be retrieved from memory. In this way one might expect KSR's somewhat unusual performance in which he says a problem incorrectly (e.g., stimulus: 5 x 6; spoken problem: "three times eight"), says the answer incorrectly (spoken answer: "forty"), yet writes the correct answer in arabic form (written answer: 30). Thus it appears that a theory which assumes that arithmetic facts are stored in an abstract semantic form is capable of accounting for KSR's performance.

Campbell's Multi-format Position

While Campbell has not produced an explicit theory of arithmetic fact retrieval, his hypothesis that arithmetic facts are stored in several different codes could potentially be compatible with the results of the KSR study. The results of this study suggest that arithmetic facts are not stored exclusively in a phonological form. However, arithmetic facts could be stored in several different forms, including a phonological form. It is important to note the results from the KSR study can not discount the possibility that normal individuals have phonological representations of arithmetic facts, providing there are also other forms in which
the arithmetic facts are stored. The KSR experiment does not directly address the possibility that there are multiple format specific arithmetic fact representations, and this remains a question for future research.

Dehaene's Revision of the Phonological Storage Hypothesis

Recently, Dehaene (1995) has modified his theory to accommodate the findings from brain-damaged patient KSR. Dehaene suggests that KSR's performance provides sufficient evidence to reject the notion that arithmetic facts are retrieved using a phonological form of the problem. He has modified his position by proposing that arithmetic facts might still be stored and retrieved using a level of representation in which there is a plan for verbal numeral production, but in which phonological forms have not yet been inserted. For example, McCloskey et al. (1986) propose that verbal numeral production involves a syntactic plan which includes both a lexical stack choice (tens, teens, ones) and a position within the stack (1, 2, 3...) for each number word (e.g., Tens:{6}, Ones:{4} for "sixty four"). Dehaene argues that "arithmetic facts are rote-memorized associations between filled word frame representations of the operands, rather than between their phonological representations" (p. 109; Dehaene, 1995).

Thus, in order to solve the problem 6 x 4, the problem must first be converted into a verbal word frame such as Ones:{6} {Times} Ones:{4}, in order to retrieve the appropriate answer, also in a verbal word frame (Tens:{2}, Ones:{4}).

This modification of Dehaene's theory does appear to be compatible with the findings from patient KSR. Dehaene suggests that within the new framework, KSR may still be able to retrieve arithmetic facts using the syntactic plan for phonological production, but has a specific impairment in selecting the phonological forms to fill in the syntactic plan for spoken production. For example, even if KSR is able to retrieve Tens:{2},Ones:{4} in response to the problem 6 x 4, he is still unable to select the appropriate phonological forms which correspond to Tens:{2} and Ones:{4}. However, the syntactic form of the answer may be converted into an appropriate arabic representation allowing the correct written arabic numeral to be produced.
While this theory is potentially consistent with the findings from patient KSR, it is not clear to what extent this representation should be considered 'phonological' in nature. The verbal word frame essentially provides a representation of the numerals and operation which acknowledges the specific oddities of a particular language's spoken representations of numerals (e.g., teens), but still requires that the positions within the lexical stack be encoded in an abstract form which is not tied to the phonology of the numerals. Thus it appears that the modification of Dehaene's theory comes close to a completely form-independent semantic theory of arithmetic fact retrieval.

In summary, it appears that arithmetic facts are not stored exclusively in a phonological form. Current explicit theories of arithmetic fact retrieval do appear converge on the notion that arithmetic facts are stored in a form which abstracts away from surface characteristics of the problem.

In the next chapter, a new theory of arithmetic fact retrieval is introduced which suggests that arithmetic facts are retrieved using a semantic representation of the problem (McCloskey, 1992), from a network of stored facts (e.g., Ashcraft, 1992; Campbell, 1985). The semantic representations of number will be based on research from non-arithmetic numerical tasks which suggests that humans possess innate representations of numerical magnitude. I will argue that a network retrieval theory involving semantic number representations is capable of providing a basis to account for the major findings within the cognitive arithmetic literature.
Chapter 1: Introduction to Semantic Network Retrieval Theory

This chapter introduces an alternative to current theories of arithmetic fact retrieval, providing both a detailed description of the theory, and a simulation used to explore the complex of the theory's predictions regarding errors, and reaction times.

The semantic network retrieval theory can be considered an amalgamation of two current theoretical positions on arithmetic fact retrieval, adopting the position that arithmetic facts are retrieved from a network of related facts (Ashcraft, 1992; Campbell, 1985), and the position that arithmetic facts are stored and retrieved in a semantic form (McCloskey, 1992). Evidence from non-arithmetic numerical tasks (such as magnitude comparison judgments) is used as a basis for assumptions about how numerals may be represented semantically (Gallistel & Gelman, 1992).

Semantic Representations of Magnitude

Evidence From Non-Arithmetic Numerical Tasks

Several studies of simple tasks such as deciding which of two numerals is the larger, or deciding if two numerals represent the same or different quantities have revealed interesting effects which suggest that internal representations of quantities are not unrelated individual tokens. Rather, it appears that the internal representation of quantities reflect the relations between magnitudes (e.g., 4 is closer in magnitude to 5 than 7). Described below are several robust findings from numerical tasks other than arithmetic fact retrieval which provide important clues about the nature of internal magnitude representations. These findings will be used to specify the nature of the semantic representations which are involved in arithmetic fact retrieval.

Magnitude Comparison

The most robust evidence for a semantic representation of quantity comes from the number comparison task. Moyer and Landauer (1967) revealed that the time required to decide
which of two numbers is the larger (or smaller) decreases as the numerical distance between the numbers being compared increases (commonly termed the distance effect). For example, response times are faster when subjects select the larger of 1 and 9 than when they select the larger of 4 and 6. Moyer and Landauer also report a 'magnitude effect': if the difference in magnitude between numbers is held constant, response times increase as the magnitude of the compared numbers increases (e.g., 2 and 4 are compared more quickly than 7 and 9). Both the distance and magnitude effects have been replicated numerous times with different types of stimuli such as arabic digits (e.g., Besner & Coltheart, 1979; Moyer & Landauer, 1967), written number words (e.g., Foltz, Poltrock & Potts, 1984), patterns of dots (Buckley & Gilman, 1974), and two-digit numerals (Hinrichs et al., 1981; Dehaene et al., 1990). The replication of the Moyer & Landauer (1967) findings with many different types of stimuli suggest that the effect is not the result of characteristics of the specific stimuli, but rather the result of differences in the characteristics of the abstract (i.e. semantic) representations of the magnitudes.

Moyer and Landauer proposed that numbers were being represented in terms of their magnitudes. Numeral pairs with similar magnitudes require more time to be compared because their magnitude representations were more similar to one another than other numeral pairs with dissimilar magnitudes. In addition, they assume that the differences between magnitude representations become less distinguishable the larger the numerals are. For example, they suggest that the difference between the representations of 2 and 3 are greater than the differences between the representations of 8 and 9. These variations relative to the magnitude of the numerals are used to account for the magnitude effect. Moyer and Landauer noted the remarkable similarities between the characteristics of performance in this task and psychophysical judgments of sound and light intensities, and suggested that perhaps the same type of comparison mechanisms and perhaps representations are used for both psychophysical and numerical magnitude judgments.
Same/Different Judgments

The findings from magnitude comparison judgments have been replicated in same/different judgment tasks. Duncan and MacFarlane (1980) reported both distance and magnitude effects for stimuli presented in the same format (e.g., 2 2; five six). Recently, Dehaene and Akhavein (1995) extended the Duncan and MacFarlane studies by studying performance when the pairs of numerals are in either the same format (e.g., 2 3) or in different formats (e.g., 2 three). Dehaene and Akhavein report both distance and magnitude effects whether or not the pairs of numerals were presented in the same format (e.g., 2 2; two four) or in mixed formats (e.g., 3 two), suggesting that a common representation was either used for comparison, or at least influenced performance. In fact, distance effects were still reported on same format trials even when subjects were only required to match physical similarity (e.g., deciding if 8 and 2 are physically identical), suggesting that semantic numeral representations are rapidly and automatically invoked even when not required to perform the task.

Dehaene and Akhavein also consider the source of the distance effect. One possibility is that this effect can be attributed to comparisons within mental lexicons, rather than comparisons of magnitude representations. Some evidence from normal subjects and brain-damaged patients can be interpreted as indicating that lexical forms are stored in ordered stacks (e.g., arabic numerals: 1,2,3,4,5,6,7,8,9; written numerals: one, two, three, four, five...) and that number words are grouped according to their class (e.g., ones: one, two...; teens: ten, eleven, twelve...; tens: twenty, thirty...). Retrieval of the appropriate form requires both information about which stack to choose from, and the position within that stack.
Table 13: Comparison of lexical stack positions of three, thirteen, nine, and nineteen.

The distance effect might simply reflect the time it takes to move along the lexical stack from one form to another. To test this possibility, Dehaene and Akhavein asked subjects to perform same/different judgments on numerals which come from different lexical stacks, but have the same position within each stack. For example, sample stimuli pairs might be three/nineteen and nine/thirteen. As displayed in Table 13, both pairs of numerals should be equally distant in terms of their position within the lexical stacks. However, the numerical magnitude between the pairs is quite different. If the distance effect reflects lexical positions, then no difference should found between the pairs. However, Dehaene and Akhavein found that numerical distance did matter, and that there was a distance effect such that pairs of numerals with larger numerical differences (e.g., three/nineteen) were faster than those with smaller differences (e.g., thirteen/nine), suggesting that same/different judgments are based on numerical magnitude representations, not positions within a lexical stack.

**Accumulator Theory**

In accounting for these findings Dehaene challenges the notion that number processing reduces to lexical effects or some other representation without magnitude (see Dehaene, 1992).
He suggests that such a position does not sufficiently account for the abilities of infants, animals, or adults. Instead, he suggests that humans have an approximate representation of numerical magnitude. Dehaene suggests that this approximate representation of magnitude might be characterized as a mental number line which becomes increasingly compressed as magnitudes increase (as displayed in Figure 14), and is accessed by a process which repeatedly activates small areas of the number line in approximately the correct location to produce a distribution of activation across the number line.

![Figure 14: Example of a compressive number line with activation pattern for 4.](image)

While all the details have not been fully worked out, the compressive number line theory is designed to provide two basic properties. First, the representations of numbers which are close in magnitude are more similar to one another than two numbers with dissimilar magnitudes (i.e., activation pattern for neighboring numbers such as 4 and 5 are likely to share common areas of activation, while other pairs such as 1 and 9 should not). Second, the larger the number being represented, the smaller the differences between the representation of that number and its close neighbors (e.g., 8 and 9 are closer than 2 and 3).

Gallistel and Gelman (1992) have also proposed that adults have an innate ability to represent numerical magnitude, and suggest that adult representations are linked to those found in 'prelinguistic' children, and animals. They propose that all animals (including humans) possess a non-verbal counting mechanism (see Mech & Church, 1983). According to this preverbal counting theory, an 'accumulator' collects impulses from a counting mechanism, the sum of which comes to represent the magnitude of a number. The impulses can be thought of as equivalent to cups of water added to a bucket. For example, when three objects or the numeral 3 is presented, it is proposed that the accumulator would accept three impulses (or
cups), and return the sum of the impulses (the bucket with three cups of water) as the representation of three. However, central to this theory's explanatory power is the notion that the impulses are inexact. The more impulses there are, the more variation there will be in the representation of that quantity.

While Moyer and Landauer, Dehaene, and Gelman and Gallistel make slightly different claims about how quantities are represented, two key elements of the representations of numerical magnitude seem constant across formulations: (1) representations of numerically close quantities are less distinct from one another than representations of numerically distant quantities, and (2) the larger the numeral being represented the less distinct its representation will be from representations of other numerals with similar magnitudes. These key aspects of magnitude representations are adopted in the semantic network retrieval theory.

Evidence From Arithmetic Production Tasks

Two aspects of arithmetic fact retrieval performance suggest that a representation of magnitude with the properties just described may be involved in arithmetic fact retrieval. First, operand errors in normal and artificial arithmetic tend to correspond to problems with operands which are close in magnitude to the correct operands (i.e., the operand distance effect). Within a network retrieval theory, the operand distance effect may be interpreted as suggesting that activation spreads from the operand representations to both the correct fact, and also to facts with operands which are close in magnitude to the correct operands. If the operands are represented so that the representations of similar magnitudes share common features, then facts with related operands would tend to become active and therefore sometimes erroneously retrieved. The more distant the erroneous response's operands are from the correct operands, the less likely that fact would be selected. This prediction is consistent with the operand distance effect in normal and artificial arithmetic: the closer the operands of the error are to the correct operands, the more likely that error will occur.
Another finding consistent with the notion that magnitude representations play a role in arithmetic fact retrieval comes from the artificial arithmetic studies. Reaction times and error rates were higher for problems with larger operands than those with smaller operands. One possible interpretation is that the representations of larger operands are more similar to one another and allow more activation of competing answers, resulting in more associative interference during retrieval than occurs with smaller operands.

Given the robust findings from non-arithmetic numerical tasks, and the possible influences magnitude representation may have on normal arithmetic, I suggest that arithmetic facts are stored and retrieved using a representation of numerical magnitude. Below I introduce one possible instantiation of a magnitude representation. However, theoretically I accept the basic assumptions provided by Dehaene and Gelman and Gallistel that numerical magnitude is represented such that: (a) the closer two numerals are in terms of their magnitude the more similar their representations are, and (b) the distinctions between adjacent magnitude representations become smaller the larger the quantity represented.

**Representing Numerical Magnitude in the Semantic Network Simulation**

Throughout the discussion of the semantic network retrieval theory (SNRT), relevant aspects of the network simulation of the SNRT theory will be introduced. The semantic network simulation is designed to remain as faithful as possible to the assumptions of the SNRT theory.

A representative diagram of how semantic number representations were incorporated into the network simulation is provided in Figure 15. The representations of the numerals 0 through 9 are presented. Each column in Figure 15 represents a unique pattern of activation for the same 250 nodes. Each node has an activation value of 0 if thin and unshaded, and .01 if wide and shaded. The total activation across nodes for each numeral representation sums to 1.
As can be seen from Figure 15, numerals with similar magnitudes share features with one another. The closer the magnitudes of the numerals, the more similar their representations are. For example, while the representations of 1 and 6 have only a small proportion of their activation in common, the representations of 1 and 2 share most features with one another.

The representation of numerals in Figure 15 also attempts to capture the notion that larger numerals share more features with one another than smaller numerals do. Differences between the representations of the numerals become smaller as the magnitude of the numerals being represented increases. For example, the representations of 1 and 2 share approximately
two-thirds of their representations, whereas the representations of 8 and 9 share almost 90% of their representations in common. While this instantiation of numerical magnitude may only approximate the actual representations, it does provide a possible manner in which the similarities among the representations of different magnitudes may be instantiated (see Dehaene 1992, for a more detailed discussion of possible magnitude representations for distributed networks).

**Cognitive Architecture**

**Basic Structure of Semantic Network Retrieval Theory**

The semantic network retrieval theory (SNRT) assumes that arithmetic facts are retrieved from an associative network of facts using a semantic representation of the problem. The fact retrieval system is composed of three groups of nodes representing: the current arithmetic problem, the stored arithmetic facts, and the answer output. The problem input includes a representation of the magnitude of the first operand, the magnitude of the second operand, and the arithmetic operation (not presented in Figure 16). The answer output nodes represent the magnitude of the tens component of the answer, and the magnitude of the ones component of the answer.

![Diagram of Semantic Network Retrieval Theory](image-url)
Each arithmetic fact is represented by an individual problem node which provides excitatory connections between its problem representation (e.g., the representation of 4 x 3) and its corresponding answer (tens:1; ones:2). Problem nodes also have inhibitory connections to problem and answer nodes which are not involved in representing its answer (e.g., the 3 x 4 problem node has inhibitory connections to tens nodes which are not involved in representing ten). Connections between problem nodes and answer nodes are bidirectional, allowing the activation of the answer to influence the activation of individual problem nodes. However, the connections between the presented problem and the problem nodes are unidirectional: the presented problem representations remain fixed through the fact retrieval process. It is assumed that problem nodes are also connected to other problem nodes in an inhibitory manner.

One unusual aspect of this simulation is the contrast between the distributed representations of the operands and answers, and the localized representations of the problems. Each problem is represented by an individual problem node whereas each operand and answer component is represented by the pattern of activation across several nodes. The primary motivation for this distinction comes from acquired arithmetic impairments. Brain-damaged patients suffer scattershot damage in the fact retrieval system such that some arithmetic facts may be severely impaired while others can be virtually unimpaired. This suggests that each arithmetic fact may be represented (to some degree) independently of other arithmetic facts. While representing each fact with an individual node may over simplify the actual representation of arithmetic facts and groups of nodes might actually represent each fact, this representation.

In contrast there are no reported cases of patients suffering an inability to represent a specific magnitude (e.g., 7) while retaining the ability to represent numerically close magnitudes (e.g., 6, 8). The ability to represent numerical magnitude appears to be extremely resistant to impairment (Dehaene, 1992), suggesting that numerical magnitude representations may be highly distributed in nature. In fact only one case study has been reported in which the patient
was unable to represent quantities larger than 4 (Cipolotti, Butterworth & Denes, 1991). This is consistent with the notion that magnitude representations of numerals are less distinct the larger the numeral being represented, and that the representations of larger numerals share common properties. Thus there appears to be support in the literature both for the notion that arithmetic facts represented independently, and that representations of numerical magnitude are distributed in nature.

**Simulation Framework**

The basic components of the semantic network simulation include 3 groups of nodes: 500 input nodes representing the presented problem, 64 problem nodes representing the stored multiplication problems, and 500 output nodes used to represent the answers to a problem. The 500 input nodes are divided into two equal groups: those that represent the magnitude of the first operand, and those representing the magnitude of the second operand. Only one arithmetic operation was simulated and therefore no arithmetic operation units were included in the simulation. The 500 output units also are divided into two equal groups of 250 nodes. The first group of nodes represent the magnitude of the tens component of the answer, and the second group represent the magnitude of ones component of the answer.

Each problem node represents a unique arithmetic fact by the nature of its connection weights with the operand and answer nodes. Problem nodes have positive connections to operand nodes which compose the representation of its operands. For example, the 8 x 5 problem node has positive connections to the nodes in the first operand representation which represent 8, and to the nodes in the second operand which represent 5. All remaining operand nodes have negative connections with the 8 x 5 problem node. An example of some connections is provided in Figure 17 below. This displays the connection weights between the last three operand1 nodes and the problem nodes 8 x 5 and 9 x 5. The representation of 8 in the operand nodes only includes the first node displayed, and so only that node has a positive connection weight with the 8 x 5 node. In contrast, all three nodes are involved in representing
9 x 5 and so all 3 nodes have positive connection weights with the 9 x 5 problem node.
Positive connections have a weight of 1 and negative connections have a weight of -1. These associations are unidirectional, from the operand nodes to problem nodes.

Problem nodes are also connected to answer units. Each problem node has positive connections to answer nodes which correspond to its answer. For example, the 4 x 6 problem node has positive connections to all of the units in the tens answer representation which correspond to the magnitude 2, and to all of the units in the ones answer representation which correspond to the magnitude 4. All remaining answer representation nodes which are not involved in representing that problem node's answer have negative connections with the problem node. These connections are bidirectional: answer units receive activation from the problem units, and the problem units receive activation from the answer units. Positive connection weights have a strength of +0.01 and negative connection weights have a strength of -0.01. The weights between problem and answer nodes are weaker than those between operand and problem nodes. This difference allows operand activation to be the primary influence on problem node activation.
Figure 18: Basic semantic network simulation architecture.

Finally, each problem node is connected to every other problem node. During activation, the problem nodes compete and mutually inhibit one another. The weight of the inter-problem-node connections is -0.05.

Retrieval of Arithmetic Facts from Memory

Activation is hypothesized to spread from the operand and operation nodes to the problem nodes, and then to answer nodes. Activation spreads both to the correct problem node, and other problem nodes with operands which are close in magnitude to the correct operand nodes. For example, consider how activation will spread through the network when the problem 4 x 6 is presented. Problem nodes which share the first operand with the presented problem (e.g., 4 x 5, 4 x 6, 4 x 8, 4+6) will all receive strong activation from the first operand representation. Other problem nodes with operands close in magnitude to the first operand (e.g., 3 x 6, 5 x 7) will receive partial activation from the first operand. Those with a very different first operand (e.g., 9 x 6) will receive inhibition from the first operand representation. These differences are all dependent on the amount of overlap in operand representations between the presented problem, and the operands of each problem node.
Activation will spread from the second operand in a similar manner. Those problems with second operands which are similar or identical to the presented second operand (e.g., 4 x 6, 4 + 6) will receive strong activation from the operand representation, and those with a less similar operand will receive less activation (e.g., 4 x 5) or be inhibited (e.g., 4 x 2) depending on the difference between the presented operand and the problem node's operand.

Activation also spreads from the operation nodes (note the simulation does not currently include operation nodes). Problems with the correct operation will receive activation while those in other operations will be inhibited.

Several factors affect the activation levels of the problem nodes. The strongest single factor is the activation from the operand and operation nodes. Problem nodes with operands close in magnitude to the presented problem (in the correct operation) will receive much more activation (and not inhibition) from the problem representation than problems with very different operands.

The second factor is inter-problem-node inhibition. Problem nodes mutually inhibit one another so that as activation accumulates over time, one node will generally win out and dominate answer node activation.

Answer node activation is the third influence on problem nodes. As the problem nodes accumulate activation, they in turn activate answer representations. Activated answer nodes feed activation back to problems with answers corresponding to the answer node activation pattern, and inhibit problem nodes with dissimilar answers. For example, if the answer node activation pattern approximated 72, all problems with a answer in the seventies (i.e., 8 x 9) would be activated by the tens representation, and all others would receive less activation, or be inhibited by the tens representation (depending on the difference between their tens representation, and the representation of seventy). Similarly, all problems with ones digit of 2 would be activated by the ones answer representations (e.g., 3 x 4; 6 x 7; 8 x 9), and others with more dissimilar activation patterns would receive less activation, or receive inhibition.
Finally, there are two other factors which are hypothesized to affect problem node activation. First, to accommodate error priming, it is assumed that problem nodes retain some activation after arithmetic fact retrieval which will affect subsequent retrieval attempts. For example if the answer to 4 x 6 was recently retrieved, the 4 x 6 problem node might retain activation, making this node more likely to be the winning node in subsequent trials. Second, it is also assumed that there is some random noise in the system such that incorrect problem nodes will sometimes exceed the activation of the correct problem node and receive the most activation from the problem representation. This provides an opportunity for errors to occur.

Arithmetic fact retrieval involves accumulation of activation in the retrieval system until the pattern of activation across the tens and ones answer nodes each arrives at the activation pattern for one of the ten magnitude representations (e.g., the tens nodes adopt an activation pattern representing thirty, and the ones nodes adopt an activation pattern representing two). On a typical retrieval attempt, several problems nodes initially become active. Problem nodes with operands which are closest in magnitude to the presented problem (including the correct problem node) will accumulate more activation than other nodes, creating sufficiently strong inhibition to reduce the activation of other nodes to zero. Over time this process of mutual inhibition typically results in a single winning node. Once the winning node sufficiently inhibits other problem nodes, the winning node will be the predominate influence on the answer nodes and will activate its answer node activation pattern. When one problem node is dominant, the activation pattern within the answer nodes will no longer be an amalgamation of several answers, and instead will form one of the ten activation patterns, allowing an answer to be retrieved. The answer retrieved has a separate tens and ones magnitude value which can then be converted into the appropriate form for production.

**Simulating Arithmetic Fact Retrieval**

---

3 Another logical possibility is that answer nodes also retain residual activation. However, the available reports of error priming have only noted errors for problems with similar operands, not problems with similar answers but very different operands (such as 9 x 2 priming 4 x 4).
The simulation of arithmetic fact retrieval involves the spread of activation from the operand nodes to the problem and answer nodes. A given trial begins with the introduction of operand node activation patterns corresponding to the two problem operands. Answer nodes begin with zero activation levels, and problem nodes begin with zero activation levels excepting any activation which may remain from previous trials (10% of previous final activation levels for the previous trial).

On each iteration the activation levels of the answer nodes are updated based on the current problem node activation levels. Operand node representations remain fixed. The activation levels of the problem nodes are modified based on the activation and inhibition received from the operand nodes, answer nodes, and competing problem nodes. In addition, each problem node receives a some random noise, a random value chosen from a normal distribution with a mean of zero and a standard deviation of 2. Activation levels for problem nodes may exceed 1 but do not decrease below 0. Activation levels of the answer nodes vary between .01 and 0.

A typical trial begins with the activation of several problems nodes. However, some problem nodes will accumulate activation more rapidly than others and will typically suppress the activation of other nodes, ultimately providing a single winning problem node in most cases. The iterative updating of the activation levels of the problem and answer nodes continues until the pattern of activation in the tens answer nodes, and ones answer nodes each approximates one of the ten possible activation patterns with less than 10% error. Percent error is defined as the sum of the difference between the correct and actual activation of each problem node, divided by the total activation of the correct numeral representation (in each case 1). When an answer has been selected, the number of iterations is recorded, and the tens and ones answer values is recorded as the retrieved answer.

Interpretation of Basic Phenomena
This section describes how the various arithmetic phenomena are accounted for within the semantic network retrieval theory. Each section includes the relevant simulation results.

**Problem Size Effect**

*Reaction Time Effect*

The Semantic Network Retrieval Theory accounts for the fact that larger problems (e.g., 7 x 8) have longer RTs than smaller problems (e.g., 4 x 6) by hypothesizing that larger problems have more inhibition from competing problem nodes than smaller problems do. Because larger operand representations are more similar to one another (e.g., the representations of 8 and 9 are more similar to one another than the representations of 3 and 4), larger operands will more strongly activate numerically close problem nodes. For example, 7 x 8 will activate its strongest competitors (e.g., 7 x 9, 8 x 8) more strongly than 4 x 6 will activate its competitors (e.g., 4 x 7, 5 x 6). Because problems with larger operands (e.g., 7 x 8) activate incorrect problem nodes (e.g., 7 x 9) more strongly than problems with smaller operands (e.g., 4 x 6) activate their competitors, the larger problems will receive more inter-problem-node inhibition than will problems with smaller operands. Because both problems with small and large operands receive the same amount of activation from operand nodes, the correct problem node for a problem with larger operands will take longer to accumulate activation than problems with smaller operands.

One way of measuring the reaction times of the simulation is to use the number of cycles the simulation took to reach the criterion activation level. If we assume that processing time in the simulation may approximate the process of updating activation levels in the actual system, then the number of cycles can be considered analogous to the time a subject might take to respond, and the 'pseudo RTs' from the simulation may be compared to reaction times from normal subjects.

As can be seen in Figure 19, the simulation of arithmetic fact retrieval does reveal a typical problem size effect. Problems with larger operands are slower on average than
problems with smaller operands, and the correlation between problem family (i.e. RTs for all 2's problems, 3's problems...) and simulation RT is quite strong ($R^2 = .82$).

![Figure 19: Simulation iteration counts according to problem family.](image)

While the general problem size effect is well replicated, the exceptions to this effect found in normal arithmetic appear somewhat weaker than those found in normal arithmetic. For example, tie problems were solved only slightly faster (average = 51) than other non-tie problems (average = 53). In addition, as can be seen in Figure 19, 5's problems are only modestly faster than might be expected relative to other problems, in comparison to the quite pronounced 5's effect in normal multiplication. The fives effect in the simulation is likely the result of similarities in the answer representations of 5's problems. Fives problems with an operand distance of 2 (e.g., current problem 5 x 5; problems with operand distance of 2: 3 x 5, 5 x 3, 7 x 5, 5 x 7) all have the same ones answers, and activate the appropriate pattern across the ones answer nodes which in turn further activates the correct problem node. Other problem families typically do not share ones answers.

*Error Effect*
The Semantic Network Retrieval Theory also predicts an increase in errors as the size of the operands increases, for the same general reasons that there was a problem size effect for reaction times. The larger the presented problem operands are in the presented problem, the smaller the difference between the activation of the correct problem node, and competing problem nodes, and therefore the greater the likelihood an erroneous problem node will exceed the activation of the correct problem node, resulting in the retrieval of an incorrect fact from memory.

The overall error rate for the simulation of multiplication was 9%. Figure 20 displays the error rates for the multiplication simulation by problem family. Consistent with the interpretation presented above, error rates increase in proportion to the magnitude of the problem operands.

![Figure 20: Simulation error rates by problem family.](image)

One somewhat unanticipated finding from the simulation is that for both the PseudoRTs and error rates, nines problems tend to be faster and more accurate than 7's and 8's problems. The nines effect in the simulation likely arises because these problems have fewer problems to compete with because the simulation is limited to the 2's through 9's problems.
In normal arithmetic there are mixed reports regarding this effect. For example, 9's problems were slightly faster than 8's problems in one report by Campbell (1985), but not in a more recent study (Campbell, 1995). In the pelification studies (e.g., Whalen et al., 1996) nines problems are slightly slower than 8's problems, but yet faster than might be expected from the regression across problem family.

**Error Types and Frequencies**

**Operand Errors**

The Semantic Network Retrieval Theory predicts that operand errors will be the most common error type because the most highly activated problem nodes, next to the correct problem node, are those which share an operand with the correct problem. For example, if we consider which problem nodes may be activated by the problem 4 × 6, it is predicted that all of the 4 × N problem nodes will receive full activation from the first operand representation, and all of the N × 6 problem nodes will receive activation from the second operand representation. Some problem nodes will receive activation from both operands. If we consider all of the 4 × N problem nodes, the 4 × 6 problem node should receive the most activation from the operand representations, since this problem node corresponds to both operands of the presented problem. In addition, the 4 × 5 and 4 × 7 problem nodes will receive full activation from the 4 operand representation, and partial activation from the 6 operand representation, since some of the semantic features of the 6 are shared by both the 5 and 7. Each problem node sharing an operand with the original problem will receive full activation from one operand, and a proportion of the activation from the second operand, depending on closeness in magnitude of the second operand of that problem to the presented problem. Other problem nodes which do not share an operand with the presented problem will not receive activation from either operand node (and perhaps receive inhibition).

The types and frequencies of errors the simulation produced were compared to the errors seen in normal arithmetic. Across one hundred runs of the 64 arithmetic problems the
simulation produced 578 errors, for an overall error rate of 9%. Below Table 14 compares the frequencies of different types of errors in normal arithmetic, and the SNRT simulation. Overall the error patterns for the SNRT simulation appear quite consistent with the findings from normal arithmetic. The most common type of error was the operand error, followed in frequency by table errors and non-table errors. No operation errors were reported because the simulation only modeled one arithmetic operation at a time. The differences in frequencies between normal arithmetic and the SNRT simulation for operand, table and non-table errors were not significant ($\chi^2(2)=4.7; p>.05$).

<table>
<thead>
<tr>
<th>Error Type</th>
<th>SNRT Simulation Frequency</th>
<th>Percentage</th>
<th>Normal Multiplication Frequency</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operand</td>
<td>486</td>
<td>84</td>
<td>509</td>
<td>77</td>
</tr>
<tr>
<td>Table</td>
<td>52</td>
<td>9</td>
<td>79</td>
<td>12</td>
</tr>
<tr>
<td>Non-Table</td>
<td>40</td>
<td>7</td>
<td>53</td>
<td>8</td>
</tr>
<tr>
<td>Operation</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 14: Errors in simulated and normal multiplication (Campbell, 1985).

Operand Distance Effect

The semantic network retrieval theory also predicts that among the problem nodes which share one operand with the presented problem, those with a second operand which is closer in magnitude to the second operand of the presented problem will receive more activation than other problem nodes from the same problem family with a very different second operand. For example the problem $4 \times 6$ will activate the $4 \times 5$ problem node more than $4 \times 2$ problem node because the representation of 5 is closer to 6 than is the representation of 2. Problem nodes with operands close in magnitude to the presented operands receive more activation from the operand representations, and would therefore be more likely to be selected as errors.
The simulation reveals a pattern comparable to the one found in normal arithmetic. Operand errors with smaller 'operand distances' were found to be more frequent than those with larger distances. In fact, approximately 90% of the operand errors had operand distances which were less than or equal to ±2. This is consistent with normal arithmetic, in which the vast majority of operand errors are within an operand distance of 2.

<table>
<thead>
<tr>
<th>Operand Distance</th>
<th>≤-3</th>
<th>-2</th>
<th>-1</th>
<th>+1</th>
<th>+2</th>
<th>≥+3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNRT Simulation</td>
<td>24</td>
<td>62</td>
<td>151</td>
<td>156</td>
<td>67</td>
<td>26</td>
</tr>
<tr>
<td>Normal Multiplication</td>
<td>30</td>
<td>68</td>
<td>182</td>
<td>143</td>
<td>65</td>
<td>21</td>
</tr>
</tbody>
</table>

Table 15: Operand errors separated according to operand distance.

The simulation also produced more operand errors with positive operand distances (i.e., the operands corresponding to the erroneous response tended to be larger than those of the actual problem) than operand errors with negative operand distances. This effect is likely attributed to the fact that the representation of any operand is slightly more similar to larger numeral representations than to smaller numeral representations. In contrast, it appears that in normal arithmetic (Campbell, 1985), there were more operand errors with negative operand distances. However, distribution differences are not significant either in terms of the operand distances ($\chi^2(5)=9.03; p>.05$), or when comparing differences between the number of negative and positive operand distances ($\chi^2(1)=2.76; p>.05$).

**Table Errors**

According to the semantic network retrieval theory, the same mechanisms responsible for operand errors should also produce table errors. The problem nodes which should receive the most activation from the operand representations include: (1) the correct answer; (2)
answers which share one operand in common with the problem (potential operand errors); and (3) those which do not share either operand, but are close in magnitude to each operand (potential table errors). For example, when the problem 4 x 6 is presented, it is hypothesized that the semantic representation of the first operand spreads activation not only to 4 x N problem nodes, but also to a lesser extent to the problem families 5 x N, 3 x N. The same type of process also occurs for the second operand, activating the N x 6 problem family, and to a lesser extent the problem families N x 7, N x 5, N x 8. Thus in addition to the correct answer and problems which share an operand, other problems such as 5 x 5, and 3 x 7 will also receive some activation from both operands. In this way we might expect that on some smaller proportion of trials, the activation level of these answers will exceed other problem nodes, resulting in the selection of that answer. This theory also predicts that there should also be a 'table error distance' effect: i.e. problem nodes with operands close in magnitude to the presented problem will receive more activation from the problem operands than problem nodes which are distant in magnitude from the presented problem. For example, when the problem 4 x 6 is presented, the table error 21 (3 x 7) should occur more often than the table error 18 (2 x 9) because the representation of 4 x 6 will provide more activation to the 3 x 7 problem node than to the 2 x 9 problem node.

Interestingly, the simulation did reveal an operand distance effect for table errors. When the operands corresponding to a table error were compared to the correct operands, it was found that the errors tended to have small operand distances. As can be seen in Table 16, the majority of table errors involved an error with operands which are each one different than the correct ones. In fact, only about 3% of table errors involved answers with operands which differed with the correct operand by 3 or more. This suggests, consistent with the SNRT perspective, that only problem nodes with operands close in magnitude to the correct operands become active, and compete to be selected. Differences between the SNRT simulation and normal
arithmetic in the frequencies of the different types of table errors were non-significant
($\chi^2(2)=2.57; \ p>.05$)

<table>
<thead>
<tr>
<th>Difference</th>
<th>SNRT Simulation</th>
<th>Normal Multiplication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operand 1</td>
<td>Operand 2</td>
<td>Frequency</td>
</tr>
<tr>
<td>±1</td>
<td>±1</td>
<td>37</td>
</tr>
<tr>
<td>±1</td>
<td>±2</td>
<td>9</td>
</tr>
<tr>
<td>±2</td>
<td>±2</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 16: Table errors separated by operand distances.

In summary, both operand errors and table errors are thought to be generated due to the
same underlying process of activation spreading from operand nodes to problem nodes for
problems which have operands which are close in magnitude to the presented problem.

*Operation Errors*

The Semantic Network Retrieval Theory posits that the same semantic representations of
the operands are used to retrieve arithmetic facts from memory regardless of the operation
performed. The role of the arithmetical operation representation is to provide more activation to
the problem nodes for the correct operation. However, on some occasions this process may not
be successful, allowing problem nodes from several operations to compete with one another.
In this case, an erroneous answer such as 2 x 8 = 10 might be expected. As in the correct
operation, the correct answer for the operands presented should typically be the most highly
activated. Only one operation was simulated at a time, and so further simulation will be
required to provide evidence about the frequency of this error type.

*Non-Table Errors*
In some cases, retrieval from the network of stored facts can result in two highly
activated problem nodes (e.g., 4 x 6 and 4 x 7). In some of these cases, the combination of the
activation from these problem nodes may result in a blend of activation in the answer
representations which produces a non-table error. For example, while both problem nodes 4 x
6 and 4 x 7 will activate a 2 representation in the tens value, the problem nodes respectively
activate 4 and 8 representations in the ones representation. Because the representations of 4 and
8 both share common features with the representation of 6, in some cases the 6 representation
might receive more total activation from both problem nodes than either problem node will
contribute to their individual correct answers. In this way, on some trials there may be an error
which does not reflect any correct answer within the table. However, the mutual inhibition
between problem nodes should generally result in a single problem node dominating the others
in terms of activation, resulting in the retrieval of its corresponding answer from memory.

Consistent with this interpretation are the results from the SNRT simulation. Non-table
errors were relatively rare, accounting for 7% of errors. These errors generally occurred when
there was no single winning problem node, and two or more problem nodes with similar
operands (e.g., 4 x 6 and 4 x 7) closely competed with one another. Interestingly, most non-
table errors resulted in an error which had the correct tens value, and a ones value which was
close to correct (e.g., 6 x 7 = 44; 8 x 7 = 54), and only rarely was the tens digit incorrect (e.g.,
9 x 8 = 62). This effect appears to relate to the fact that most competing problems (e.g., 4 x 6
and 4 x 7) tend to have the same tens digit answer. On occasions where the non-table errors
were produced, the tens answer representation generally settled on one particular representation,
and the competing problem nodes both received beneficial activation from the tens
representation, and competed to dominate the ones answer representation. Harley (1990)
performed an intensive study of non-table errors in normal arithmetic and found that errors in
the ones digits were somewhat more likely than errors in the tens digit (tens digit errors: 45;
one digit errors: 59).
Primbing

Campbell has repeatedly reported that previously retrieved answers are more likely to be errors to subsequent problems than other answers. The semantic network retrieval theory accommodates positive error priming by assuming there may be residual activation of problem nodes after answer retrieval. For example, if the problem 4 x 6 were presented previously (and the 4 x 6 node was most highly activated), then this problem node may have residual activation for subsequent trials. If on a subsequent trial a problem is presented with operands which are close in magnitude to 4 x 6, such as 4 x 7, the combination of the activation from the 4 x 7 representation and the previous activation would make the 4 x 6 problem node much more competitive than would otherwise be the case. The combination of residual activation and new activation (because 4 x 6 is close in magnitude to 4 x 7) might be sufficient for the 4 x 6 problem node to again receive the most activation, resulting in the retrieval of the answer 24 rather than 28. While 24 could be an erroneous answer to the problem 4 x 7 in any case, it would be expected that the residual activation of the 4 x 6 problem node would increase its likelihood of being the most highly activated.

Indeed the simulation was performed both with and without residual activation of problem answer nodes. The introduction of 10% of the previous trial’s activation on the next trial did produce a mild increase in the number of errors, from an error rate of 7% without residual problem node activation to a level of 9% with residual activation. Consistent with the notion that these errors may be attributed to previous activation, the introduction of residual activation increased the number of repeated answers (e.g., previous trial: 6 x 4 = 24; current trial: 7 x 4 = 24) from approximately one per multiplication block to 3 per block.

Thus it appears that residual activation of problem nodes may be a reasonable account of how error priming occurs within the SNRT theory. However, this simulation only promotes an answer in the immediately following trial. Future simulations will have to extend the residual
activation effect, as it is reported in normal arithmetic for roughly ten trials, not just the immediately preceding trial.

**Dyscalculia**

This section will consider how the SNRT theory might account for patterns of impaired performance. Only theoretical issues will be discussed as the simulation was not 'lesioned' to simulate dyscalculia.

*Problem Size Effect*

Acquired arithmetic fact retrieval deficits impair larger problems on average more than smaller problems. Within the SNRT framework, this is consistent with 'scatter-shot' damage throughout the fact retrieval system. Because problems with larger operands already face more competition to be selected than smaller problems, random damage to connection weights would affect the retrieval of these facts most seriously. Another possibility is that brain damage may result in general increases in the amount of random activation in the fact retrieval system. This would also result in more retrieval errors for larger problems because the competition between problem nodes is greater, and the amount of random noise needed to allow an erroneous problem node to exceed the activation of the correct problem nodes would be smaller.

*Non-Uniform Error Pattern*

The semantic network theory can straight-forwardly account for the fact that arithmetic fact retrieval impairments result in non-uniform patterns of impairment. For example, patients can be severely impaired on the problem 6 x 8, virtually unimpaired for 7 x 8, and mildly impaired for 8 x 8. Because each problem has its own problem node, and its own operand-problem and problem-answer connections, if any of these connections or the nodes themselves are impaired, this will primarily affect that specific problem (excepting the influence that problem node may have had on other problems).

*Error Types*
Two basic patterns of errors were reported in studies of dyscalculia. First, most patients appear to produce the same types and relative frequencies of errors as in normal arithmetic (e.g., operand errors, table errors, etc.), though their overall error rates may be much larger. According to the SNRT theory, this pattern might be accounted for by assuming that individual problem nodes (or their connections to operand representations) may be damaged, making them rarely able to exceed the activation of other problem nodes. Were this the case, the problem nodes which would normally be the strongest competitors with the correct problem node would be expected to be the most likely candidates to exceed the activation of the correct problem node, just as is the case in normal arithmetic. Because these errors are generated using the same mechanisms as in normal arithmetic, the same types and frequencies of errors would be expected.

However, the second pattern of errors from brain-damaged patients is somewhat different. Some patients produce a larger proportion of non-table errors than is normally reported. The non-table errors do appear to have a systematic relation with the correct response. If the non-table error is compared with the correct answer, the error typically has tens and ones values which are close in magnitude to the correct answer (e.g., $4 \times 7 = 29$; $9 \times 8 = 62$). For example, 80% of FW's non-table errors have an incorrect tens or ones digit which is within ±2 of the correct answer. This is consistent with the possibility that associations between the problem nodes and answer representations may be specifically affected. Even if the correct problem node is most highly activated, it must be able to activate the appropriate tens and ones representation in order for the appropriate arithmetic fact to be retrieved. If the association between the problem node and answer representations are affected, then one might expect that an answer with a tens and ones value close to the correct one will be selected.

**Simulation Results and the Semantic Network Retrieval Theory**

Broadly, the SNRT simulation appears to adequately account for a number of arithmetic phenomena. The simulation produced a problem size effect both in terms of solution time and
error rates. The error rates and types produced by the SNRT simulation also appear to be generally consistent with the findings from normal multiplication. The simulation had an overall error rate of 9% (normal speeded error rates range from 5-15%). As found in normal arithmetic, most errors were operand errors, followed in frequency by table errors and non-table errors. The simulation also replicated the operand distance effect for both operand and table errors. Errors tended to have operands which were close in magnitude to the correct operands. Finally, incorporating residual activation of problem nodes appears to be capable of accounting for error priming. Problems with residual activation were much more likely to occur as errors when the problem had operands which were close in magnitude to the correct operands.

Despite replicating all major phenomena, there do appear to be some challenges for the SNRT theory as currently formulated. For example, in the simulation 9’s problems tended to have an unexpected RT and error rate advantage over 7’s and 8’s problems. Perhaps the simulation might overcome the differences in the number of competing problems (between 7’s, 8’s and 9’s problems) by exaggerating the magnitude effect in the operand representations. However, there are some reports of normal arithmetic which are consistent with this finding (Campbell, 1985).

More generally, future work must incorporate all 4 operations, and include a simulation of performance after brain-damage. Nevertheless, the simulation does indicate that the general framework of the Semantic Network Retrieval Theory is capable of accounting for the major cognitive arithmetic phenomena.
VI: EVALUATION OF SEMANTIC NETWORK RETRIEVAL THEORY

While the findings presented up to now appear consistent with the assumptions of the Semantic Network Retrieval Theory, there has not yet been a direct evaluation of this theory. The experiment described below is designed to test the theory's predictions about the effects of inter-problem competition on retrieval.

Rationale

The semantic network retrieval theory assumes that several problem nodes become active and compete to be selected during fact retrieval. The notion of inter-problem-node competition is the basis for accounting for differences in RT across problems, and the error rates and types. For example, the semantic network retrieval theory proposes that RT and error rate differences between large and small problems are the result of differences in the amount of competition between the correct problem node and other problem nodes. Problems with larger operands activate competing problem nodes more strongly than problems with small problem operands (because of the nature of the operand representations). Stronger activation of competing nodes results in greater inhibition of the correct problem node, resulting in slower activation of the correct problem node for larger problems.

Next to the correct problem node, problem nodes with operands which are close in magnitude to the correct problem's receive the highest activation levels, and therefore will be the primary source competition which inhibits the correct problem node. For example, when the problem 4 x 6 is presented, the problem node 4 x 6 will receive the most activation from operand representations. However, other problem nodes such as 3 x 6 and 4 x 7 will also receive significant activation from the operand representations, and will provide the greatest inhibition and to the correct problem node. This experiment is designed to evaluate the notion that problem nodes with operands close in magnitude to the actual problem operands compete with, and inhibit the correct problem node.
According to the semantic network retrieval theory, if a problem were to have fewer competing problems with operands which are close in magnitude, there would be less inhibition of the correct problem node, resulting in a relative RT and error rate benefit for the problem. For example, if we artificially removed the 4 x 5, 4 x 7, 3 x 6 and 5 x 6 problem nodes from the network of stored arithmetic facts, the SNRT theory predicts that performance on the problem 4 x 6 would be more accurate and faster than was previously the case when all its 'competitors' were present. The reason for this difference lies in the fact that the 4 x 5, 4 x 7, 3 x 6 and 5 x 6 problem nodes will receive the most activation when the problem 4 x 6 is presented (next to the 4 x 6 problem node). Therefore these problem nodes are primary source of inhibition for the 4 x 6 problem node. Without these nodes present, the 4 x 6 node will receive less inhibition than would otherwise be the case, thereby resulting a net increase in performance. Consistent with this hypothesis are the results from a network simulation of this example (using the network simulation described in the previous chapter). Figure 21 reveals that the removal of the four problems discussed above provides significant improvements in the number of iterations required for a solution to be retrieved for the problem 4 x 6.

![Figure 21](image-url)

Figure 21: Simulation solution times before and after the removal of the problems 4 x 5, 4 x 7, 3 x 6 and 5 x 6.
The removal of these four problem nodes should also result in fewer errors for the problem 4 x 6. Because the problem nodes 4 x 5, 4 x 7, 3 x 6 and 5 x 6 normally are the most highly activated next to the correct problem node, these problem nodes are the most likely to exceed the activation of the correct problem node and be erroneously retrieved. If these problems are removed, then errors can only occur as a result of other problem nodes exceeding the activation of the correct problem node. However, the remaining problem nodes (e.g., problem nodes 3 x 7; 5 x 5) will receive much less activation than the 4 x 6 problem node when 4 x 6 is presented, and therefore would be unlikely to exceed the activation of the correct problem node. Thus, removal of the strongest competitor problem nodes is also expected to reduce the frequency of errors.

While the removal of these problems (4 x 5, 4 x 7, 3 x 6 and 5 x 6) has a large effect on performance of the problem 4 x 6, the removal of the problems would not significantly effect performance for the problem 6 x 7 because the problem nodes which compete most strongly with 6 x 7 (e.g., problem nodes 6 x 6, 6 x 8, 5 x 7, 7 x 7) are still present in the network of stored facts and will still compete with and inhibit the 6 x 7 problem node. The problem nodes which were removed such as 4 x 5 received very little activation from 6 x 7, and therefore their removal will result in very little change in the inhibition from competitors of the 6 x 7 problem node. The simulation of multiplication produced an effect which is consistent with this interpretation. Figure 21 reveals there was relatively little change in the solution times for the problem 6 x 7 after the four problems were removed.

Because arithmetic facts cannot be 'removed' from real multiplication, a study was conducted using an artificial arithmetic operation. Rather than teach subjects a complete table of 64 artificial arithmetic facts, subjects were trained to retrieve a set of facts which omits a subset of problems (e.g., 5 ÷ 6 might never be trained or tested). The 'removal' of artificial arithmetic facts should have greater performance consequences for some problems (e.g., 5 ÷ 7) than
others (2 ≠ 3). These differences were systematically varied across problems to evaluate notion of inter-problem competition.

**Experimental Design**

The construction of the artificial arithmetic operation followed the basic design of the previous pelletification experiments (e.g., Whalen et al., 1996). For example, the problems to be trained and tested were selected from the set 2 ≠ 2 - 9 ≠ 9, and the answers ranged from 23-98 excepting any real multiplication answers. This artificial operation was called 'diamond arithmetic' because a diamond sign "≠" was used as the operation symbol.

The critical manipulation in this experiment was the removal of problems from the diamond arithmetic tables in order to create two sets of problems: one set with very strong competition from neighboring problem nodes during retrieval, and a second set with relatively little competition from competing problem nodes. Each subject was trained and tested on 48 of the possible 64 problems in the 2 ≠ 2 - 9 ≠ 9 table. This number of problems was chosen so that the amount of competition each problem node received could be systematically varied, but the number of arithmetic facts to be stored and retrieved remained large and would approximate the complexity of normal arithmetic. The next sections describe the manner in which the problems and answers were selected, and the subjects were trained.

**Removing Problems from The Diamond Arithmetic Tables**

A computer program was used to pseudo-randomly select which problems would not be trained or tested for each subject. There were several constraints on the removal of problems from each subject’s diamond arithmetic table to ensure that problem node competition was systematically varied and measured independently from the effects of other possible performance factors such as operand size and answer size.

Two problems were removed from each problem family (e.g., 2 ≠ N, N ≠ 4). The problems removed provided a diamond arithmetic table in which half of the problems had
strong competition from neighboring problems, and the other half had relatively little
competition from competing problem nodes (the method of calculating competition values is
described in detail below). There was no correlation between measures of problem size (e.g.,
problem family, sum of operands, product of operands) and the problem competition values.

**Competition Values**

Each problem is assigned a 'competition value' which estimates the amount of inhibition
that a problem node will encounter during fact retrieval from other problem nodes which have
operands close in magnitude to its own operands. This section describes how competition
value estimates were computed based on normal multiplication error patterns.

According to the SNRT theory, the frequency with which a particular error occurs
should be related to the amount of activation that problem node receives during retrieval. The
higher the activation of the competitor, the greater the likelihood that particular problem node
will exceed the activation of the correct problem node and be erroneously retrieved. For
example, if one error (e.g., 6 x 8 = 56) is more frequent than another (e.g., 6 x 8 = 72), then
the problem node for the more frequent error (7 x 8) likely receives more activation from the
operand nodes than the problem node for the less likely error (9 x 8). Problem nodes that are
strongly activated by the problem operands will also be strong inhibitors of the correct problem
node. For example, when 6 x 8 is presented, the problem node 7 x 8 receives more activation
than the problem node 9 x 8 and therefore also provides greater inhibition to the 6 x 8 node than
does the 9 x 8 node. Because the frequency of errors, and strength of inhibition are related,
error rates can be used as a basis for calculating the amount of inhibition a particular problem
node imposes on the correct problem node.

For this reason, error frequencies from a normal multiplication production task were
used to estimate competition values. However, because there are a limited number of errors
reported in the literature, the error corpus will be pooled together to form a single competition
value formula which will be for all problems, rather than calculating competition values for each
problem separately based on the errors for that particular problem. A single competition value formula will not distinguish between problems with small and large operands, even though the semantic network retrieval theory suggests that problems with larger operands should have greater competition than problems with smaller operands. Thus this computation may be considered a first step in studying inter-problem competition which provides a relatively coarse-grained analysis.

\textit{Error Frequencies}

What we wish to determine is the amount of competition a competing problem node will provide the correct problem node. This value will vary depending on the difference between operands of the competing problem node, and the correct problem node (presumably competitors with operands which are close in magnitude to the correct problem node's operands will be the strongest competitors).

To calculate competition values according to the differences between the operands of the competitor and the correct problem node, the errors from normal arithmetic were separated according to the difference between the correct operands, and the operands corresponding to the error. For example, if the problem 4 x 6 was presented, the operand error 28 would be categorized as having one operand which is equal to the actual operand, and a second operand which differs by one with the actual operands (i.e., operand distance 1 = 0; operand distance 2 = 1). In contrast, a table error will not have either operand in common with the actual problem. For example, if the problem was 4 x 6, the table error 25 would be classified as having an operand distance of 1 for both the first and second operands (no distinction was made between positive and negative operand distances).

Operand and table errors which corresponded to problems with operands within ±2 of the actual operands were included in the analysis, since the vast majority of errors come from this range (e.g., 2 x 3 = 8 and 2 x 3 = 10 were included, 2 x 3 = 21 was not). The errors in the
Harley (1990) study of multiplication were used in this analysis\(^4\), which is provided in Table 17.

\begin{table}[h]
\centering
\begin{tabular}{ccc}
\hline
Operand Distance 1 & \multicolumn{2}{c}{Operand Distance 2} \\
\hline
 & ±1 & ±2 \\
0 & 427 (4 x 6 = 28) & 227 (4 x 6 = 32) \\
±1 & 106 (4 x 6 = 35) & 47 (4 x 6 = 40) \\
±2 & 22 (4 x 6 = 48) & \\
\hline
\end{tabular}
\caption{Table 17: Reported frequencies of errors sorted according to the difference between the actual operands, and the operands of the erroneous response (examples provided in parentheses).}
\end{table}

Table 17 reports the frequencies of the different error types. The frequency reported in any one cell is actually the sum of the four possible errors (i.e. the (0,±1) cell includes the following errors: (0,+1), e.g., 4 x 5 = 24; (0,-1), e.g., 4 x 5 = 16; (-1,0), 4 x 5 = 15; (+1,0), 4 x 5 = 25), except the (±1,±2) cell which includes eight different possible error types. Operand errors have one operand in common with the actual operands, and are therefore reported in the first row. For example, the frequency of operand errors with an operand distance of ±1 (e.g., 4 x 5 = 24) is reported in the (0,±1) cell. Table errors are reported in the second and third rows.

Each of these error frequencies provides a measure of the amount of competition provided to the correct problem from other problem nodes relative to the difference between the competing problem node's operands and the actual operands. For example it appears that the strongest competition arises from competitors with one operand in common with the correct problem, and a second operand which differs with the correct problem's second operand by one (e.g., presented problem: 4 x 6; strongest competitors: 3 x 6, 5 x 6, 4 x 5, 4 x 7).

---

\(^4\)The same procedure was performed with Campbell and Clark's (1985) data with nearly identical results.
**Conversion to Comparable Values**

Direct comparisons can not be made between values from different cells (e.g., about twice as many (0,±1) errors as (0,±2) errors) because some error types have fewer opportunities to occur than others. For example, the problem 2 x 5 is not expected to produce an error in which the first operand is either one or two less than correct because those errors would fall outside of the 2 x 2 - 9 x 9 set of problems. Similarly, the problem 6 x 8 is not expected to produce an error in which the second operand is greater than 9, because error falls outside of the 2 x 2 - 9 x 9 problem set (this holds true for any N x 8 problem). To accommodate the systematic variation in the number of problems contributing to each cell in the table provided above, a conversion was applied to make the absolute error rates comparable in terms of their relative opportunities.

Finally, the adjusted error rates were subdivided into the individual possible error types (e.g., (0,±1) was divided into (0,+1), (0,-1)). This provides a measure of how much inhibition is contributed by each individual problem node. The new values in proportions are presented in a concrete example below in Table 18.

Overall competition values for a particular problem are calculated by adding the appropriate amount of competition for each of the correct problem node's competitors. For the problem 4 x 6, the competition values from each competitor have been summed together to produce the competition value for that problem (914). According to Table 18, when the problem 4 x 6 is presented problem nodes such as 4 x 5 and 4 x 7 each contribute 107 units of competition, or 11.7% of the total competition value to the of the correct problem node. In contrast, the 2 x 8 problem node provides considerably less competition (8, or 0.9%) to the 4 x 6 problem node.
Table 18: The amounts of competition problem 4 x 6 receives from each of its competitors.

As presented in Table 18, problem 4 x 6 has all its competitors present, and receives competition from 24 different problems for a total combined competition value of 914. However, some problems (particularly in diamond arithmetic) will not have all possible competing problems. Continuing with the example provided earlier, if the problems 3 x 6, 5 x 6, 4 x 5, and 4 x 7 are removed the 4 x 6 problem node will no longer have its full complement of competition. In fact, as presented in Table 19, then the total combined competition from the remaining problems lowers to 486, about half that of 4 x 6 with its full complement of competitors. This value is again calculated by adding together the appropriate competition values corresponding to each competitor problem node present.
Table 19: The competition problem node 4 x 6 receives when 3 x 6, 5 x 6, 4 x 5 and 4 x 7 are removed.

While the removal of these four problems significantly lowers the total competition for the problem 4 x 6, other problems such as 6 x 7 will be much less affected by this change. As shown in Table 20, the removal of these four problems reduces the competition for the problem 6 x 7 from 914 to 810.

Table 20: The competition problem node 6 x 7 when 3 x 6, 5 x 6, 4 x 5 and 4 x 7 are removed.

Competition values for diamond arithmetic are calculated exactly the same way. For each problem, the total competition value was calculated by adding the appropriate competition...
value for each relevant competing problem node. For example, if a subject was only taught $4 \times 6$ and $4 \times 7$, the competition value for $4 \times 6$ would include only one competitor value, that of the problem $4 \times 7$ (107). However, if the same subject was then also taught the problem $4 \times 5$, the total competition value for the problem $4 \times 6$ would then include the competitor value for $4 \times 5$ (107) plus the competitor value for $4 \times 7$ (107), and equal 214. More generally, if all 24 competing problem nodes are present, then the problem node receives the highest competition value of 914. If all 24 competing problem nodes are removed, the problem receives the lowest competition value of 0.

<table>
<thead>
<tr>
<th>$A-2 \times B-2$</th>
<th>$A-2 \times B-1$</th>
<th>$A-2 \times B$</th>
<th>$A-2 \times B+1$</th>
<th>$A-2 \times B+2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(8)</td>
<td>(9)</td>
<td>(66)</td>
<td>(9)</td>
<td>(8)</td>
</tr>
<tr>
<td>$A-1 \times B-2$</td>
<td>$A-1 \times B-1$</td>
<td>$A-1 \times B$</td>
<td>$A-1 \times B+1$</td>
<td>$A-1 \times B+2$</td>
</tr>
<tr>
<td>(9)</td>
<td>(30)</td>
<td>(107)</td>
<td>(30)</td>
<td>(9)</td>
</tr>
<tr>
<td>$A \times B-2$</td>
<td>$A \times B-1$</td>
<td>$A \times B$</td>
<td>$A \times B+1$</td>
<td>$A \times B+2$</td>
</tr>
<tr>
<td>(66)</td>
<td>(107)</td>
<td></td>
<td>(107)</td>
<td>(66)</td>
</tr>
<tr>
<td>$A+1 \times B-2$</td>
<td>$A+1 \times B-1$</td>
<td>$A+1 \times B$</td>
<td>$A+1 \times B+1$</td>
<td>$A+1 \times B+2$</td>
</tr>
<tr>
<td>(9)</td>
<td>(30)</td>
<td>(107)</td>
<td>(30)</td>
<td>(9)</td>
</tr>
<tr>
<td>$A+2 \times B-2$</td>
<td>$A+2 \times B-1$</td>
<td>$A+2 \times B$</td>
<td>$A+2 \times B+1$</td>
<td>$A+2 \times B+2$</td>
</tr>
<tr>
<td>(8)</td>
<td>(9)</td>
<td>(66)</td>
<td>(9)</td>
<td>(8)</td>
</tr>
</tbody>
</table>

Table 21: The competition weighting scale for any problem.

*Competition Values for Normal Arithmetic*

The competition value algorithm was applied to the all 64 normal arithmetic problems. Presented in Table 22 are the competition values for normal arithmetic where all 64 problems are included in the analysis (exactly the same result would obtain for a diamond arithmetic table with 64 problems). Note that only the problems in the middle of the table (e.g., $4 \times 6$, $5 \times 6$) have all 24 competing problems, and therefore these problems are the only ones that receive the maximum competition score 914.
Sample Diamond Arithmetic Table

Below is an example of a diamond arithmetic table of facts and each step in creating a table including the removal of problems, the calculation of competition values, and the selection of answers. Table 23 presents a set of 64 problems in which 16 have been removed. The stars correspond to the cells for which no problem was presented, or tested. For each problem family (e.g., the 2 x N family in row 1) there are exactly two problems removed (2 \* 2 and 2 \* 6).

<table>
<thead>
<tr>
<th>Operand 1</th>
<th>5</th>
<th>*</th>
<th>*</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>7</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>8</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>9</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

Table 23: Display of 16 problems removed from training and testing for one diamond arithmetic fact set.

Table 24 provides the competition values for this particular set of problems. The average competition value is 512, and the standard deviation is 110. The smallest competition value is 283, while the largest competition value is 841. Problems with few competing
problems (e.g., 2 ÷ 9, 6 ÷ 8) have relatively smaller competition values than problems with more neighbors (e.g., 5 ÷ 6, 4 ÷ 4).

\[ \text{Operand 2} \]

\[
\begin{array}{cccccccc}
2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
2 & 283 & 356 & 338 & 475 & 366 & 302 & \\
3 & 439 & 512 & 558 & 521 & 539 & 329 & \\
4 & 384 & 594 & 676 & 713 & 686 & 558 & \\
\end{array}
\]

\[
\begin{array}{cccccccc}
\text{Operand 1} & 5 & 6 & 3 & 1 & 6 & 5 & 8 & 7 & 3 & 1 & 7 & 6 & 8 & 5 & 7 & 6 & 3 & 3 & 8 & 6 & 576 & 649 & 841 & 768 & 622 & 631 & \\
7 & 356 & 558 & 631 & 612 & 548 & 439 & \\
8 & 366 & 494 & 494 & 558 & 512 & 347 & \\
9 & 366 & 366 & 420 & 338 & 356 & 283 & \\
\end{array}
\]

Table 24: Competition values for reduced problem set presented in Table 23.

**Diamond Arithmetic Facts**

The diamond arithmetic facts were designed to share as many characteristics of normal arithmetic as possible. Answers were selected from the numerals 23-98, excepting any real multiplication answers for the 2's to 13's times tables. Answers did not systematically vary with the size of the operands (e.g., size being defined as the sum of operands, product of operands, or problem family). Problems and answers were randomly paired so that non-retrieval strategies (e.g., solving 6 x 7 by adding six 7s) were not available to solve the problems (e.g., 2 ÷ 3 = 71; 2 ÷ 4 = 94...), ensuring that subjects could only solve the problems by retrieving diamond arithmetic facts from memory.

\[ \text{Operand 2} \]

\[
\begin{array}{cccccccc}
2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
2 & 47 & 73 & 89 & 38 & 97 & 41 & \\
3 & 59 & 62 & 57 & 29 & 86 & 76 & \\
4 & 73 & 57 & 82 & 53 & 39 & 23 & \\
\end{array}
\]

\[
\begin{array}{cccccccc}
\text{Operand 1} & 5 & 6 & 3 & 1 & 6 & 5 & 8 & 7 & 3 & 1 & 7 & 6 & 8 & 5 & 7 & 6 & 3 & 3 & 8 & 6 & 2 & 95 & 37 & 17 & 83 & 47 & 9 & \\
7 & 38 & 86 & 43 & 34 & 91 & 58 & \\
8 & 97 & 23 & 67 & 91 & 26 & 94 & \\
9 & 41 & 76 & 52 & 37 & 58 & 68 & \\
\end{array}
\]

Table 25: Sample of diamond arithmetic answers.
Training

The training procedure largely followed the procedure used in Whalen et al. (1996) in the pelification experiments. Broadly, the sessions can be divided into three distinct stages: early training, evaluation/training, and a final test phase.

For the first two sessions subjects were trained on all 48 problems once per session. The 48 problems were divided into 6 blocks of 8 problems. During a block each of the 8 problems studied individually were paired with their answer (e.g., \(2 \diamond 4 = 73\)) for 4 seconds. The same 8 problems were then tested individually (e.g., \(2 \diamond 4 = \)). Subjects said the answer aloud, and feedback (correct/incorrect) was provided. The block was complete when the 8 problems were again presented one at a time paired with their answers as a review for 4 seconds (e.g., \(2 \diamond 4 = 73\)). Groups of 8 problems were trained, tested, and reviewed in this manner until all 6 blocks were completed. The diamond arithmetic facts were randomly assigned to different groups for each run of training.

The third and fourth sessions followed the same general procedure as the first two sessions except for two changes. First, subjects were trained on all problems twice per session. Second, subjects were provided with less study time: problems were matched with their answers for only 2 seconds, rather than 4 seconds. All other design characteristics remained the same.

The remaining sessions (session 5 and beyond until final test phase) began with a test of all 48 problems to determine if the subject was able to recall the problems with the desired level of response accuracy and speed (average RT of less than 1 s, and 100% correct). In the evaluation test, all 48 problems were presented individually (without answers), and the subject said the answer aloud. Reaction times were measured using a voice activated computer relay. No feedback was provided, and there was no study or review. At the end of the 48 problems, the computer software provided the percent correct and an average (correct trial) latency. If the subject reached the performance criterion (100% correct and RT average of less than 1 s), the
subject was tested in a similar manner on 4 additional blocks of all 48 problems, for a final test
database of 5 trials per problem per subject. If the subject did not achieve sufficient speed and
accuracy, the training procedure (used for the third and fourth sessions) was repeated and
subjects returned the next day to try again to reach the performance criterion.

Subjects and Apparatus

Twelve college undergraduates participated in the experiment. Seven of the subjects
were tested at Cleveland State University, and five were tested at Johns Hopkins University.
Subjects had normal or corrected to normal vision and reported proficiency in arithmetic fact
retrieval and normal acquisition of arithmetic facts during elementary grades. Subjects were
paid $2 per session, and a $15 bonus for completing the experiment, and required 10 to 22
sessions to complete the experiment.

The Psyscope program and button box (Cohen, MacWhinney, Flatt & Provost, 1993)
was used to present stimuli, record responses, and provide feedback using a Macintosh
computer. All responses were verbal and triggered the Psyscope ButtonBox voice activated
relay.

Results

Competition Effect

The main goal of the experiment is to study the effects of inter-problem competition on
the accuracy and latency of responses. According to the Semantic Network Retrieval Theory,
both response latencies and accuracies should be affected by the amount of inhibition imposed
by competing problems.

Final Test Reaction Times

Results indicate that there is an effect of problem neighborhood density on final RTs.
Using the set of final test RTs, a median correct reaction time was calculated for each problem
for each subject. These median reaction times were averaged across problems to provide a measure of performance for problems with small and large competition values, and small and large operands for each subject, allowing a 2 x 2 repeated measure ANOVA. There was a significant effect of inter-problem-node competition on RTs. Reaction times were significantly faster for problems with low competition (853 ms) than high competition problems (887 ms; F(1,11)=6.69; \( p < .05 \)). There was also a significant effect of operand size. Problems with small operands were solved more quickly (846 ms) than those with larger operands (890 ms). No competition x operand size interaction was found (F(3,11)=1.12, \( p > .05 \)). Final test performance was nearly error free (0.9% errors), and so no analysis of final test errors was performed.

It is important to note that the competition effect can not be attributed to other effects previously reported in the cognitive arithmetic literature. Competition values were uncorrelated with both operand size and answer size. Each problem was trained equally often, and was presented in each section, so the competition effect can not be attributed to either problem frequency or the order of problem presentation.

Evaluation Test Reaction Times and Error Rates

In addition to considering the final reaction times, we may also consider RT performance in the evaluation tests preceding the subject's reaching the performance criterion (of 100% correct, RTs less than 1 s). The evaluation tests are identical to the final tests, except that the subjects did not reach the performance criterion. Evaluations begin at a relatively early stage of learning, and to ensure that arithmetic facts were adequately learned for competition to be measured, only evaluation tests from the latter half of each subject's training were used in this analysis. Median RTs were retrieved for each subject for each problem, and then averaged into large or small competition values for each subject.

Results from the evaluation tests reveal effects consistent with the final test results, but with somewhat smaller differences in performance. RTs during training were found to be faster
for problems with smaller competition values (1213 ms) than those with larger competition values (1235 ms). Error rates were only slightly smaller for problems with smaller competition values (8% errors) than those with larger competition values (9%). One reason for the smaller differences in the training may relate to the fact that some subjects (e.g., Subject 7) appear to have adopted a speed-accuracy trade off in which they actually respond faster to problems with high competition levels (1352ms vs. 1260ms for high and low competition groups), but make more errors on the high competition group (10% vs. 7% errors). These speed-accuracy trade offs were not possible in the final testing conditions because subjects had to maintain very high levels of accuracy.

In summary, results appear generally consistent with the predictions of the semantic network retrieval theory. Final RTs were found to be influenced by variations in competition. Problems with more competition were also found to be slower during training, and slightly more error prone.

**Replication of Previous Findings**

An important consideration in the study of an artificial arithmetic operation is if the results from the artificial arithmetic operation bear resemblance to normal arithmetic performance, suggesting that similar cognitive processes are utilized for both arithmetic operations. For this reason, the diamond arithmetic data are examined below for some of the major effects reported in normal arithmetic.

**Problem Size Effect**

As was found in previous studies, there appears to be a strong relationship between operand size and reaction time. As depicted in Figure 22, there is a clear relationship between problem family (e.g., 2's problems, 3's problems) and final RT ($R^2=.749; p<.01$). As was found with previous artificial arithmetic experiments, RTs did not correlate significantly with answer size ($R^2=.049; p>.05$).
Interestingly, as in normal arithmetic, in this experiment both a 5's effect, and ties effect were found. As can be seen in Figure 22, 5's problems were on average considerably faster than might be expected relative to other problem families. In addition, there was also a strong ties effect. Problems with identical operands (e.g., 4 ÷ 4) were solved considerably faster (841 ms) than non-tie problems (882 ms).

**Error Types**

The errors types and frequencies found in previous artificial arithmetic operations have largely been replicated in this experiment. In Table 26 is the breakdown of error types and frequencies. As can be seen the most frequent type of error is the operand error, followed in frequency by table errors, non-table errors, and operation errors.
Table 26: Proportions of different error types in the diamond arithmetic experiment.

These proportions of the different types of errors are quite similar to those reported in previous artificial arithmetic experiments (e.g., Whalen et al., 1996) and normal arithmetic (e.g., Campbell, 1985). This provides some evidence that similar retrieval mechanisms may have been used in both this experiment and normal multiplication.

**Simulation of Diamond Arithmetic Experiment**

The semantic network simulation was used to ensure that the predictions formed using normal error data approximate the predictions of the semantic network retrieval theory about RT and error rate effects found in the diamond arithmetic experiment. The simulation was only modified to accommodate 48 problems and answers rather than 64. Operand and answer representations, remained the same. What changed for each diamond arithmetic table was the connection weights between the problem nodes and the operands and answer nodes (since each table had a unique set of 48 problems, and answers). Other settings such as the strength of the problem node/answer connections and the strength of inter-problem-node inhibition were unchanged.

**Simulation Results**

The number of iterations required for the network to approximate (with less than 10% error) one of the output patterns for both the tens and ones values was used as a pseudoRT measure for predicting performance. As in the normal case, median RTs for each subject were
averaged across low and high competition values and small and large operands to produce a 2 x 2 repeated measures ANOVA. Each subject's diamond arithmetic table was simulated 10 times.

The two basic effects reported in the actual experiment: a competition effect, and operand size effect, were also found in the simulation of the diamond arithmetic experiment. The simulation was found to require significantly more iterations to correctly retrieve answers for problems with more competition (56.7 iterations) than for the problems with low competition values (51.5 iterations; F(1,11) = 31.3, p<.01). The simulation also produced a somewhat stronger effect of problem size. Problems with larger operands were found to take significantly more iterations to settle on an answer (57.9 iterations) than problems with smaller operands (50.3 iterations; F(1,11) = 80, p<.01). No interaction between competition values and operand size and was found.

Error rates from the simulation also reveal an effect of inter problem competition. Problems with stronger competition had an error rate of 7%, while the problems with smaller competition values an error rate of only 3%. Interestingly the overall error rate was somewhat smaller (5%) than the error rate found in the simulation of multiplication. This effect can likely be attributed to the overall lowering of the problem competition values in comparison to normal arithmetic.

The simulation also produced a number of the common effects reported in normal arithmetic. The types and frequencies of errors were generally consistent with previous findings: operand errors were the most common (84%), followed in frequency by table errors (9%), and non-table errors (7%). The simulation produced more operand errors and fewer table errors than are reported in the actual diamond arithmetic experiment, but the relative frequencies of the different error types remain similar. The simulation also produced an operand distance effect. The majority of operand errors had an operand distance of 2 or less.
In summary, the simulation appears to produce effects which are consistent both with the findings from the actual diamond arithmetic experiment, and more generally with previous studies of arithmetic RTs and errors.

**Discussion**

This experiment provides evidence that is consistent with the predictions of the semantic network retrieval theory. Problems with stronger competition from neighboring problems had longer RTs, and were somewhat more error prone than problems with less competition. This pattern of performance suggests that several problem nodes with operands close in magnitude to the correct operands compete to be retrieved from memory. The basic effects found in the experimental data were largely mimicked by the simulation: RTs and error rates in the simulation were higher for problems with greater competition than those with less competition.

As with other artificial arithmetic experiments, questions arise as to how much the findings from an artificial arithmetic operation may be related to normal arithmetic. In this regard, one must consider the similarities between normal and diamond arithmetic operations, both in terms of design and results. In terms of the design of diamond arithmetic, there are a number of similarities with normal arithmetic operations. First, diamond arithmetic required subjects to learn a substantial number of arithmetic facts: 48, which is presumably sufficient to elicit retrieval competition. Second the diamond arithmetic facts resemble normal arithmetic in that the operands are the numerals 2-9 and the answers are numerals between 23 and 99. Perhaps the biggest differences between diamond arithmetic and normal arithmetic is the fact that answers and operands were uncorrelated, and there were no non-retrieval strategies available to 'work out' the answers to diamond arithmetic problems. Thus, subjects could only retrieve diamond arithmetic facts from memory. However, despite the lack of non-retrieval strategies, both the current and artificial arithmetic experiments (e.g., Whalen et al., 1996; Harley, 1990) revealed reaction time and error phenomena which were very similar to normal arithmetic.
Diamond arithmetic performance (like 'pelification'; Harley, 1990; Whalen et al., 1996) had a number of key similarities with normal arithmetic. Perhaps most importantly, subjects were trained with sufficient practice to approximate normal arithmetic skill (100% correct), and RTs (averaging less than 1s). This level of performance approximates the level of skill, and perhaps the strength of the associations found in normal arithmetic. In addition, the diamond arithmetic experiment replicated a number of major phenomena reported in normal arithmetic, such as the types and frequencies of errors reported, and a problem size effect relative to the operands of the problem. In summary, there appear to be several commonalities in both performance and design between normal and diamond arithmetic. For this reason it is reasonable to associate the performance in the diamond arithmetic experiment with normal arithmetic, suggesting that in normal arithmetic the same types of inter-problem-node inhibition may influence performance.
VII: GENERAL DISCUSSION

This dissertation consists of two experiments and a network simulation which were used to evaluate current theories of arithmetic fact retrieval and the newly introduced Semantic Network Retrieval Theory. The first experiment evaluated the phonological storage hypothesis: the notion that arithmetic facts are stored and retrieved in a phonological form. The pattern of performance exhibited by brain-damaged patient KSR was found to be inconsistent with this hypothesis. The second experiment was designed to evaluate the predictions of a proposed theory of arithmetic fact retrieval introduced and simulated in this dissertation: the Semantic Network Retrieval Theory. This theory accounts for previous evidence as well as new findings from an artificial arithmetic operation which provided support for this hypothesis.

This chapter includes a brief summary of the experimental evidence reported in the dissertation, and then explores how the newly introduced Semantic Network Retrieval Theory might be further developed. Included in the discussion of the SNRT theory is both consideration of arithmetic phenomena that the theory was unable to account for, and other possible influences on arithmetic fact retrieval which are beyond the scope of this dissertation but constitute important theoretical questions for future research.

Evaluating the Phonological Storage Hypothesis: Patient KSR

The first study evaluated the notion that arithmetic facts may be stored and retrieved in a phonological form. This phonological storage hypothesis makes the prediction that the initial representation of a problem must be converted into the correct phonological form in order to retrieve the correct arithmetic fact from memory. Brain-damaged patient KSR's pattern of performance was found to be inconsistent with this hypothesis. KSR was able to retrieve arithmetic facts from memory even when unable to say the problem aloud (which presumably requires the problem to be converted into phonological form). For example, when KSR was presented the problem $6 \times 3$ in arabic form, he said the problem as "four times eight", yet wrote
the correct answer, 18. KSR's errors in saying the problem aloud were studied for several
types of errors (e.g., speech production errors, perseveratory responses), but these possible
error types could not account for several spoken errors. The majority of errors in saying the
answer aloud were hypothesized to be due to erroneous conversions of the problem into
phonological form. According to the phonological storage hypothesis, when KSR was unable
to convert the problem to phonological form, he should be unable to retrieve the correct
arithmetic fact from memory. However, contrary to this prediction, KSR correctly retrieved the
correct arithmetic fact from memory for 98% of trials when unable to convert the problem into a
phonological representation. This pattern indicates that KSR is not retrieving arithmetic facts
from memory using a phonological form of the problem and therefore, that arithmetic facts are
not stored exclusively in a phonological form.

While KSR's pattern of performance was inconsistent with the notion that arithmetic
facts are stored exclusively in phonological form, an alternate hypothesis is that arithmetic facts
are stored in an abstract semantic form (McCloskey, 1992) as proposed under the SNRT
theory. This theory could account for KSR's pattern of performance by assuming that
impairment was limited to converting semantic number representations into their phonological
forms (resulting in impairments in naming numerals aloud and saying answers to arithmetic
facts). Other processes such as arithmetic fact retrieval, and arabic numeral comprehension and
production remained unimpaired, allowing KSR to successfully write the answer to arithmetic
problems presented in arabic form.

Another hypothesis is that arithmetic facts are stored not in a phonological form, but in a
more abstract level of the phonological production process (Dehaene, 1995). Dehaene suggests
that arithmetic facts are retrieved within a planning stage of phonological production in which
syntactic slots for spoken production are present, but in which the phonological forms have not
yet been inserted. This appears to be a viable alternative, but one that clearly abandons the
notion that arithmetic facts are stored in a phonological form. A final possibility is that
arithmetic facts may be stored in several different forms. KSR's pattern of performance indicates that arithmetic facts are not stored exclusively in phonological form. However, it does not preclude the possibility that arithmetic facts are stored in many different format specific codes (e.g., Campbell, 1996). Given that no fully specified multi-format proposal has been produced, this possibility remains difficult to evaluate.

Among the theories described above which might successfully account for KSR's pattern of performance, only the semantic network retrieval theory provides a sufficiently detailed account of arithmetic fact retrieval to generate predictions about the major arithmetic RT and error phenomena. For this reason, the predictions of the semantic network retrieval theory were further explored.

The Semantic Network Retrieval Theory

The second experiment presented in this dissertation was designed to evaluate the semantic network retrieval theory. The SNRT theory assumes that arithmetic facts are retrieved using a semantic magnitude representation from a network of stored arithmetic facts. When a problem is presented several arithmetic facts become activated, and compete to be selected. The amount of activation each fact receives depends upon the similarity between the magnitude representation of its operands, and the magnitude representation of the actual operands. The closer the representations are, the more active that fact will become.

According to the SNRT theory, a problem with fewer neighboring problems with operands close in magnitude to its own will have comparatively less competition during retrieval than problems with more neighboring problems, resulting in RT and error rate benefits for problems with fewer competitors. This prediction was tested by training normal subjects on an artificial arithmetic operation which varied the amount of competition problems received. Results were consistent with the predictions of the SNRT theory. Problems with greater inter-problem competition were responded to more slowly and had higher error rates than problems with comparatively less competition.
Semantic Numeral Representations

In this dissertation I argue that semantic numeral representations have two fundamental properties: (1) representations of similar magnitudes are more similar to one another than representations of dissimilar magnitudes, and (2) the larger the magnitude being represented, the more similar the representation of that number is to other representations of similar magnitudes.

Several aspects of normal and artificial arithmetic suggest that arithmetic facts are retrieved using a magnitude representation in which the representations of similar magnitudes share more features than representations of dissimilar magnitudes. For example, in normal and artificial arithmetic, errors tend to have operands which are close in magnitude to the correct problem's operands. This operand distance effect is highly consistent across studies, and is reported even when the answers to problems are very different as in the pelification and diamond arithmetic experiments. Another source of evidence suggesting arithmetic fact retrieval involves magnitude representations of the operands is the diamond arithmetic experiment reported in this dissertation. Problems with operands close to the correct operands were found to provide more competition to the correct problem node than other problems with more distant operands. This is consistent with the notion that the representations of numerical close operands share common features, which results in the spreading of activation not just to the correct problem node, but also to other problem nodes with numerically close problem nodes.

However, there is less evidence to suggest that the larger a magnitude representation is, the more similar its representation is to the representations of other similar magnitudes. In this dissertation I argue that the problem size effect found in normal and artificial arithmetic can be attributed to this aspect of semantic number representations. Representations of larger numbers are hypothesized to be more similar with one another than representations of smaller numbers are with one another, resulting in increased competition during arithmetic fact retrieval, which ultimately results in increases in RT and error rates for larger problems.
Clearly, these aspects of semantic representations of magnitude provide only a rough sketch of what magnitude representations may actually be like. Future study will be needed to develop our understanding of semantic representations of magnitude.

**Theoretical Elaboration**

There remain certain phenomena which are not accounted for, and possible generalizations of the theory which have not yet been explored. Below are some aspects of arithmetic fact retrieval which the Semantic Network Retrieval Theory will have to address, beginning with the effects considered within this dissertation, and then turning to the issues outside the scope of the current formulation of the SNRT theory.

**Unexplained Phenomena**

While the SNRT theory appears to account for all major phenomena reported in the arithmetic literature, other effects remain unaccounted for. First, in normal arithmetic problems from the 5's family (e.g., 4 x 5 = 20) are solved much more quickly than would be expected based on the RTs from other problem families (e.g., 4's problems, 6's problems). Neither the semantic network retrieval theory, nor the semantic network simulation predicts this effect. It is currently unclear what this effect might be attributed to. However, there are some hypotheses which may lie outside the framework of the semantic network retrieval theory.

The 'fives effect' appears to be unique to multiplication. Studies of RT patterns in addition do not show similar benefits for 5's problems relative to other problem families. One of the most unique aspect of the 5's problems in multiplication is that the answers often consist of single word response answers, and answers (e.g., twenty, thirty, forty), while other answers all end with a five. These answer characteristics are unique to the fives problems and may be responsible for the differences in 5's problem performance.

Another finding which was not predicted by the semantic network retrieval theory is the finding that tie problems are generally faster than non-tie problems. One possibility is that the
transcoding from the initial representation (e.g., arabic numerals) to a magnitude representation
requires less time when both operands are the same. This is another question for future research.

More generally, these effects may be the result of effects which lie outside the scope of
an arithmetic fact retrieval theory. For example, fives and ties problems may be more salient
due to the numbers involved, or may be presented somewhat more frequently than other
problems. Nevertheless, these effects currently remain unexplained and an issue for future
consideration.

Future Questions

In addition to the issues which were considered in the dissertation, there remain
additional questions which need to be addressed in order to provide a more comprehensive
theory of arithmetic fact retrieval.

Accounting for Multiple Arithmetic Operations

Another challenge for the semantic network theory is to broaden its ability to account not
just for multiplication, but also for all arithmetic operations. One focus of current research is to
determine whether or not the same representations are involved in reciprocal operations, such as
multiplication and division (e.g., Campbell, 1996; Rickard, Healy, & Bourne, 1994). There
appear to be at least two possible ways of integrating subtraction and division into the current
theoretical model. One is to assume that subtraction and division require their own problem
representations, and that the problem nodes for these operands have weights which are
configured to allow the two presented operands (including one operand with both a tens and
ones representation) to activate the appropriate problem nodes which in turn will give rise to the
appropriate subtraction or division answer.

A second alternative is to assume that reciprocal operations use the same problem nodes
to retrieve arithmetic facts from memory (e.g., 8 x 6 = 48 and 48 / 8 activate the same problem
node). The only difference in arithmetic fact retrieval between the complementary operations is which magnitude representations are considered to be problem representations and which are considered answer representations. Currently the empirical evidence is mixed regarding the notion of same or different fact representations for different operations. This is an area where both empirical evidence, and further simulations will be required may provide more insight into the relationships between different operations.

**Non-Retrieval Strategy Use**

One issue not addressed in this dissertation is how often adults solve normal arithmetic problems by resorting to non-retrieval strategies. The frequency of non-retrieval strategy use continues to be an important consideration because current arithmetic fact retrieval theories (including the semantic network retrieval theory) assume that virtually all problems are solved using arithmetic fact retrieval and thus that reaction time and error phenomena in speeded arithmetic may be attributed to arithmetic fact retrieval. It is very likely that adults sometimes solve simple arithmetic problems through non-retrieval strategies. However, the frequency of strategy use, and the influence strategies have on overall RT and error rates are currently unknown. Thus determining how non-retrieval strategies might affect overall reaction time and error rates, are important issues for theories of arithmetic fact retrieval. If strategies play a significant role in adult performance, current arithmetic fact retrieval theories might need to be significantly modified.

**Multiple Arithmetic Fact Representations?**

The possibility exists that arithmetic facts may be stored in more than one form, including format specific forms. While this dissertation provides evidence in support of the notion that semantic representations of number are involved in arithmetic fact retrieval, this does not rule out the possibility that other arithmetic fact representations also exist. One possibility is that other forms of the facts do exist, but typically have only a minor role in arithmetic fact
retrieval. Unfortunately, current multiple arithmetic fact positions are not sufficiently formulated to generate testable predictions, or suggest possible modifications of current single format views. Thus an important consideration for future research is to determine if arithmetic facts may be represented in more than one form or code.

**Conclusion**

This dissertation sought to bring together several lines of inquiry to evaluate the notion that arithmetic facts are stored and retrieved in a semantic form from a network of stored arithmetic facts. The network simulations, performance of brain-damaged patient KSR, and results from the artificial arithmetic experiment reported in this dissertation are all consistent with this hypothesis. This Semantic Network Retrieval Theory also appears capable of accounting for all major arithmetic phenomena, and is consistent with the previously hypothesized notion that adults have the ability to represent numerical magnitude.

While initial studies of arithmetic processes sought to study only very specific aspects of arithmetic processing (e.g., the problem size effect), current theories of number processing are increasing inclusionary, seeking to account for a broad spectrum of arithmetic and non-arithmetic numerical abilities, as well as the development of numerical abilities and the effects of acquired and developmental deficits. Perhaps the integration of these formerly circumscribed interests combined with the search for increasingly well defined theoretical perspectives and hypotheses will provide a sufficiently broad understanding of numerical cognition to allow for improvements in the teaching and retraining of numerical and arithmetic concepts, and more generally influence and advance theories of memory and related cognitive processes.
VIII: REFERENCES


APPENDIX 1: DIAMOND ARITHMETIC TABLES

Reported below are the results of the diamond arithmetic experiment. Each table entry has (a) the answer to the diamond arithmetic problem, (b) the final test median RT, and (c) the competition value associated with that problem. Blanks in the table denote problems that were never trained or tested.

Subject 1

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>140</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>73</td>
<td>67</td>
<td>58</td>
<td>43</td>
<td>51</td>
<td>74</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>585</td>
<td>626</td>
<td>619</td>
<td>791</td>
<td>939</td>
<td>1252</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>305</td>
<td>325</td>
<td>441</td>
<td>481</td>
<td>517</td>
<td>325</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>67</td>
<td>52</td>
<td>29</td>
<td>87</td>
<td>89</td>
<td>57</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>622</td>
<td>781</td>
<td>812</td>
<td>854</td>
<td>867</td>
<td>976</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>366</td>
<td>554</td>
<td>572</td>
<td>594</td>
<td>594</td>
<td>325</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>83</td>
<td>52</td>
<td>61</td>
<td>91</td>
<td>39</td>
<td>82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>975</td>
<td>786</td>
<td>744</td>
<td>1175</td>
<td>1195</td>
<td>923</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>477</td>
<td>536</td>
<td>627</td>
<td>587</td>
<td>594</td>
<td>459</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>86</td>
<td>69</td>
<td>78</td>
<td>62</td>
<td>98</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>630</td>
<td>816</td>
<td>769</td>
<td>710</td>
<td>778</td>
<td>826</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>517</td>
<td>754</td>
<td>686</td>
<td>595</td>
<td>594</td>
<td>449</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>47</td>
<td>78</td>
<td>41</td>
<td>38</td>
<td>34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>896</td>
<td>1186</td>
<td>896</td>
<td>845</td>
<td>1116</td>
<td>1112</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>302</td>
<td>699</td>
<td>726</td>
<td>686</td>
<td>686</td>
<td>594</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>89</td>
<td>91</td>
<td>97</td>
<td>59</td>
<td>26</td>
<td>93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>898</td>
<td>907</td>
<td>848</td>
<td>725</td>
<td>940</td>
<td>955</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>554</td>
<td>627</td>
<td>686</td>
<td>627</td>
<td>594</td>
<td>477</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>74</td>
<td>68</td>
<td>62</td>
<td>34</td>
<td>26</td>
<td>46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>981</td>
<td>979</td>
<td>854</td>
<td>940</td>
<td>886</td>
<td>714</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>366</td>
<td>511</td>
<td>554</td>
<td>514</td>
<td>554</td>
<td>366</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>53</td>
<td>57</td>
<td>82</td>
<td>23</td>
<td>46</td>
<td>71</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>890</td>
<td>848</td>
<td>1016</td>
<td>945</td>
<td>769</td>
<td>615</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>327</td>
<td>442</td>
<td>400</td>
<td>334</td>
<td>325</td>
<td>305</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Subject 2

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>87</td>
<td>59</td>
<td>38</td>
<td>89</td>
<td>93</td>
<td>52</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>626</td>
<td>649</td>
<td>469</td>
<td>1167</td>
<td>1571</td>
<td>747</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>220</td>
<td>344</td>
<td>419</td>
<td>350</td>
<td>424</td>
<td>269</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>83</td>
<td>46</td>
<td>62</td>
<td>51</td>
<td>58</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>539</td>
<td>1077</td>
<td>794</td>
<td>857</td>
<td>817</td>
<td>596</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>471</td>
<td>536</td>
<td>594</td>
<td>576</td>
<td>529</td>
<td>424</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>94</td>
<td>68</td>
<td>39</td>
<td>78</td>
<td>91</td>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1160</td>
<td>753</td>
<td>680</td>
<td>1113</td>
<td>1201</td>
<td>1129</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>686</td>
<td>686</td>
<td>752</td>
<td>704</td>
<td>576</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>38</td>
<td>46</td>
<td>39</td>
<td>23</td>
<td>79</td>
<td>82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>474</td>
<td>791</td>
<td>762</td>
<td>722</td>
<td>825</td>
<td>1561</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>504</td>
<td>517</td>
<td>722</td>
<td>657</td>
<td>641</td>
<td>302</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>89</td>
<td>78</td>
<td>23</td>
<td>71</td>
<td>98</td>
<td>53</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>630</td>
<td>1142</td>
<td>760</td>
<td>795</td>
<td>961</td>
<td>1103</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>507</td>
<td>664</td>
<td>600</td>
<td>740</td>
<td>677</td>
<td>554</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>37</td>
<td>62</td>
<td>91</td>
<td>98</td>
<td>34</td>
<td>86</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>688</td>
<td>878</td>
<td>1081</td>
<td>993</td>
<td>762</td>
<td>932</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>526</td>
<td>594</td>
<td>645</td>
<td>560</td>
<td>536</td>
<td>441</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>93</td>
<td>51</td>
<td>76</td>
<td>34</td>
<td>73</td>
<td>57</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2578</td>
<td>744</td>
<td>670</td>
<td>756</td>
<td>739</td>
<td>1045</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>366</td>
<td>511</td>
<td>477</td>
<td>554</td>
<td>511</td>
<td>442</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>58</td>
<td>41</td>
<td>82</td>
<td>31</td>
<td>86</td>
<td>43</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>801</td>
<td>845</td>
<td>1177</td>
<td>660</td>
<td>1139</td>
<td>601</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>402</td>
<td>441</td>
<td>495</td>
<td>432</td>
<td>419</td>
<td>287</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Subject 3

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>53</td>
<td>71</td>
<td>68</td>
<td>73</td>
<td>59</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>584</td>
<td>921</td>
<td>609</td>
<td>559</td>
<td>764</td>
<td>665</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>220</td>
<td>400</td>
<td>495</td>
<td>507</td>
<td>467</td>
<td>424</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>51</td>
<td>26</td>
<td>82</td>
<td>47</td>
<td>86</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>565</td>
<td>622</td>
<td>640</td>
<td>718</td>
<td>609</td>
<td>1127</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>344</td>
<td>471</td>
<td>517</td>
<td>576</td>
<td>529</td>
<td>424</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>94</td>
<td>83</td>
<td>39</td>
<td>57</td>
<td>91</td>
<td>58</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1094</td>
<td>1036</td>
<td>599</td>
<td>730</td>
<td>689</td>
<td>738</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>536</td>
<td>686</td>
<td>664</td>
<td>645</td>
<td>517</td>
<td>459</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>68</td>
<td>26</td>
<td>39</td>
<td>79</td>
<td>46</td>
<td>67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>641</td>
<td>675</td>
<td>651</td>
<td>808</td>
<td>716</td>
<td>741</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>419</td>
<td>594</td>
<td>686</td>
<td>619</td>
<td>528</td>
<td>477</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>73</td>
<td>57</td>
<td>79</td>
<td>23</td>
<td>34</td>
<td>41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>539</td>
<td>689</td>
<td>694</td>
<td>693</td>
<td>747</td>
<td>608</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>350</td>
<td>752</td>
<td>657</td>
<td>762</td>
<td>579</td>
<td>410</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>82</td>
<td>62</td>
<td>46</td>
<td>34</td>
<td>93</td>
<td>38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>569</td>
<td>1032</td>
<td>742</td>
<td>787</td>
<td>1113</td>
<td>783</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>576</td>
<td>704</td>
<td>649</td>
<td>677</td>
<td>495</td>
<td>419</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>47</td>
<td>91</td>
<td>52</td>
<td>93</td>
<td>37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>736</td>
<td>1099</td>
<td>728</td>
<td>846</td>
<td>1712</td>
<td>731</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>424</td>
<td>529</td>
<td>576</td>
<td>554</td>
<td>536</td>
<td>511</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>89</td>
<td>86</td>
<td>67</td>
<td>38</td>
<td>87</td>
<td>76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>602</td>
<td>1613</td>
<td>718</td>
<td>799</td>
<td>624</td>
<td>551</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>269</td>
<td>424</td>
<td>302</td>
<td>441</td>
<td>442</td>
<td>287</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Subject 4

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>47</td>
<td>73</td>
<td>89</td>
<td>38</td>
<td>97</td>
<td>41</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>677</td>
<td>838</td>
<td>725</td>
<td>1247</td>
<td>1557</td>
<td>663</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>287</td>
<td>360</td>
<td>342</td>
<td>477</td>
<td>366</td>
<td>305</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>59</td>
<td>62</td>
<td>57</td>
<td>29</td>
<td>86</td>
<td>76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>830</td>
<td>750</td>
<td>1199</td>
<td>984</td>
<td>853</td>
<td>1022</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>442</td>
<td>511</td>
<td>554</td>
<td>517</td>
<td>536</td>
<td>325</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>73</td>
<td>57</td>
<td>82</td>
<td>53</td>
<td>39</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>998</td>
<td>1024</td>
<td>864</td>
<td>767</td>
<td>1555</td>
<td>728</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>382</td>
<td>594</td>
<td>675</td>
<td>712</td>
<td>686</td>
<td>554</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>98</td>
<td>82</td>
<td>93</td>
<td>71</td>
<td>67</td>
<td>87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>903</td>
<td>802</td>
<td>761</td>
<td>796</td>
<td>987</td>
<td>995</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>631</td>
<td>657</td>
<td>734</td>
<td>766</td>
<td>572</td>
<td>334</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>29</td>
<td>53</td>
<td>71</td>
<td>78</td>
<td>34</td>
<td>79</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1110</td>
<td>802</td>
<td>743</td>
<td>673</td>
<td>1236</td>
<td>1252</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>572</td>
<td>649</td>
<td>843</td>
<td>771</td>
<td>617</td>
<td>631</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>38</td>
<td>86</td>
<td>43</td>
<td>34</td>
<td>91</td>
<td>58</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1031</td>
<td>768</td>
<td>1339</td>
<td>1370</td>
<td>638</td>
<td>1112</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>360</td>
<td>554</td>
<td>627</td>
<td>609</td>
<td>594</td>
<td>441</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>97</td>
<td>23</td>
<td>67</td>
<td>91</td>
<td>26</td>
<td>94</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1150</td>
<td>584</td>
<td>863</td>
<td>734</td>
<td>815</td>
<td>515</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>366</td>
<td>495</td>
<td>495</td>
<td>554</td>
<td>511</td>
<td>442</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>41</td>
<td>76</td>
<td>52</td>
<td>37</td>
<td>58</td>
<td>68</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>643</td>
<td>1186</td>
<td>1273</td>
<td>888</td>
<td>1148</td>
<td>554</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>364</td>
<td>366</td>
<td>419</td>
<td>334</td>
<td>360</td>
<td>287</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>2</td>
<td>38</td>
<td>97</td>
<td>78</td>
<td>34</td>
<td>39</td>
<td>87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>633</td>
<td>725</td>
<td>639</td>
<td>617</td>
<td>653</td>
<td>1067</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>220</td>
<td>344</td>
<td>419</td>
<td>350</td>
<td>424</td>
<td>269</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>94</td>
<td>53</td>
<td>41</td>
<td>98</td>
<td>47</td>
<td>68</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>690</td>
<td>726</td>
<td>603</td>
<td>578</td>
<td>619</td>
<td>599</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>471</td>
<td>536</td>
<td>594</td>
<td>576</td>
<td>529</td>
<td>424</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>73</td>
<td>67</td>
<td>86</td>
<td>57</td>
<td>31</td>
<td>89</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1108</td>
<td>647</td>
<td>816</td>
<td>886</td>
<td>920</td>
<td>659</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>400</td>
<td>686</td>
<td>686</td>
<td>752</td>
<td>704</td>
<td>576</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>78</td>
<td>41</td>
<td>86</td>
<td>29</td>
<td>46</td>
<td>82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>663</td>
<td>624</td>
<td>637</td>
<td>674</td>
<td>775</td>
<td>603</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>504</td>
<td>517</td>
<td>722</td>
<td>657</td>
<td>641</td>
<td>302</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>34</td>
<td>57</td>
<td>29</td>
<td>71</td>
<td>83</td>
<td>26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>648</td>
<td>819</td>
<td>731</td>
<td>706</td>
<td>671</td>
<td>839</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>507</td>
<td>664</td>
<td>600</td>
<td>740</td>
<td>677</td>
<td>554</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>23</td>
<td>98</td>
<td>31</td>
<td>83</td>
<td>91</td>
<td>58</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>777</td>
<td>762</td>
<td>743</td>
<td>650</td>
<td>938</td>
<td>724</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>39</td>
<td>47</td>
<td>52</td>
<td>91</td>
<td>93</td>
<td>51</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>693</td>
<td>1277</td>
<td>565</td>
<td>783</td>
<td>572</td>
<td>814</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>366</td>
<td>511</td>
<td>477</td>
<td>554</td>
<td>511</td>
<td>442</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>68</td>
<td>62</td>
<td>82</td>
<td>43</td>
<td>58</td>
<td>37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>732</td>
<td>1236</td>
<td>978</td>
<td>925</td>
<td>1220</td>
<td>789</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>402</td>
<td>441</td>
<td>495</td>
<td>432</td>
<td>419</td>
<td>287</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Subject 6

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>76</td>
<td>62</td>
<td>39</td>
<td>78</td>
<td>43</td>
<td>87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>567</td>
<td>693</td>
<td>719</td>
<td>774</td>
<td>678</td>
<td>540</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>220</td>
<td>400</td>
<td>437</td>
<td>409</td>
<td>424</td>
<td>269</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>97</td>
<td>52</td>
<td>46</td>
<td>98</td>
<td>57</td>
<td>41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1346</td>
<td>1577</td>
<td>498</td>
<td>391</td>
<td>1359</td>
<td>618</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>344</td>
<td>471</td>
<td>517</td>
<td>576</td>
<td>529</td>
<td>424</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>91</td>
<td>29</td>
<td>37</td>
<td>93</td>
<td>51</td>
<td>67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1220</td>
<td>1745</td>
<td>475</td>
<td>984</td>
<td>1258</td>
<td>562</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>594</td>
<td>686</td>
<td>730</td>
<td>752</td>
<td>704</td>
<td>576</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>39</td>
<td>46</td>
<td>37</td>
<td>91</td>
<td>38</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>628</td>
<td>516</td>
<td>436</td>
<td>818</td>
<td>1411</td>
<td>808</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>671</td>
<td>649</td>
<td>649</td>
<td>649</td>
<td>302</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>86</td>
<td>93</td>
<td>71</td>
<td>83</td>
<td>34</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>952</td>
<td>1847</td>
<td>543</td>
<td>754</td>
<td>492</td>
<td>995</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>594</td>
<td>686</td>
<td>752</td>
<td>771</td>
<td>686</td>
<td>554</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>31</td>
<td>98</td>
<td>68</td>
<td>34</td>
<td>94</td>
<td>82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1538</td>
<td>424</td>
<td>1021</td>
<td>569</td>
<td>949</td>
<td>383</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>427</td>
<td>554</td>
<td>569</td>
<td>600</td>
<td>536</td>
<td>441</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>43</td>
<td>67</td>
<td>59</td>
<td>94</td>
<td>47</td>
<td>53</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>696</td>
<td>366</td>
<td>714</td>
<td>2267</td>
<td>874</td>
<td>963</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>366</td>
<td>495</td>
<td>495</td>
<td>554</td>
<td>511</td>
<td>442</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>87</td>
<td>41</td>
<td>73</td>
<td>89</td>
<td>82</td>
<td>58</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>696</td>
<td>785</td>
<td>852</td>
<td>1278</td>
<td>1455</td>
<td>236</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>364</td>
<td>366</td>
<td>419</td>
<td>334</td>
<td>360</td>
<td>287</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Subject 7

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>37</td>
<td>52</td>
<td>98</td>
<td>86</td>
<td>59</td>
<td>83</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>718</td>
<td>640</td>
<td>811</td>
<td>691</td>
<td>1090</td>
<td>798</td>
<td></td>
<td></td>
</tr>
<tr>
<td>220</td>
<td>342</td>
<td>427</td>
<td>485</td>
<td>366</td>
<td>305</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>41</td>
<td>46</td>
<td>51</td>
<td>94</td>
<td>58</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>586</td>
<td>711</td>
<td>815</td>
<td>1099</td>
<td>997</td>
<td>838</td>
<td></td>
<td></td>
</tr>
<tr>
<td>344</td>
<td>471</td>
<td>594</td>
<td>558</td>
<td>594</td>
<td>325</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>87</td>
<td>31</td>
<td>78</td>
<td>71</td>
<td>57</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>795</td>
<td>567</td>
<td>673</td>
<td>1409</td>
<td>784</td>
<td>775</td>
<td></td>
<td></td>
</tr>
<tr>
<td>536</td>
<td>686</td>
<td>744</td>
<td>771</td>
<td>686</td>
<td>554</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>98</td>
<td>46</td>
<td>31</td>
<td>47</td>
<td>79</td>
<td>82</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>976</td>
<td>911</td>
<td>696</td>
<td>874</td>
<td>1024</td>
<td>1130</td>
<td></td>
<td></td>
</tr>
<tr>
<td>477</td>
<td>594</td>
<td>635</td>
<td>667</td>
<td>572</td>
<td>334</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>78</td>
<td>47</td>
<td>29</td>
<td>38</td>
<td>39</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1097</td>
<td>1010</td>
<td>981</td>
<td>775</td>
<td>1115</td>
<td>1383</td>
<td></td>
<td></td>
</tr>
<tr>
<td>449</td>
<td>645</td>
<td>619</td>
<td>740</td>
<td>600</td>
<td>631</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>94</td>
<td>62</td>
<td>38</td>
<td>91</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>737</td>
<td>649</td>
<td>775</td>
<td>879</td>
<td>735</td>
<td>776</td>
<td></td>
<td></td>
</tr>
<tr>
<td>467</td>
<td>576</td>
<td>520</td>
<td>579</td>
<td>594</td>
<td>441</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>76</td>
<td>57</td>
<td>91</td>
<td>23</td>
<td>97</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1373</td>
<td>1074</td>
<td>820</td>
<td>728</td>
<td>790</td>
<td>708</td>
<td></td>
<td></td>
</tr>
<tr>
<td>424</td>
<td>529</td>
<td>517</td>
<td>495</td>
<td>511</td>
<td>442</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>73</td>
<td>82</td>
<td>43</td>
<td>26</td>
<td>67</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1073</td>
<td>948</td>
<td>1043</td>
<td>903</td>
<td>788</td>
<td>1023</td>
<td></td>
<td></td>
</tr>
<tr>
<td>424</td>
<td>459</td>
<td>477</td>
<td>410</td>
<td>419</td>
<td>287</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Subject 8

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>37</td>
<td>52</td>
<td>57</td>
<td>91</td>
<td>43</td>
<td>76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>941</td>
<td>771</td>
<td>819</td>
<td>946</td>
<td>967</td>
<td>763</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>287</td>
<td>442</td>
<td>382</td>
<td>419</td>
<td>366</td>
<td>305</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>97</td>
<td>98</td>
<td>23</td>
<td>82</td>
<td>46</td>
<td>41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>896</td>
<td>640</td>
<td>730</td>
<td>899</td>
<td>1069</td>
<td>1243</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>511</td>
<td>594</td>
<td>631</td>
<td>594</td>
<td>594</td>
<td>325</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>57</td>
<td>98</td>
<td>79</td>
<td>38</td>
<td>59</td>
<td>73</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>668</td>
<td>635</td>
<td>689</td>
<td>1051</td>
<td>1019</td>
<td>807</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>360</td>
<td>554</td>
<td>716</td>
<td>726</td>
<td>686</td>
<td>495</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>78</td>
<td>79</td>
<td>34</td>
<td>47</td>
<td>31</td>
<td>87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>999</td>
<td>787</td>
<td>1043</td>
<td>810</td>
<td>697</td>
<td>1016</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>342</td>
<td>675</td>
<td>675</td>
<td>766</td>
<td>495</td>
<td>410</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>82</td>
<td>38</td>
<td>47</td>
<td>29</td>
<td>83</td>
<td>93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>921</td>
<td>920</td>
<td>713</td>
<td>696</td>
<td>1114</td>
<td>1163</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>517</td>
<td>712</td>
<td>766</td>
<td>749</td>
<td>569</td>
<td>374</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>91</td>
<td>46</td>
<td>59</td>
<td>83</td>
<td>62</td>
<td>58</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>762</td>
<td>1159</td>
<td>979</td>
<td>1310</td>
<td>1013</td>
<td>978</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>477</td>
<td>536</td>
<td>686</td>
<td>609</td>
<td>554</td>
<td>419</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>73</td>
<td>31</td>
<td>94</td>
<td>62</td>
<td>53</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>990</td>
<td>839</td>
<td>633</td>
<td>959</td>
<td>834</td>
<td>633</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>366</td>
<td>554</td>
<td>572</td>
<td>631</td>
<td>594</td>
<td>511</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>76</td>
<td>41</td>
<td>87</td>
<td>58</td>
<td>67</td>
<td>68</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>700</td>
<td>1266</td>
<td>814</td>
<td>1008</td>
<td>781</td>
<td>801</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>305</td>
<td>325</td>
<td>334</td>
<td>441</td>
<td>442</td>
<td>287</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Subject 9

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>83</td>
<td>39</td>
<td>43</td>
<td>37</td>
<td>94</td>
<td>71</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>801</td>
<td>747</td>
<td>855</td>
<td>726</td>
<td>1154</td>
<td>854</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>220</td>
<td>344</td>
<td>419</td>
<td>350</td>
<td>424</td>
<td>269</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>57</td>
<td>93</td>
<td>67</td>
<td>68</td>
<td>76</td>
<td>82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>831</td>
<td>726</td>
<td>1450</td>
<td>1079</td>
<td>1201</td>
<td>1063</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>471</td>
<td>536</td>
<td>594</td>
<td>576</td>
<td>529</td>
<td>424</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>89</td>
<td>58</td>
<td>31</td>
<td>79</td>
<td>51</td>
<td>87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>858</td>
<td>719</td>
<td>583</td>
<td>757</td>
<td>959</td>
<td>749</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>686</td>
<td>686</td>
<td>752</td>
<td>704</td>
<td>576</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>67</td>
<td>31</td>
<td>29</td>
<td>46</td>
<td>34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>725</td>
<td>1040</td>
<td>642</td>
<td>793</td>
<td>1351</td>
<td>714</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>504</td>
<td>517</td>
<td>722</td>
<td>657</td>
<td>641</td>
<td>302</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>37</td>
<td>79</td>
<td>29</td>
<td>98</td>
<td>59</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>719</td>
<td>739</td>
<td>822</td>
<td>718</td>
<td>838</td>
<td>1153</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>507</td>
<td>664</td>
<td>600</td>
<td>740</td>
<td>677</td>
<td>554</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>53</td>
<td>68</td>
<td>51</td>
<td>59</td>
<td>62</td>
<td>86</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2046</td>
<td>1258</td>
<td>836</td>
<td>824</td>
<td>1052</td>
<td>944</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>526</td>
<td>594</td>
<td>645</td>
<td>560</td>
<td>536</td>
<td>441</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>94</td>
<td>76</td>
<td>38</td>
<td>62</td>
<td>91</td>
<td>41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>979</td>
<td>1237</td>
<td>691</td>
<td>992</td>
<td>867</td>
<td>765</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>366</td>
<td>511</td>
<td>477</td>
<td>554</td>
<td>511</td>
<td>442</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>82</td>
<td>47</td>
<td>34</td>
<td>78</td>
<td>86</td>
<td>52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1034</td>
<td>1099</td>
<td>650</td>
<td>817</td>
<td>1181</td>
<td>992</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>402</td>
<td>441</td>
<td>495</td>
<td>432</td>
<td>419</td>
<td>287</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Subject 10

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>68</td>
<td>39</td>
<td>86</td>
<td>71</td>
<td>59</td>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>589</td>
<td>735</td>
<td>700</td>
<td>736</td>
<td>1237</td>
<td>865</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>220</td>
<td>344</td>
<td>485</td>
<td>449</td>
<td>526</td>
<td>366</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>46</td>
<td>83</td>
<td>29</td>
<td>98</td>
<td>57</td>
<td>76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>779</td>
<td>669</td>
<td>742</td>
<td>684</td>
<td>582</td>
<td>979</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>471</td>
<td>536</td>
<td>594</td>
<td>594</td>
<td>511</td>
<td>402</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>67</td>
<td>91</td>
<td>38</td>
<td>82</td>
<td>53</td>
<td>94</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>650</td>
<td>628</td>
<td>1010</td>
<td>611</td>
<td>1184</td>
<td>569</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>686</td>
<td>686</td>
<td>664</td>
<td>645</td>
<td>441</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>86</td>
<td>29</td>
<td>38</td>
<td>73</td>
<td>79</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>596</td>
<td>827</td>
<td>761</td>
<td>736</td>
<td>1069</td>
<td>618</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>437</td>
<td>517</td>
<td>722</td>
<td>609</td>
<td>477</td>
<td>495</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>71</td>
<td>82</td>
<td>73</td>
<td>97</td>
<td>34</td>
<td>87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>642</td>
<td>630</td>
<td>758</td>
<td>799</td>
<td>672</td>
<td>776</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>409</td>
<td>752</td>
<td>649</td>
<td>740</td>
<td>560</td>
<td>432</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>98</td>
<td>53</td>
<td>41</td>
<td>34</td>
<td>62</td>
<td>58</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>817</td>
<td>948</td>
<td>1070</td>
<td>639</td>
<td>682</td>
<td>778</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>576</td>
<td>704</td>
<td>641</td>
<td>677</td>
<td>554</td>
<td>419</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>57</td>
<td>51</td>
<td>93</td>
<td>62</td>
<td>37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>635</td>
<td>729</td>
<td>1535</td>
<td>1036</td>
<td>714</td>
<td>774</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>424</td>
<td>529</td>
<td>576</td>
<td>554</td>
<td>536</td>
<td>511</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>47</td>
<td>76</td>
<td>23</td>
<td>58</td>
<td>26</td>
<td>78</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1366</td>
<td>1024</td>
<td>813</td>
<td>818</td>
<td>717</td>
<td>838</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>269</td>
<td>424</td>
<td>302</td>
<td>441</td>
<td>442</td>
<td>287</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Subject 11

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>91</td>
<td>89</td>
<td>46</td>
<td>98</td>
<td>51</td>
<td>53</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>663</td>
<td>711</td>
<td>865</td>
<td>658</td>
<td>664</td>
<td>982</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>305</td>
<td>366</td>
<td>477</td>
<td>350</td>
<td>419</td>
<td>229</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>89</td>
<td>86</td>
<td>29</td>
<td>62</td>
<td>76</td>
<td>41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>717</td>
<td>803</td>
<td>808</td>
<td>1080</td>
<td>743</td>
<td>893</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>325</td>
<td>536</td>
<td>517</td>
<td>594</td>
<td>511</td>
<td>402</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>86</td>
<td>39</td>
<td>94</td>
<td>52</td>
<td>31</td>
<td>87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>833</td>
<td>890</td>
<td>744</td>
<td>868</td>
<td>918</td>
<td>729</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>554</td>
<td>686</td>
<td>704</td>
<td>736</td>
<td>677</td>
<td>536</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>23</td>
<td>29</td>
<td>68</td>
<td>73</td>
<td>26</td>
<td>82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>683</td>
<td>775</td>
<td>816</td>
<td>825</td>
<td>1233</td>
<td>851</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>441</td>
<td>536</td>
<td>600</td>
<td>627</td>
<td>554</td>
<td>419</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>98</td>
<td>97</td>
<td>52</td>
<td>47</td>
<td>83</td>
<td>71</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>700</td>
<td>793</td>
<td>839</td>
<td>1271</td>
<td>1167</td>
<td>901</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>517</td>
<td>594</td>
<td>595</td>
<td>664</td>
<td>510</td>
<td>374</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>51</td>
<td>31</td>
<td>67</td>
<td>83</td>
<td>93</td>
<td>58</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>700</td>
<td>795</td>
<td>882</td>
<td>987</td>
<td>1007</td>
<td>1188</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>441</td>
<td>587</td>
<td>619</td>
<td>677</td>
<td>554</td>
<td>419</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>34</td>
<td>76</td>
<td>26</td>
<td>79</td>
<td>93</td>
<td>43</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>827</td>
<td>726</td>
<td>787</td>
<td>1295</td>
<td>958</td>
<td>751</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>344</td>
<td>471</td>
<td>594</td>
<td>594</td>
<td>594</td>
<td>511</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>41</td>
<td>59</td>
<td>82</td>
<td>58</td>
<td>37</td>
<td>78</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1130</td>
<td>1115</td>
<td>688</td>
<td>1154</td>
<td>697</td>
<td>713</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>344</td>
<td>382</td>
<td>477</td>
<td>449</td>
<td>442</td>
<td>287</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subject 12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 3 4 5 6 7 8 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68 59 91 58 73 34</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 1031 1077 671 726 795 1062</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>220 342 427 485 366 305</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43 97 46 29 86 78</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 770 746 1060 899 1161 1099</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>344 471 594 558 594 325</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>71 38 62 57 31 23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 770 746 1060 899 1161 1099</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 694 956 673 1003 1616 671</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>536 686 744 771 686 554</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>91 46 62 82 37 26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 928 1004 679 778 1576 728</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>477 594 635 667 572 334</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>39 57 82 41 98 53</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 1265 1036 847 730 896 899</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>449 645 619 740 600 631</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58 86 93 98 94 51</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 729 1205 694 962 1106 906</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>467 576 520 579 594 441</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>73 67 23 94 79 47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 755 1246 755 1053 1001 900</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>424 529 517 495 511 442</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>78 83 26 87 51 52</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9 968 1006 774 1165 1105 695</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>424 459 477 410 419 287</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
VITA

John Whalen

Personal


Education


1987-1991 McGill University, Montreal, Quebec. BSc, Psychology.

Publications & Manuscripts in Preparation


Paper & Poster Presentations


**Employment**

**January, 1992 to present**  
**RESEARCH/TEACHING ASSISTANTSHIP.** Cognitive Science Department, Johns Hopkins University, Baltimore, MD.

**Summer, 1994, 1993**  
**RESEARCH ASSISTANTSHIP.** Cognitive Neuropsychology Laboratory, Johns Hopkins University, Baltimore, MD.

**Summer, 1992, 1991**  
**RESEARCH ASSISTANT.** Cognitive Neurology, Sunnybrook Health Sciences Research Center, Toronto, Canada.

**Summer, 1990**  
**RESEARCH ASSISTANT.** Montreal Neurological Institute and McGill University, Montreal, Canada.

**Ad-Hoc Reviewer**

Brain Cognitive Neuropsychology  
Journal of Experimental Psychology: General  
Psychophysiology