Resolution of Lexical Ambiguity in Neural Networks: Distinctions and Cooperation between Models of Right and Left Hemispheres

Hananel Hazan

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE MASTER DEGREE

University of Haifa
Faculty of Social Sciences
Department of Computer Science

October 2007

Resolution of Lexical Ambiguity in Neural Networks: Distinctions and Cooperation between Models of Right and Left Hemispheres

By: Hananel Hazan

Supervised by: Dr. Larry Manevitz
Dr. Orna Peleg

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE MASTER DEGREE

University of Haifa
Faculty of Social Sciences
Department of Computer Science

October 2007

Approved by: ________________ Date: __/__/__
(Supervisor)

Approved by: ________________ Date: __/__/__
(Supervisor)

Approved by: ________________ Date: __/__/__
(Chairperson of M.A Committee)

<u>הכרת תודה</u>

ברצוני להביע את הכרת תודתי והערכתי

לד"ר **לארי מנביץ'** ולד"ר **אורנה פלג** על עזרתם ותמיכתם בכל תהליך לימוד והכנת עבודת הגמר, ברצוני להביע את תודתי העמוקה על הסבלנות במהלך החזרות האין סופיות לאורך כל תקופת העבודה המשותפת.

כמו כן ברצוני להודות פקולטה למדעי המחשב ולקרן קיסריה רוטשילד על תמיכתם הכספית ועזרתם במתן מילגה לעבודת המחקר

תודה מיוחדת **למשפחתי** שתמכה בי לכל אורך הדרך,

וכן **ליונת** שעזרה לי מימים ימימה במורל ריכוז וברעיונות יצירתיים.

Contents

Page number

Tables Contents

Figures Contents

Resolution of Lexical Ambiguity in Neural Networks: Distinctions and Cooperation between Models of Right and Left Hemispheres

Hananel Hazan

ABSTRACT

Reading is a complex and highly skilled act that requires different sources of information (e.g., phonological, lexical and contextual). Despite extensive study in recent years, how and when each type of information is utilized is still controversial and not fully explainable. Research shows that whereas both cerebral hemispheres participate in word processing, they do so in qualitatively different ways: First, the left hemisphere (LH) is characteristically faster and more accurate in its word processing. Additionally, the LH is more influenced by the phonological aspect of written words, whereas lexical processing in the right hemisphere (RH) is more sensitive to visual form. Finally, the LH quickly selects a single meaning of an ambiguous word, whereas the RH maintains alternate meanings. The major goal of this project was to propose and test the "split network model" where a single difference in architecture between the two cerebral hemispheres accounts for many performance asymmetries reported in the literature. It was concluded that in the LH network, orthographic units are directly related to both phonological and semantic units. However, because orthography is more systematically related to phonology than to semantics, the phonological computation of orthographic representations is faster than the semantic computation of these same representations. Consequently, meaning activation in the LH is initially influenced primarily by phonology. In the RH network, phonological codes are not directly related to orthographic codes, but are activated indirectly via semantic codes. This organization results in a different sequential ordering of events where the phonological computation of orthographic representations begins later than the semantic computation of these same representations. As a result, lexical access in the RH is initially influenced by orthography and by semantic information. This simple difference between the two cerebral hemispheres underlies not only phonological and orthographic asymmetries, but also higher-level semantic asymmetries

Parts of this research have appeared in the following publications:

1. Springer-LNAI (Lecture Notes in Computer Science) publication no.4840, 'Attention in Cognitive Systems', L. Paletta and E. Rome, Eds., to appear
2. 2007CNS Annual meeting (Cognitive Neuroscience Society) (May 2007)
3. IJCAI07 (International Joint Conference on Artificial Intelligence), India
    a. WAPCV (workshop on attention and performance in computer vision) (January 2007), proceedings of the workshop on Attention in Cognitive System. WAPCV-4,2007,p.21-34.
    b. NeSy'07 (Neural-Symbolic Learning and Reasoning) (January 2007) Published electronically. CEUR Workshop Proceedings, Vol. 230, 2007. ISSN 1613-0073. 31-37, India.

1 Introduction

Research demonstrates that whereas both hemispheres participate in word processing, they do so in different ways: First, the left hemisphere (LH) is characteristically faster and more accurate in its word processing. Additionally, the LH is more influenced by the phonological aspect of written words, whereas lexical processing in the right hemisphere (RH) is more sensitive to visual form. Finally, the LH quickly selects a single meaning of an ambiguous word, whereas the RH maintains alternate meanings. The major goal of this project was to propose and test the "split network model" in which a single difference in architecture between the two cerebral hemispheres can account for many performance asymmetries reported in the literature. In the LH, orthography, phonology, and semantics are interconnected. In the RH, however, phonology is not connected directly to orthography and hence its influence must be mediated by semantic processing.

This model was tested by examining how Hebrew readers and two fully connected networks, process two types of ambiguity:

- *Homophonic Homographs* - A single orthographic and phonological form associated with multiple meanings (e.g., *bank*)
- *Heterophonic Homographs* - A single orthographic form associated with two different pronunciations each associated with a different meaning (e.g., *tear*).

The computational model is based on the network described by Kawamoto [15] where spelling, pronunciation, part of speech, and meaning of words are represented as distributed patterns of activity over a set of simple processing units. For the LH network, the original model with fully connected units was used. For the RH stimulations, all things remained identical except that all unit connections in the orthography and phonology sub-areas were severed.

Forty-eight patterns were created to represent 24 Hebrew 3-letter homographs. Half of the patters were heterophonic and half were homophonic. They were chosen from the behavioral experiment. For each entry, 48 units represented the word's spelling, 80 units represented its pronunciation, 32 units represented its part of speech (all nouns), and 128 units represented its meaning. Overall, each entry was represented as a vector of 288 binary-valued features. Both networks were trained with a simple error correction algorithm. In each learning trial, an entry was selected randomly from the lexicon. Dominant and subordinate meanings were selected with a ratio of 5 to 3.

After the networks were trained, it was tested by presenting just the spelling part of the entry as the input. In order to assess lexical access, the number of iterations through the network for all the units in the spelling, pronunciation, or meaning fields to become saturated, was measured. Additionally, the activation of dominant and subordinate meanings of a given homograph was examined as a function of time.

Overall, the two networks produced processing asymmetries comparable to those found in behavioral studies:

(1) The LH network had superior performance.

(2) The LH network was more sensitive to phonological processes.

(3) The subordinate meaning was suppressed at an earlier stage in the LH network.

It was concluded that in the LH network, orthographic units are directly related to both phonological and semantic units. However, because orthography is more systematically related to phonology than to semantics, the phonological computation of orthographic representations is faster than the semantic computation of these same representations. As a result, meaning activation in the LH is initially influenced primarily by phonology. In the RH network, phonological codes are not directly related to orthographic codes, but are activated indirectly via semantic codes. The results of this organization resulted in a different sequential ordering of events where the phonological computation of orthographic representations began later than the semantic computation of these same representations. As a result, lexical access in the RH is initially influenced by orthography and by semantic information. This simple difference between the two cerebral hemispheres underlies not only phonological and orthographic asymmetries, but also higher-level semantic asymmetries.

Given the above hemispheric differences in lexical processing, the second goal of this experiment was to examine how the two cerebral hemispheres interact during ambiguity resolution. Because the two hemispheres are connected via the Corpus Callosum, it was assumed that the exchange of information between the two hemispheres is useful for human understanding as well as everyday reading and interpretation of text. Therefore, the possibility that corrections (reinterpretation) in ambiguous word processing can be aided by information possessed by the opposing hemisphere was examined. Specifically, investigations were conducted to see how the LH uses the information of the RH to complete a specific task. The aim of the experiment was to determine what occurs when the LH is innately drawn to one solution, but during processing, in light of new information, chooses the other solution. Since the different models have different rates of convergence, the hypothesis was tested by halting processing and using an analogue to priming in the network in order to compare the rate of convergence to corrected semantics in the LH. The LH worked alone but using information obtained from the RH in its processing. Results indicate that usage of information by the RH does help in the disambiguate processes.

2 Background Information

Neuropsychological studies show that both cerebral hemispheres process orthographic, phonological, and semantic aspects of written words in different ways. These studies show that the LH is more influenced by the phonological aspect of written words whereas lexical processing in the RH is more sensitive to visual form. Additionally, semantically ambiguous words (e.g., "bank") were found to cause different time-lines of meaning activation in the two hemispheres. Computational models of reading in general and lexical ambiguity resolution do not incorporate this asymmetry into their structure.

2.1 Lexical Ambiguity Resolution: A Connectionist Perspective

The connectionist approach is appealing because its models of cognitive processes are computational. That is, they actually produce a response to a stimulus. The predictions that such models make about reaction time (e.g., how long it takes the model to generate a response) or error rate can be compared to the behavior produced by subjects in experiments. Computational models provide not only a replication of empirical findings but also important insights as to why such findings arise. The main advantage of computational models is that modifications can be done that cannot, easily, be done in human based studies. The computational approaches used throughout this paper attempt to address the nature of underlying mechanisms at a level where intuition is not easily penetrable. Many computational language models, such as: information retrieval, automatic abstracting, and machine translation [e.g. Manevitz, L.M., & Zemach, Y. [19]] require the resolution of lexical ambiguity for words in an input text. The more accurate the lexical ambiguity resolution, the better the results produced by the computational models.

There are two types of connectionist models designed to handle visual word recognition in general and lexical ambiguity resolution:

1. A feed-forward network, e.g., Harm’s & Seidenberg’s network [12], Seidenberg, M.S. & McClelland J. L [28].
2. A fully recurrent network - e.g., Kawamoto’s network [15].

2.1.1 Harm’s & Seidenberg’s Network:

Harm’s & Seidenberg’s Network was based on the Seidenberg’s & McClelland’s network, a model designed for reading. It focused on the translation from print to sound.

Harm and Seidenberg designed their model based on findings from behavioral studies addressing how meaning is computed in a system where both visual (orthography to semantics) and phonologically mediated (orthography to phonology to semantics) pathways are available (Seidenberg, M. S. & McClelland J. L [28]).

These behavioral studies refer to a resolution of the meaning with homophones and non-homophones. They do not refer to differences between homophonic homographs and heterophonic homographs. Similarly, they reference the time it takes to access the meaning of homophonic homographs compared to the time it takes to access heterophonic homographs.

Seidenberg's & McClelland's Model

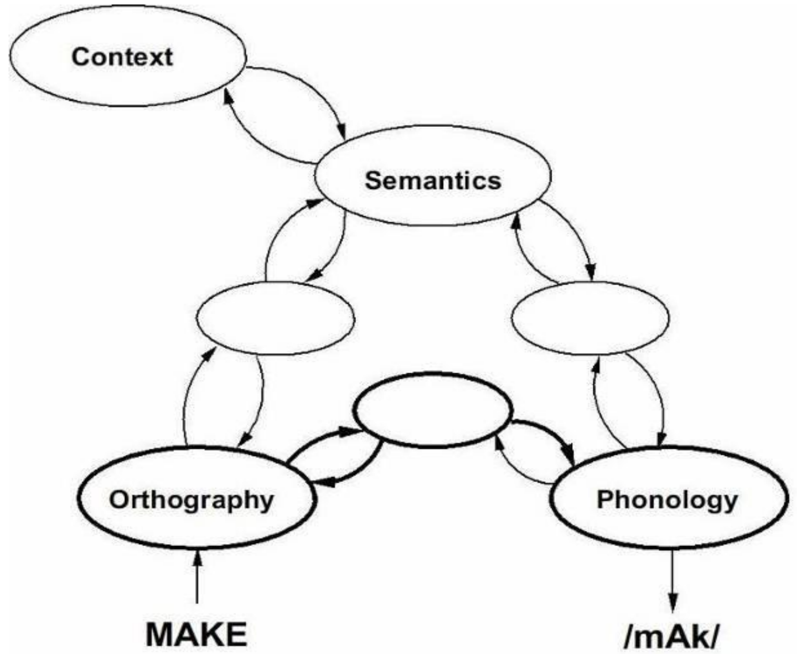

Figure 1: Seidenberg and McClelland's (1989) "triangle" model shows how phonological codes are computed from orthography given the availability of both direct (orth->sem) and phonologically mediated (orth->phon->sem) pathways. (The Diagram is taken from Seidenberg, M. S., & McClelland J. L.,[28])

2.1.2 Kawamoto's Network:

Kawamoto designed his neural network model so an entire word, including its orthographic, phonological, and semantic features appear as "attractors" in the recurrent network (a Hopfield network [13]). According to his model, the more frequent a meaning of a word in a certain context is, the stronger the meaning's attraction to the word. In choosing a meaning, phonological features are the primary attractor.

Another examined factor was the time lapse between accessing the dominant meaning and the time lapse of accessing the secondary, subordinate, meaning (Kawamoto [15].).

The network that Kawamoto built to solve the problem of ambiguity in meanings of English words was based on the frequency of word meaning's of the word. His model predicted the ambiguity of words as humans do. Kawamoto's work is a milestone in the research of the neural network in the English language.

Kawamoto [15] presented a connectionist account of lexical ambiguity resolution. In his recurrent network (see Figure 2), ambiguous and unambiguous words were represented as a distributed pattern of activity over a set of simple processing units. Each lexical entry was represented over a 216-bit vector and was divided into separate sub-vectors representing the "spelling," "pronunciation," and "meaning." His network was trained with a simple error correction algorithm, presenting the network with the pattern to be learned. These patterns (the entire word including its orthographic, phonological, and semantic features) became "attractors" in the 216 dimensional representational spaces.

After the training, the network was tested. It was presented with part of the lexical entry (e.g., its spelling pattern) to see how long parts of the network took to settle into a pattern corresponding to a particular lexical entry. Kawamoto trained his network in such a way that the more frequent combination of a particular orthographic representation was the "deeper" attractor. For example, the completion of the other features (semantic and phonological) would usually fall into this attractor. This was accomplished by biasing the learning process of the network. Using a technological analogy of "priming" to bias the appropriate completion, the resulting attractor could be the less frequent combination.

The Kawamoto Model

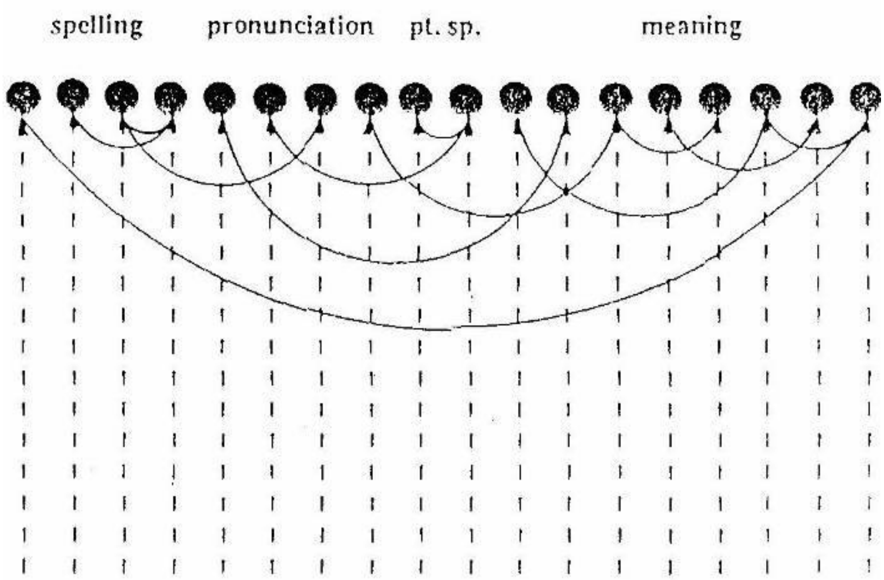

Figure 2: The structure of the connectionist network representing the spelling, pronunciation, part of speech, and meaning of a word is illustrated. 216 interconnected units comprise the network. (Note: Only some of the units and connections are shown in the diagram. The Diagram is taken from Kawamoto [15])

2.1.3 Conclusions of Early Models

Early models [28] exhibit local representation schemes, representing words and learned connection strengths. The Kawamoto Model instantiates a pure connectionist model using a distributed representation scheme. The Kawamoto Model is similar to Seidenberg and McClelland's [28] word recognition model. However, there are many important differences. The Seidenberg and McClelland Network is an extremely complicated network that is neither completely connected nor simple. The Kawamoto Model was used in this research because unlike the Seidenberg and McClelland Network it is a completely connected network expressed in the simplest possible way. Kawamoto's Model was reconstructed to recreate its theoretical tests (elaborated on in Appendix A. Ambiguity Resolution of Homographs and presented BISFAI 05). There are several important differences in the implementation of the Seidenberg and McClelland models and the Kawamoto Model. The most important difference is in the structure of the network. The difference between these two models is that Kawamoto added syntactic and semantic features to the lexicon. The network implemented by Seidenberg and McClelland was a feed-forward network with hidden units. Kawamoto's Model is fully recurrent without hidden units. These structural differences are important in the Kawamoto Model because they initiate differences in the units' response times for the lexical decision task and the naming task. In Seidenberg and

McClelland's implementation, the response made by the network corresponds to the possible output that is most similar to that made by the network. The magnitude of the error is used as an index for the response time. Contrastingly, the time course of response is emphasized in the Kawamoto Model.

A correlation can be drawn between Kawamoto's findings and the behavioral data found in this experiment. Consistent with these human empirical results, after the network was trained, the resolution process was affected by the frequency of the different lexical entries and context. This is reflected in the strength of the connections in the network used in this experiment. Kawamoto did not incorporate hemispheric differences into his network. His model was a general model of processing response times.

2.2 Lexical Ambiguity Resolution: a Psycholinguistic Perspective

A plethora of evidence from the lexical-semantic literature indicates that comprehenders utilize two important sources of information to resolve lexical ambiguity. First, a particular meaning of a homograph may be more frequent or dominant than another. Second, the particular context in which the homograph is embedded may be biased toward one particular interpretation [e.g., Duffy, Morris & Rayner [39]; Peleg, Giora & Fein [23, 24]].

2.2.1 Hemispheric Differences in Lexical and Contextual Processing

Both frequency and context have different implications in the processing of ambiguous words in the two cerebral hemispheres [e.g., Beeman [1]]. For example, Burgess and Simpson [5] showed that relative meaning frequency has different implications for the processing of homophonic homographs in the hemispheres. The LH accesses all of the meanings of an ambiguous word very quickly and then suppresses the less frequent meaning. The RH, however, activates both meanings more slowly and maintains these meanings. This research spawned a large amount of research to use not only single words, but also sentence contexts. These subsequent studies show that when context is biased towards one meaning of a final ambiguous word, the RH immediately activates only the contextually appropriate meaning. Both meanings are initially activated in the LH [e.g., Titone [31]; Swinney & Love [29]]. With longer delays, the opposite pattern is found: the LH selects the contextually appropriate meaning of an ambiguous word, whereas the RH maintains alternate meanings [e.g., Faust & Gernsbacher [87]; Faust & Chiarello, [7]]. Three major proposals account for the sustained activation of less frequent and/or contextually incompatible meanings in the RH as opposed to their fast decay in the LH. First, according to The Coarse Coding Model suggested by, Beeman [1], representations in the LH are finely coded. This means that narrow representations exist that include closely related meanings. Semantic representations in the RH are coarsely coded. This means that there are broader representations, including less related meanings. Second, several researchers proposed that hemispheric differences in word meaning activation result from a selection mechanism, specific to LH processing, that inhibits or suppresses less related meanings [e.g., Tompkins [32]]. Finally, Burgess [4] and Lund suggested that initial differences in activation speed could account for differences in meaning activation. In other words, dominance leads to both stronger and longer activations of word meanings for both LH and RH processing. Therefore, less related meanings decay faster. Since RH processing has initially slower speed activation, less related meanings are still activated.

The connectionist model developed by Kawamoto [15] is highly consistent with the time course obtained in the LH by Burgess and Simpson [5] in addition to others [Swinney & Love [29]; Titone [31]]. Kawamoto's Model demonstrated that although just one sense of an ambiguous word is eventually accessed, all the meanings of an ambiguous word are initially activated to some extent. This model cannot account for RH patterns.

2.2.2 Hemispheric Differences in Phonological and Orthographic Processing

In Latin orthographies, such as English, the orthographic representation (the spelling) of a word is usually associated with one phonological representation. Most studies of lexical ambiguity use homophonic homographs. Homonyms are a single orthographic and phonological representation associated with two meanings. As a result, models of hemispheric differences in lexical processing have mainly focused on semantic organization [e.g., Beeman [1]]. From this, we inferred that a reliance on homonyms might limit understanding of hemispheric involvement in meaning activation by neglecting the contribution of phonological asymmetries to hemispheric differences in semantic activation. Simultaneously, we believed that this limited the range of proposed models describing the process general reading.

Visual word recognition studies demonstrate that despite both hemispheres having access to orthographic and phonological representations of words, the LH is more influenced by the phonological aspects of a written word than the RH [e.g., Zaidel, [37]; Zaidel & Peters [38]; Lavidor and Ellis [16]]. Lexical processing in the RH is more sensitive to the visual form of a written word [e.g., Marsollek, Kosslyn & Squire, [20]; Marsolek, Schacter & Nicholas [19]; Lavidor and Ellis [16]]. Given that many psycholinguistic models suggest that silent reading always includes a phonological factor [e.g., Berent & Perfetti [3]; Frost [10]; Van Orden [28], Pennington & Stone [17], Lukatela and Turvey [18]], it is conceivable that such asymmetries may also impact the assignment of meaning to written words during online sentence comprehension.

2.3 The Complexity of Hebrew

Our study used Hebrew orthography. In contrast to less opaque Latin orthographies, Hebrew offers an opportunity to compare different types of ambiguities within the same language [e.g., Frost and Bentin [9]]. In Hebrew, letters represent mostly consonants. Vowels can be superimposed on consonants as diacritical marks. Since vowel marks are usually omitted, readers frequently encounter words with more than one possible interpretation (homograph).

The first is a "homophonic homograph" - a word that could have different meanings while its pronunciation does not change. For example, the string of letters "גיל" (GIL) can mean "age" ("גִּיל") (GIL) or "happiness" ("גִּיל") (GIL).

The second is a "heterophonic homograph" - Two words that sound differently, but are spelled the same way. For example, in Hebrew the consonant string ("ספר") can represent (sefer) - “a book" (“סֵפֶר”) or (sapar)-“a barber" (“סַפָּר”). The reader can tell the difference only if the word is dotted, or if a context is given.

For further elaboration on homographic differences and the repercussions of processing homographs, see Appendix A. Ambiguity Resolution of Homographs

2.4 Summary and Purpose

Research demonstrates that whereas both hemispheres participate in word processing, they do so in qualitatively different ways: First, the LH is characteristically faster and more accurate; in addition, the LH is more influenced by the phonological aspect of written words, whereas lexical processing in the right hemisphere (RH) is more sensitive to visual form. Finally, The LH quickly selects a single meaning of an ambiguous word, whereas the RH maintains alternate meanings. However, computational models of reading in general and of lexical ambiguity resolution in particular, have not incorporated this asymmetry into their architecture. The major goal of this project was to propose and test the "split network model" where a single difference in architecture between the two cerebral hemispheres can account for many performance asymmetries reported in the literature. In the LH, orthography, phonology, and semantics are interconnected. In the RH, however, phonology is *not* connected directly to orthography and hence its influence must be mediated by semantic processing.

LH Structure: Orthographic, phonological, and semantic codes are fully connected in the LH. The connections between these different sources of information are bidirectional and the different processes may very well run parallel. However, the model incorporates a sequential ordering of events that result from some processes occurring faster than others do. For example, in the LH, orthographic codes are directly related to both phonological and semantic codes. Because orthography is more systematically related to phonology than to semantics, the phonological computation of orthographic representations is faster than the semantic computation of these same representations. As a result, meaning activation in the LH is initially influenced primarily by phonology [e.g., Lavidor & Ellis [16]]. This result in the immediate exhaustive activation of all meanings related to a given phonological form, regardless of frequency or contextual information [e.g., Burgess & Simpson [9]; Titone [31]; Swinney & Love, [29]].

RH Structure: Phonological codes are not directly related to orthographic codes and are activated indirectly via semantic codes. This organization predicts a different sequential ordering of events in which the phonological computation of orthographic representations begins later than the semantic computation of these same representations. Consequently, lexical access in the RH is initially influenced by orthography [e.g., Lavidor & Ellis [16]] and by semantic information so that less frequent or contextually inappropriate meanings are not immediately activated. These meanings can be activated later when phonological information becomes available [e.g., Burgess & Simpson [5]; Titone [31]].

The "split model" was tested by examining how the two connectionist networks process homophonic homographs, (two possible meanings associated with a single visual and phonological form.) versus heterophonic homographs (a string of letters associated with two different pronunciations each of which has a different meaning).

3 The Split Reading Model

3.1 Simulation Details

The computational model is based on the network described by Kawamoto [15] where the spelling, pronunciation, part of speech, and meaning of words are represented as distributed patterns of activity over a set of simple processing units was used. For the LH network the original model, with fully connected units was used. For the RH network simulation, everything remained identical with the exception of the connections between orthography and phonology sub-areas that were severed.

In conduction this experiments a model was proposed (see Figure 3) in which a single difference in structure between the two cerebral hemispheres accounts for performance asymmetries reported in psycholinguistic literature. In the LH, orthography, phonology, and semantics are interconnected. In the RH, phonology is not directly connected to orthography. Therefore, the influence of orthography needs to be mediated by semantic processing.

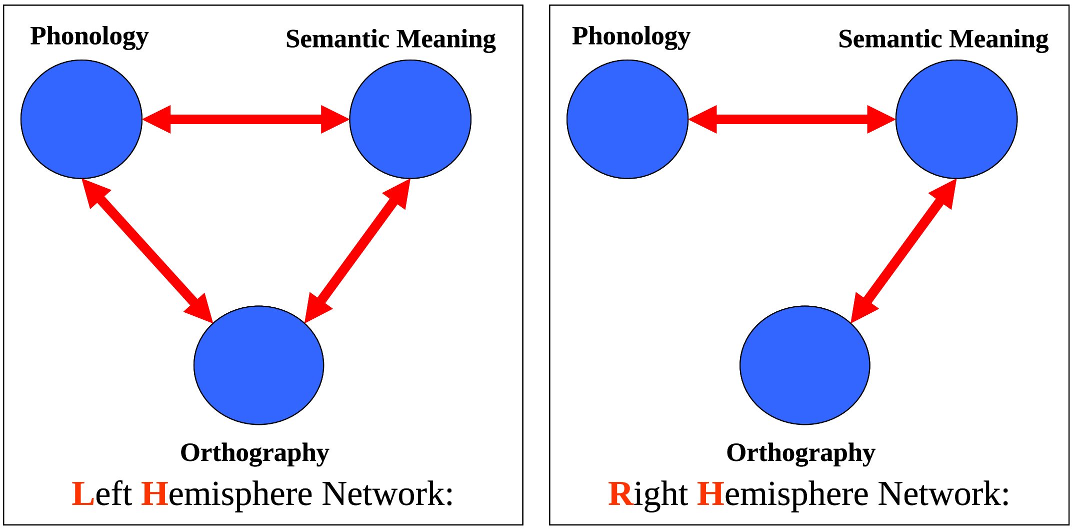

Figure 3: Left and Right hemisphere structures. The (red) arrows show the directions of information exchange between the sub-vectors of the network.

3.2 Lexicon Representation in the Two Networks:

Each network learned a lexicon consisting of three-letter words (see Table 1): 24 pairs of Homophonic Words, 24 pairs of Heterophonic Words. Each word in the lexicon had a pronunciation, part of speech, meaning, and frequency. The frequency was based on a questioner that tested human subjects to see what they thought of when they read certain words. Each entry was represented as a vector of 288 binary-valued features: 48 features represented its spelling. 80 features represented its pronunciation. 32 features representing the part of speech. The last 128 features represented its meaning.

A vector of 16 bipolar bits represented each character (-1, 1). Combined with different methods of representation, these selections caused the size of the network to be proportional to the vector being taught. A unit received and processed every bit of encoding. After receiving and processing, the values from the units would spread the bit throughout the network (see Figure 6).

Four sets of 27 sub-vectors were created. Each sub-vector corresponded to a letter of the Hebrew Alphabet (see appendix on encoding). Each sub-vector was chosen using a hamming distance function. This function measures the distance between the sub-vectors. To create the training set an entry from the lexicon was encoded using the proper encoding tables. These tables were created for orthography, phonology, part of speech, and meaning. In each part of the vector, a different sub-vector was assigned (see Figure 4). The lexicon represented each letter in a like group utilizing a specific character (see the character 's' at pronunciation or the character 'י' in meaning in Figure 4), however, the same letter was represented in a different group using a different character (see the character 'פ' at orthography and meaning in Figure 4).

In the learning phase, this lexicon was used to create a training set that used the frequency as a ratio between the dominant and the subordinate words. Using this ratio in the training set (see Table 1) dominant words appeared more often than subordinate words. In the training set the lexicon learned that the meaning of dominant words were of greater significance than subordinate words.

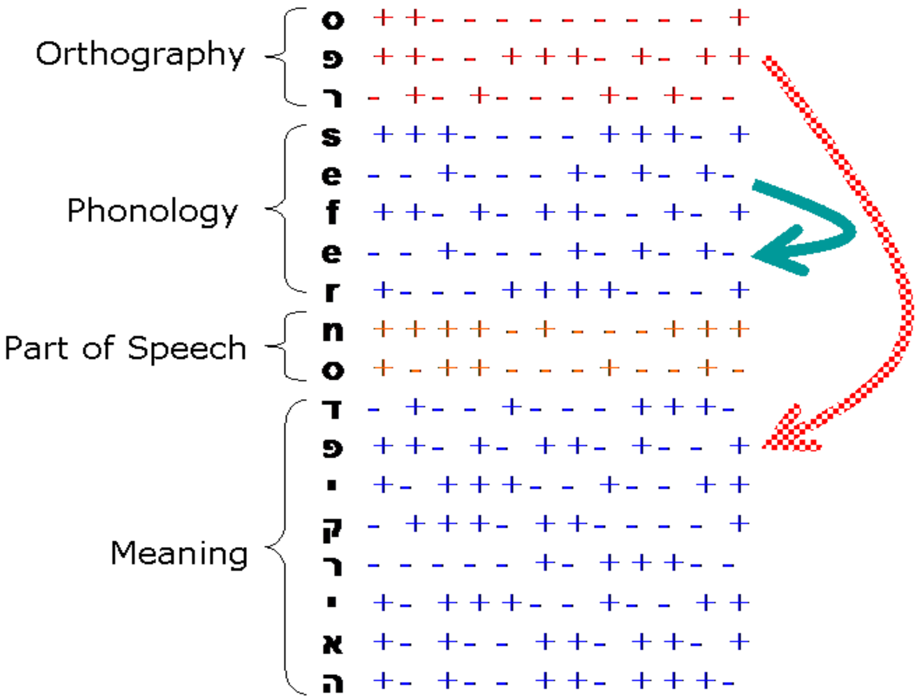

Figure 4: Representation of a vector from one entry of the lexicon. It is important to note that different encoding in the vector (red or grey arrow), and equal encoding in the same part (green or solid arrow). Each part has a different encoding.

Homophonic Homographe

Heterophonic Homographe

Orthography	Pronunciation	Part of Speech	Meaning	Frequency
שיח	ci$ax	no	עצקטנגנה	5
שיח	ci$ax	no	שיחהדיונ	3
חלל	xalal	no	חלולאויר	5
חלל	xalal	no	נפלבמלחמ	3
מפה	mapa$	no	דרכימארץ	5
מפה	mapa$	no	נפרסשולח	3
שאף	s$aaf	no	חמצוננשם	5
שאף	s$aaf	no	מטרהרצונ	3
הגה	$hege	no	אוטונוהג	5
הגה	$hege	no	השמיעקול	3
בית	bai$t	no	גרימדירה	5
בית	bai$t	no	חלקשלשיר	3
רגל	regel	no	הליכהאבר	5
רגל	regel	no	אורךמדיד	3
מנה	mana$	no	ארוחההחלק	5
מנה	mana$	no	חשבתרגיל	3
חבל	xevel	no	לקשורחוט	5
חבל	xevel	no	אזורבארץ	3
עצב	e$zev	no	יגונוצער	5
עצב	e$zev	no	שרירבגוף	3
שאל	saa$l	no	תשובהרצה	5
שאל	saa$l	no	ביקשלקבל	3
מטה	mate$	no	פיקודצות	5
מטה	mate$	no	שרביטאלה	3
בור	bor	no	חולחפירה	5
בור	bur	no	רפהשכלמח	3
יוד	yu$d$	no	פצעטיפול	5
יוד	yo$d$	no	אותהקליד	3
ספר	cefer	no	קראנובלה	5
ספר	capar	no	תסרוקתפנ	3
מלח	melax	no	תבלינלבנ	5
מלח	malax	no	חובלאניה	3
קצב	kezev	no	טמפושירה	5
קצב	kazav	no	חותכסטיק	3
אלף	e$lef	no	מספרגדול	5
אלף	a$lef	no	אותראשונ	3
שמש	semes	no	חמאורקיץ	5
שמש	samas	no	עובדכיתה	3
שוק	su$k$	no	דוכנירקת	5
שוק	so$k$	no	ירךשלעופ	3
דבר	davar	no	חפצאובקט	5
דבר	dever	no	מגפהמחלה	3
אתר	a$tar	no	ימבומבקר	5
אתר	e$ter	no	מחומרמרד	3
אמנ	a$man	no	יוצראמנו	5
אמנ	a$men	no	אמרותפלה	3
ספק	cafek	no	לאבטוחהס	5
ספק	capak	no	מביאדברם	3

Table 1: Networks learned during the training phase. The training set was created using this lexicon

3.3 Networks' Structure

Each network was composed of 288 units corresponding to the 288 features of the lexical entry. Each lexical entry was a pattern of activity across all the units. Each unit in the network was a part of the representation of a lexical entry. Every unit received an input from the external environment, including all other units in the network. The output of each unit was measured as a current state of the unit and was transmitted to the input of all other units (see Figure 6). Using this structure (see Figure 5), the units spread the learning process across the network. The information gained during the learning phase was stored throughout the network, not in one specific place.

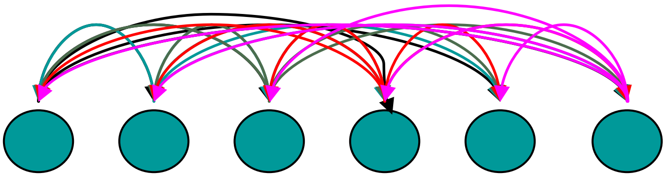

Figure 5 : Connection between the units

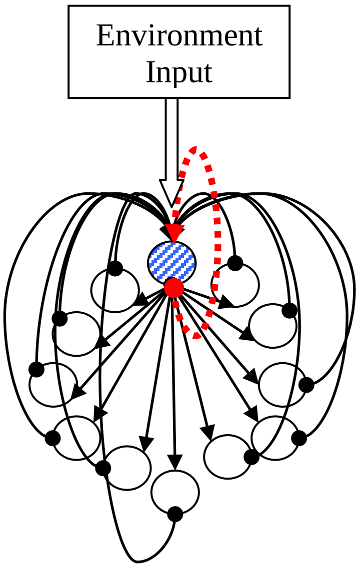

Figure 6: The output of the neuron (marked with blue solid lines) in the network is connected to the input of all other neurons. The input toward the neuron (marked with blue lines) is the result of environmental influences, previous value of himself (red dotted line), and outputs of all other units.

The LH network was constructed according to the simplest, completely connected connectionist architecture. The network was composed of numerous neurons called units (see Figure 6). Each unit's output in the network was connected to the input of all other neurons. The input of the neuron was the result of environmental influences and outputs of all other units (see Figure 5 and Figure 6). Each unit in the structure had the same influence on all neurons in the iterations and spread the local input of one individual neuron through the entire network (see Figure 6 and Figure 5). The second network was labeled the RH. In this network, the phonological units were not directly connected to orthography units. The information to those units came only from the meaning units. The size of both networks and the number of neurons that composed each network were proportional to the encoded characters and the size of the vector needed to be taught.

It is important to note that the only difference between the LH Model and the RH Model were the connections between the network units. Aside from this difference, everything that will be described from this point on: building, representing, training, and testing are the same for both networks. In creating a different structure, the differences between the two networks were investigated to see if already existing models explain the findings of these experiments.

3.4 The Network's Learning Process

3.4.1 The Learning Algorithm

The learning equations of the two models were implemented from Kawamoto [15] model, the algorithm of the learning was implemented also from Kawamoto but with some modification that will be elaborate in the appendixes. The initializations of the weights between the units start with zeros. The learning phase contained two steps:

A. First, randomized training sets were from the lexicon.
B. Second, each line from the training sets was given to the networks. Then the weights between the units were updated.

3.4.1.1 Step A:

A training set was created as a repetition of 1300 entries chosen from the lexicon. The ratio between the dominant and subordinate words was maintained through the process. The ratio of 5 to 3 was chosen. This means that where the subordinate words were chosen 3 times, the equivalent dominant words were chosen 5 times. The words were chosen from the lexicon in random order, maintaining the dominant/subordinate ratio.

3.4.1.2 Step B:

The networks learned a vector (i.e. one entry) from the training set by inputting the values of the vector into the input of the units and simultaneously adjusting all the weights between the units using Equation 1. This process was used for every vector in the training set.

The learning was made possible by giving the input to all the units and changing the weight, using Equation 1. This training scheme was an "error-correction" algorithm that is based on an approach developed by Kawamato [e.g. Kawamoto [15], Rosenblatt [26], and Widrow & Hoff [36]]. To train the network, one entry was presented to the network. This activated the corresponding units in the network and set the activation level to the appropriate value: +1.0 if the feature is present or -1.0 if the feature was absent.

The networks' learning process was achieved by altering the weights between the units in the network. The strengths of the connections to a unit were computed by modifying, to minimize (i.e., correct), the error between its activation level, determined by the environment and the network input. This adjustment in the connection strength from a given unit j to a unit i, ΔW_{ij}, can be expressed in terms of Equation 1

Equation 1: Learning function by changing weights

$$\Delta W_{i,j} = \eta\left(t_i - i_i\right)t_j \text{, where } i_i = \sum_j W_{ij} t_j$$

η is a scalar learning constant. It is set to 0.0015. (A further elaboration of η is provided in the appendix.) t_j and t_i are the target activation levels of units i and j. i_i is the sum of all other units outputs to unit i. The magnitude of the change in connection strength was determined by the magnitude of the learning constant and the magnitude of the error $t_i - i_i$.

This learning algorithm was sensitive not only to the input during a particular learning trial, but also to the frequency at which the inputs were presented during the training. Accordingly, the network would receive greater exposure to words that are more frequent in the language.

3.5 Running / Testing each of the Networks Separately

The accuracy in the reaction and speed of saturation of the networks' were tested. To do this several tests were performed after each network learned the lexicon, including respective frequencies of dominant and subordinate meanings. In the initial test, the network was presented with the orthography of a word. In the subsequent tests, the network was presented with the orthography of the word and clues[1], hinting at the word's meaning. Testing commenced with the presentation of one character from the meaning as a clue. Testing continued with the increasing of meaning characters presented until all characters were presented as clues. The aim of this test was to check the activity and flow of information in the network. The activity of a unit in the network is the output of the unit, measured by a real number ranging between -1.0 and +1.0. This activity was determined using three factors: the external input from the unit's environment, the output of all the other units, and the value of the unit from the last iteration multiplied by its decay value. These factors led to changes in the activity of a unit as a function of time (where time changes in discrete steps). That is, the activity of a unit i, a_i, at time t +1 is:

Equation 2: Activation function

$$a(t+1) = \text{Limit}\left[\delta a(t) + \left[\sum_j w_{ij}(t)\,a_j(t)\right] + s_i(t)\right]$$

δ is a decay constant that set to 0.7, its purpose to add to the current value of unit 'a' at time (t+1) 70% of the previous iteration of 'a' at time (t). $s_i(t)$ is the input from the environment to unit i. The LIMIT function (see Equation 3) bounds the activity of the unit into real numbers ranging in value from -1 to +1. Here δ is a decay variable on the last value of unit i that is influenced by the strength of the past activity of i on the current state, $s_i(t)$ is the influence of the input on unit i, and the LIMIT bounds the activity to the range of -1 to +1.

Equation 3: Limit function on activation function

$$\text{Limit}\ (x)\begin{cases} +1 & x > +1 \\ -1 & x < -1 \\ x & +1 \geq x \geq -1 \end{cases}$$

The input stimulus from the environment could be any number between -1 to +1. We chose an input of -0.5 and +0.5 as -1 and +1 respectively in order for the network to become

[1] Clues" were implemented by giving external stimulus to some of the semantics units. See Figure 10 at page 19

saturated more slowly. Zero is a neutral value, which does not influence unit i or any other unit that is influenced by the outputs of unit i while calculating the sum of all other output including unit i. Zero was used for the units we did not want to give any input to (not -1 and not +1). Instead, these units received information from the units that were given an input, eventually causing all units to become saturated.

Three things influenced unit i (see Figure 7):

1. External stimuli (s_i) (red or dotted arrow in Figure 7).
2. Previous values from the last iteration multiplied by the decay rate (δ) (blow or striped arrow in Figure 7).
3. Sum of the output of all other units in the last iteration of the network (green or solid arrows in Figure 7).

The behavior of the unit was dependent on the differing strengths and thereby influenced by the three previously stated inputs. If one of the three inputs was stronger than the other two, that unit was less affected by the weaker inputs. Because the individual units responded collectively, the network worked to disambiguate the word while "ignoring" some of the input. For example, if the strength of the input of past experiences was stronger than the sum of the other entire units' output, the network would "ignore" the other inputs to pursue disambiguation according to the information gathered from past experiences.

The activity in all the units at a given point in time corresponds to the state of the network at that time. The state of the network continuously changed because the activity of the units was always changing. All of the activities of the networks' units were updated simultaneously. Each of these updating cycles marked one iteration in the network.

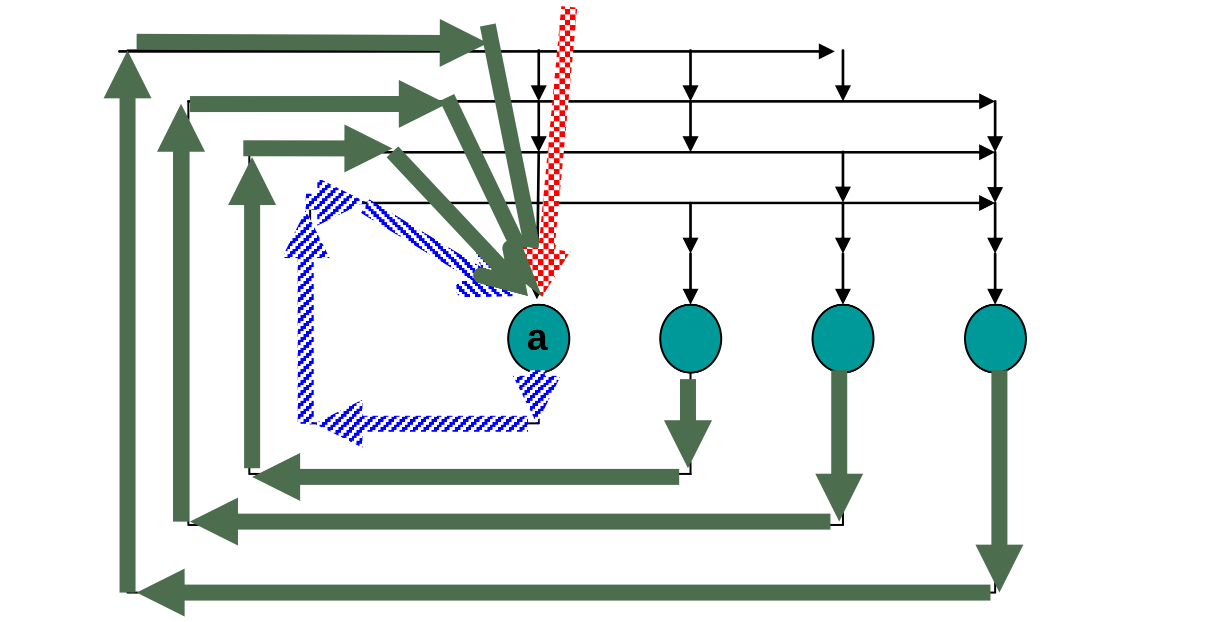

Figure 7 : The abouve shows the network structure with the three different inputs. The green, or solid, arrows are the inputs to unit a from all outputs of the other units. The red, or dotted, arrows are the inputs from the envoirment. The blue, or striped, arrow is the previous value of unit a.

The iteration of the network was restricted to 50 iterations. Since the network was not a pure Hopfield Network [13], there was no guarantee of convergence. However, non-convergence was a non-phenomenon (see appendix). If the network did not converge by 50 iterations, it was classified as an erroneous result. A network converged when all of its units reached their maximum potential. This signified that the values of the units in the previous iteration were equal to those of the current iteration and that there was no change in the state of the network. If the network did not converge at or before 50 iterations, the test results were classified as errors. Unlike the Hopfield Network [13], when the network passed 50 iterations, it does not converge. Thus is, for the purpose of this experiment, the networks' potential to converge after 50 were minuscule. This concept will be expanded on later in the discussion of log analysis. The network would stop on two conditions:

First, if the network reached 50 iterations and did not converged.

Second, if all the units reached the maximum potential and the network converged.

All the activity of the LH and RH networks ware logged and analyzed using the methods described in the following section.

3.6 Log Analysis

During the testing phase, the networks were tested using several inputs. As previously stated, in the initial test the network was presented with the orthography of a word. In the following tests, the initial input includes clues[2] from the word's meaning. All the activities between the units of every iteration on each input. The ensuing log contains the activities of each unit in the network from the initial iteration to the network's point of convergence. This log enabled the evaluation and analysis of the informational flow between the units as well as the evaluation to see if the network activated more than one meaning while working to disambiguate words. To enhance the study, this log was also used to compare the two architectures of the networks, to see differences, and to evaluate the flow of information of both hemispheres in comparison to human subject findings.

After each test, we recorded the network's activities and then, in preparation for the next test the network was reset to its original condition, which it had after the initial teaching of the lexicon. When evaluating the log of each test the following parameters were used:

1. The number of iterations necessary for the network to converge to a meaning (dominant, subordinate, etc).
2. The state of the network during the test by comparing the values of the units to the original vectors that were taught in the learning set. This enabled the evaluation of the level of activation of the units during the disambiguation process as well as what other meanings were activated during this process.
3. The saturation of each sub-vector through evaluations such as determining if the part of phonology in the vector was saturated faster than the meaning vector, etc.
4. A comparison between different network 'subjects,' that detail the teaching of the lexicon to the network, of either identical input or input from the same category (homophones, heterophones, normal words, etc…). This was followed by a statistical analysis of the summary results (see Figure 8).

[2] Clues were implemented by giving external stimulus to some of the semantics units. See Figure 10 at page 20

Part of the log file of one of the experiments, on no clue or one clue.

Test = 08-Jul-2007Normal until final round of learning on LH=2,RH=2 clouse s=1 t6 mu =0.0001
numberOfLinesToLearn =30000 - Normal/n
clues = 0
# iteration Homophone Dominant = 10.4583 std = 0.61416 anova=0.65581
# iteration Homophone Subordinat = 0 std = 0 anova = NaN-0
# iteration Hetrophone Dominant = 11.8646 std = 1.4482 anova = 0.6746
# iteration Hetrophone Subordinat = 0 std = 0 anova = NaN-0
# iteration Normal Word Dominant = 8.7917 std = 1.3986 anova = 1
# iteration Normal Word Subordinat = 17.2917 std = 10.483 anova = 0.99307

Hetrophone Dominant is slower from Homophone by = 1.4062 rounds std=0.40984

round result on iteration on pronunciation Homophone Dominant = 5.75 std = 1.4868
round result on iteration on pronunciation Homophone Subordinat = 0 std = 0
round result on iteration on pronunciation Hetrophone Dominant = 10.2708 std = 1.2094
round result on iteration on pronunciation Hetrophone Subordinat = 0 std = 0
round result on iteration on pronunciation Normal Word Dominant = 7.5208 std = 1.3532
round result on iteration on pronunciation Normal Word Subordinat = 14.5104 std = 8.3224

Hetrophone Dominant is slower from Homophone by = 4.5208 rounds std=0.34883

round result on iteration on meaning Homophone Dominant = 9.4688 std = 0.75328
round result on iteration on meaning Homophone Subordinat = 0 std = 0
round result on iteration on meaning Hetrophone Dominant = 10.8333 std = 1.3967
round result on iteration on meaning Hetrophone Subordinat = 0 std = 0
round result on iteration on meaning Normal Word Dominant = 7.8542 std = 1.3991
round result on iteration on meaning Normal Word Subordinat = 15.8542 std = 10.4971

Hetrophone Dominant is slower from Homophone by = 1.3646 rounds std=0.35539
****
Homophone Correct - Dominant -> Subordinat = 96 ; 0 / 96
Homophone Corrent - Subordinat -> Dominant = 0 ; 96 / 96
Hetrophone Corrent - Dominant -> Subordinat = 96 ; 0 / 96
Hetrophone Corrent - Subordinat -> Dominant = 0 ; 96 / 96
Normal Word Corrent - Dominant -> Subordinat = 96 ; 0 / 96
Normal Word Corrent - Subordinat -> Dominant = 96 ; 0 / 96

Sum Of Errors Homophone Dominant = 0
Sum Of Errors Hetrophone Dominant = 0
Sum Of Errors Normal Dominant = 0
Sum Of Errors Homophone Subordinat = 0
Sum Of Errors Hetrophone Subordinat = 0
Sum Of Errors Normal Subordinat = 0
Sum Of Errors per net = 0 0 0 0 0 0 0 0 0 0 0 0 0
Sum Of Errors per net = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

clues = 1
# iteration Homophone Dominant = 7.8854 std = 0.61336 anova=0.99997
# iteration Homophone Subordinat = 15.2283 std = 6.7239 anova = 0.49135
# iteration Hetrophone Dominant = 9.0625 std = 0.81837 anova = 0.93959
# iteration Hetrophone Subordinat = 21.7283 std = 9.726 anova = 0.62212
# iteration Normal Word Dominant = 7.5312 std = 1.0152 anova = 0.99944
# iteration Normal Word Subordinat = 10.3229 std = 1.5861 anova = 0.98357

Hetrophone Dominant is slower from Homophone by = 1.1771 rounds std=0.20956
Hetrophone Subordinat is slower from Homophone by = 6.4871 rounds std=2.1831

round result on iteration on pronunciation Homophone Dominant = 5.375 std = 0.99736
round result on iteration on pronunciation Homophone Subordinat = 5.7391 std = 0.84995
round result on iteration on pronunciation Hetrophone Dominant = 7.0729 std = 0.87353
round result on iteration on pronunciation Hetrophone Subordinat = 13.6957 std = 7.2198
round result on iteration on pronunciation Normal Word Dominant = 6.0208 std = 0.76749
round result on iteration on pronunciation Normal Word Subordinat = 8.0938 std = 1.4222

Hetrophone Dominant is slower from Homophone by = 1.6979 rounds std=0.12453
Hetrophone Subordinat is slower from Homophone by = 7.8547 rounds std=1.498

round result on iteration on meaning Homophone Dominant = 6.9375 std = 0.61237
round result on iteration on meaning Homophone Subordinat = 13.4674 std = 6.8251
round result on iteration on meaning Hetrophone Dominant = 7.8438 std = 0.98759

```
# round result on iteration on meaning Hetrophone Subordinat  = 19.2609 std = 9.8649
# round result on iteration on meaning Normal Word Dominant           = 6.4062 std = 1.0621
# round result on iteration on meaning Normal Word Subordinat  = 8.7812 std = 1.3235

# Hetrophone Dominant is slower from Homophone by = 0.90625 rounds  std=0.21403
# Hetrophone Subordinat is slower from Homophone by = 5.7674 rounds  std=2.0801
 ******************************************************************************************
# Homophone Correct - Dominant -> Subordinat = 96 ; 0 / 96
# Homophone Corrent - Subordinat -> Dominant = 92 ; 0 / 96
# Hetrophone Corrent - Dominant -> Subordinat = 96 ; 0 / 96
# Hetrophone Corrent - Subordinat -> Dominant = 92 ; 0 / 96
# Normal Word Corrent - Dominant -> Subordinat = 96 ; 0 / 96
# Normal Word Corrent - Subordinat -> Dominant = 96 ; 0 / 96

Sum Of Errors Homophone Dominant = 0
Sum Of Errors Hetrophone Dominant = 0
Sum Of Errors Normal Dominant = 0
Sum Of Errors Homophone Subordinat = 4
Sum Of Errors Hetrophone Subordinat = 4
Sum Of Errors Normal Subordinat = 0
Sum Of Errors per net = 1  1  1  1  0  0  1  0  1  0  2  0
Sum Of Errors per net = 0  4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0
0  0  0  0  0  0  0  0  0  0  0
```

Figure 8: part of the log of one of the experiments, on no clue or one clue.

3.7 Testing the Two Models

All tests were performed twelve independent times. In the following report the averages of these tests are shown. The differences between these trials occur in: (I) initialization weights (chosen randomly) and (II) order of training the networks. For each trial, a randomly chosen training set was created using the same examples and frequencies of presentation. Each trial was tested using the same inputs.

4 Correspondence Between Our Networks and Parallel Human Experiments

4.1 Human Experiments (not performed by the author)

The role phonology plays in silent reading by examining the activation of dominant and subordinate meanings of homophonic and heterophonic homographs in the two hemispheres was investigated. A divided visual field paradigm allowing the discernment of differential hemispheric processing of tachistoscopically presented stimuli was used. Heterophonic and homophonic homographs were used as primers in a lexical decision task where the target words either were related to the dominant meaning, to the subordinate meaning of the ambiguous word, or were completely unrelated.

The subjects were asked to read a sentence presented in the middle of a monitor. After the sentence, a homographic word was flashed in the middle of the screen. After 150 milliseconds, another word was flashed to one side of the monitor. If the word were flashed to the Right Visual Field (RVF), the information would be processed by the LH of the human brain. Similarly, if the word were flashed to the Left Visual Field (LVF), the information would be processed by the RH of the human brain. The last word flashed, flashed to either of the visual fields, could be classified as either a non word, a word that supported the dominant meaning of the homograph, or a word that supported the subordinate meaning of the homograph. After the subject saw the word, he was required to answer, by pressing a button on keyboard, if the word shown was a real word or not.

The aim in conducting this experiment was to find if the response times for semantic facilitation of a subject are affected by word classification –whether a word is homophonic or heterophonic. The experiment was designed to see if subjects were affected by whether the word flashed to the LH or to the RH.

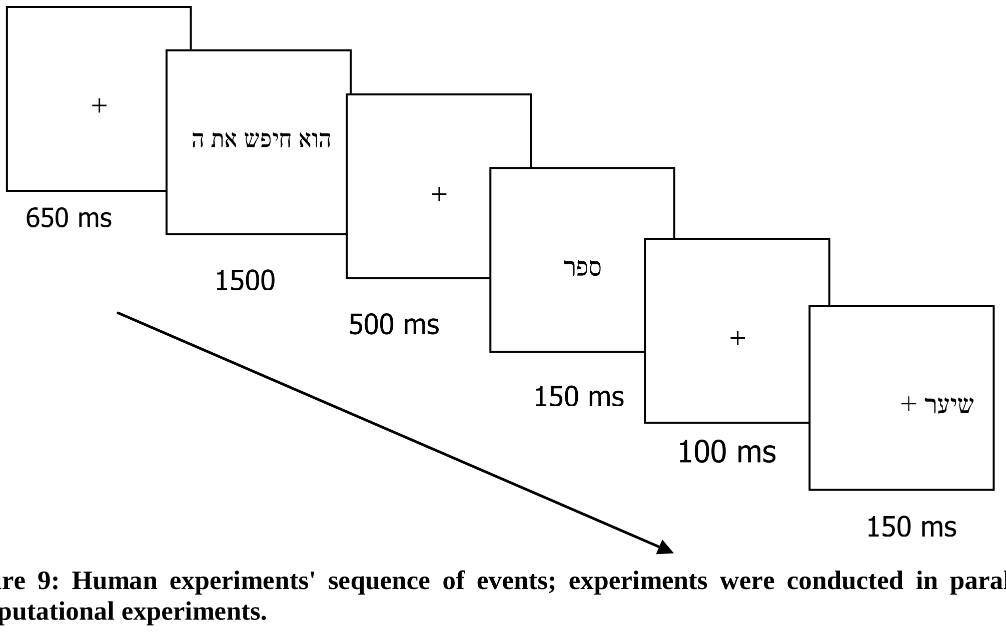

Figure 9: Human experiments' sequence of events; experiments were conducted in parallel to the computational experiments.

4.2 Computational Simulation Experiments

The simulated network aimed to do similar tasks to that of the silent reading experiment in psychology (see Figure 10). A word was presented to the network by inputting the orthography features to the orthography units (as described in section 3.7, Testing). Time was then required to converge to the right meaning. This time was counted in terms of iterations. The tests on the network were performed by presenting the orthography features as in vector and the remaining parts of the vector set to zero. Inputting the bits to orthographic units caused the network to reach a dominant meaning. By inputting the orthography using clues (see Figure 10) that supported a subordinate meaning, the network was made to converge to the subordinate meaning.

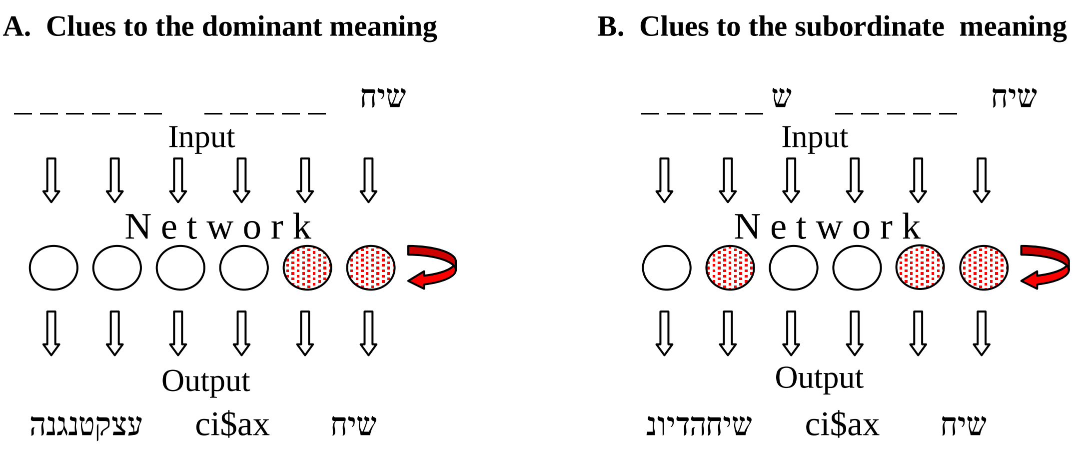

Figure 10: The network was given a vector. The vector was composed of three parts orthography, phonology, and meaning. The network ran until all of its units were saturated, and achieved a stable status. In part A, the network received a vector containing only the orthography part of the vector. Under these conditions, the network achieved the dominant meaning. In part B, the network received a vector containing the orthography part and clues belonging to the orthography sub-vector. Under these conditions, the network achieved the subordinate meaning.

5 Simulation Result and Empirical Findings

If presumptions made prior to the experiment were correct and orthographic codes activate phonological codes directly in the LH and indirectly in the RH, the distinction in processing the two kinds of word types (i.e. homophonic and heterophonic homographs) should have been observed to occur at different stages in processing in the LH and RH. Within the LH, these differences would have been observed in the early stages of lexical access. In the RH, these differences would be seen at a later stage in lexical access

5.1 *Efficiency Asymmetries*

The number of iterations needed for all units of given homographs to become saturated (entire vector) were used as indices of lexical decision times. When homographs were presented, only the dominant meaning was accessed in both networks. Meanings were accessed faster in the LH network. No errors were made by the LH network. The mean error rate in the RH was 9.72%.

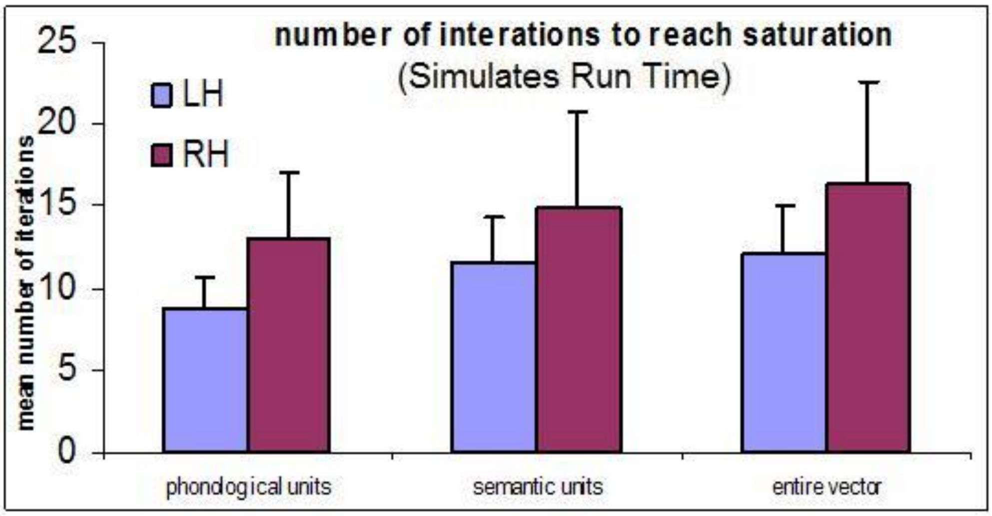

Figure 11 : Superior LH network performance with standard deviation. (Results on the simulated networks)

5.2 *Phonological Asymmetries*

Both homophonic and heterophonic homographs have multiple meanings associated with a single orthographic representation. Heterophonic homographs are phonologically ambiguous. It was assumed that processing differences between these two types of homographs could be attributed to the multiple phonological codes of the heterophonic homographs.

5.2.1 Empirical Data

Thirty-six right-handed native Hebrew speakers performed a lexical decision task on lateralized targets presented 150 or 250 ms after the onset of ambiguous priming (heterophonic or homophonic homographs). The targets were semantically related to the dominant or subordinate meaning of the ambiguous prime or unrelated. Results shown in Figure 12 indicate that, 150 ms after homograph presentation, the multiplicity of phonological codes (for heterophone) affected responses only in the RVF/LH.

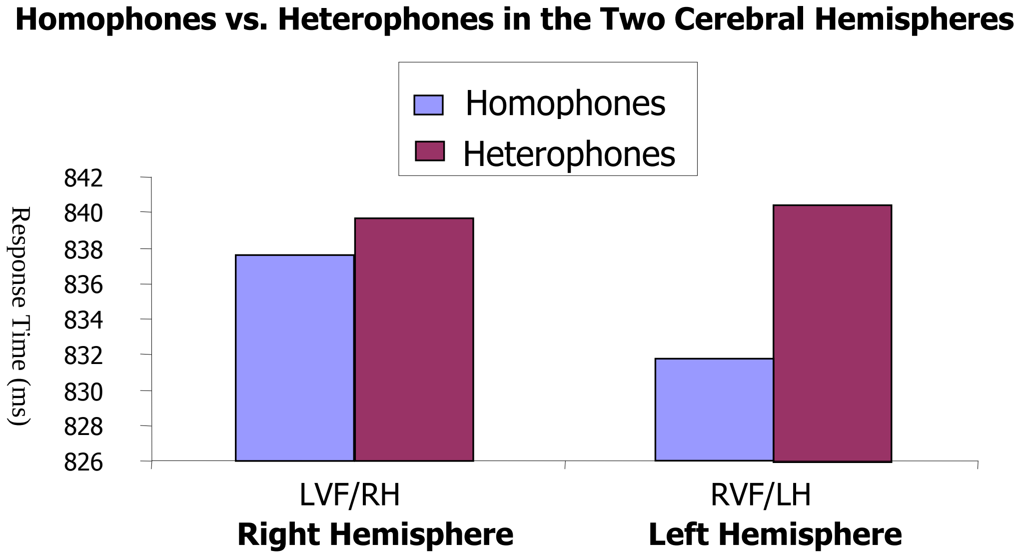

Figure 12 : Phonological Asymmetries - Empirical Data 150ms. The difference between homophones and heterophones is not significant in the RH, but is significant in the LH.

5.2.2 Simulations

Results indicate that heterophonic homographs and homophonic homographs are processed differently by Hebrew readers and by a fully recurrent connectionist network (see Figure 12 and Figure 14). Taken together, this data suggests that phonology plays an important role in silent reading. In the LH network, lexical access was longer for heterophones than for homophones due to the time-consuming competition between the two phonological representations of the heterophones (see Figure 14). According to this model, when a word was viewed, orthographic units were activated first. This activation spread to the phonological and semantic units. Because heterophonic homographs have different pronunciations, these homographs involved the mapping of a single orthographic code onto two phonological codes, as well as the mapping of a single orthographic code onto two semantic codes. As a result, of this one-to-two relationship between orthographic and phonological codes, the speed of lexical access was slower for heterophonic homographs than for homophonic homographs when no context was presented. This is consistent with the idea that in the LH, phonological information guides early stages of meaning activation. In the RH network (see Figure 13), processing times of homophones and heterophones are similar, consistent with the idea that the RH lacks the capacity to convert orthography to phonology directly. Thus, meaning activation is more influenced by orthography.

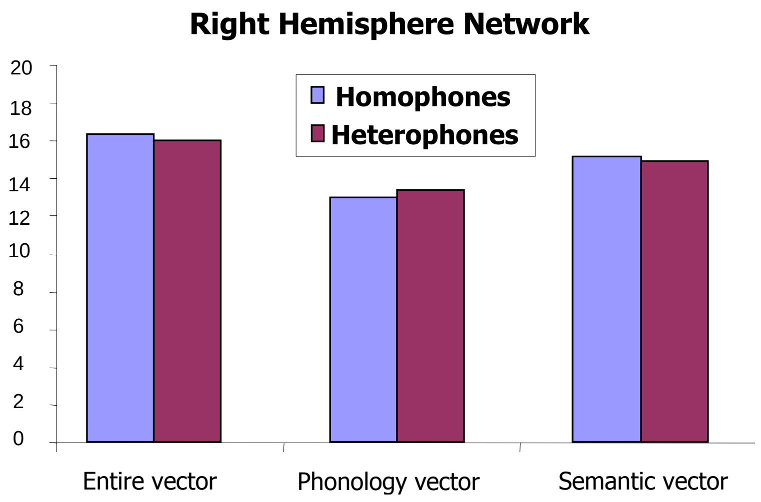

Figure 13: Phonological Asymmetries in the RH - Simulations. The differences between homophones and heterophones are not significant. (result on the simulated network)

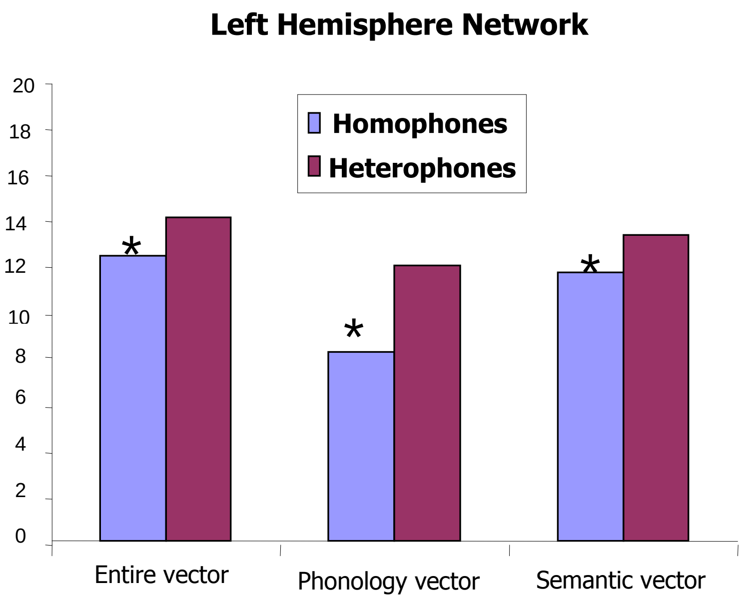

Figure 14 : Phonological Asymmetries in the LH - Simulations The differences between homophones and heterophones are significant. (result on the simulated network)

These results converge with a large body of neurological evidence assigning the focus of phonological processing to the LH.

5.3 Lexical Asymmetries (frequency) - The Time Course of Disambiguation

The activation of dominant and subordinate meanings of a given homograph were examined as a function of time.

5.3.1 Empirical Data from Human Experiment

While dominant meanings were exclusively activated in the LH/RVF, both dominant and subordinate meanings were activated in the RH/LVF 250 ms after the homograph was presented (see Figure 15)

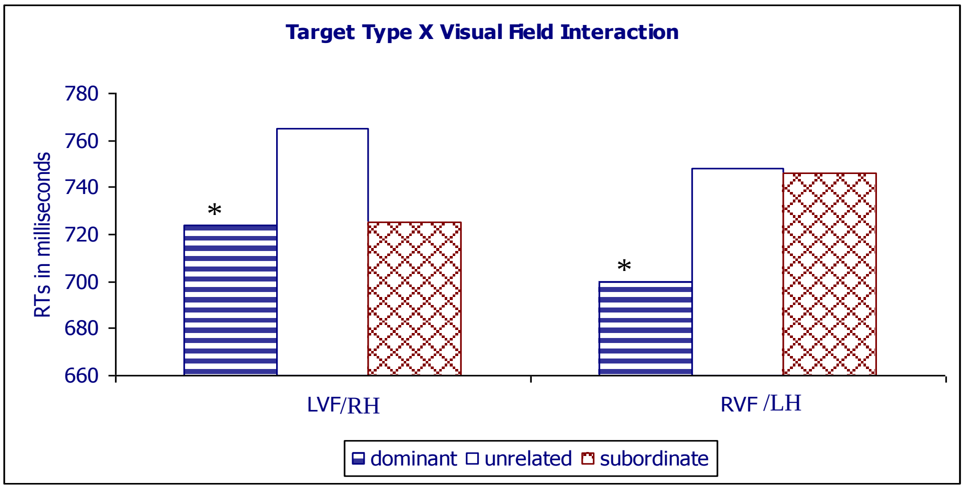

Figure 15: Lexical (Frequency) Asymmetries – Human results, LVF

5.3.2 Simulations

According to Figure 16 the subordinate meaning is suppressed earlier in the LH network than the right. Note that the difference in reaction time from the dominant and subordinate was present only in the LH. Moreover, in the RH both meanings are activated for a longer time as indicated by a reduced run time, while in the LH only the dominant was activated.

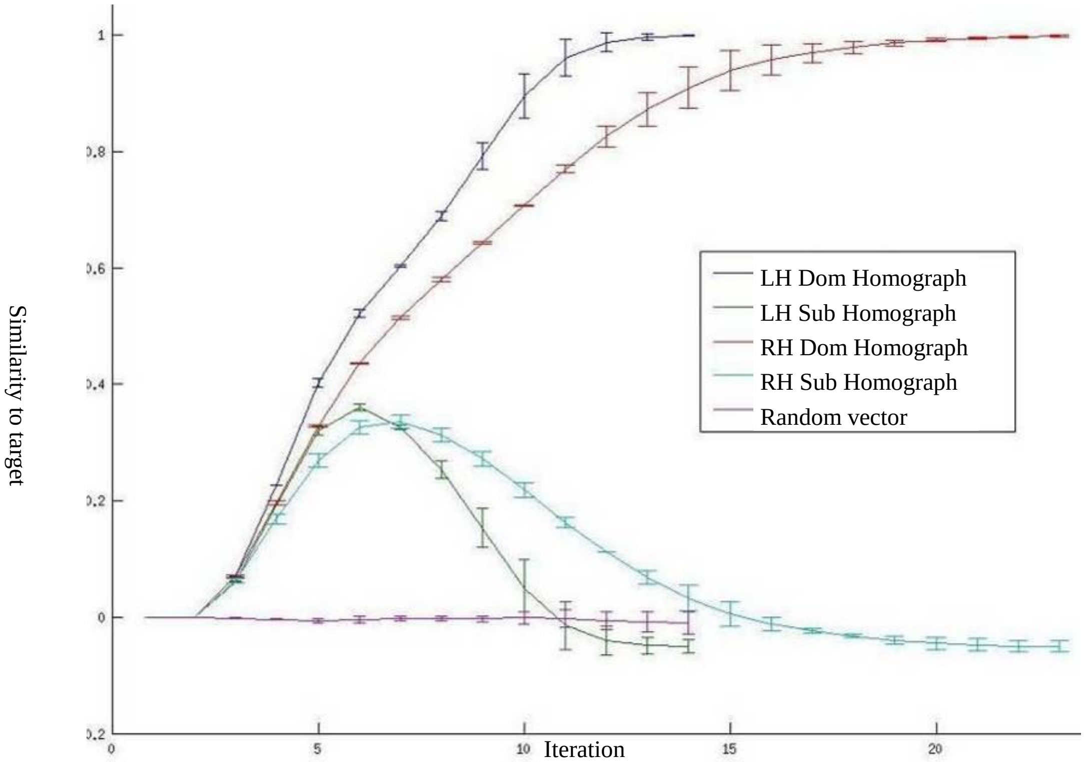

Figure 16: The LH quickly focuses on a single salient meaning, whereas the RH maintains alternate meanings for a longer period of time. (result on the simulated network)

5.4 *Context Effects*

5.4.1 Empirical results (250 SOA):

The absence of a biasing context marks hemispheric asymmetry. This is seen when phonology influences lexical access in the LH but not in the RH (see Figure 12). In this structure, frequency impacts lexical access in the LH but not in the RH. Furthermore, it is the dominant meanings which are exclusively activated in the LH, whereas both dominant and subordinate meanings are activated in the RH (see Figure 15).

When there is a biasing context present, there is hemispheric symmetry, when both context and frequency affect lexical access in both hemispheres. In this, the dominant meanings are exclusively activated in dominant-biasing contexts, while both dominant and subordinate meanings are activated in subordinate-biasing contexts (see Figure 17 and Figure 18).

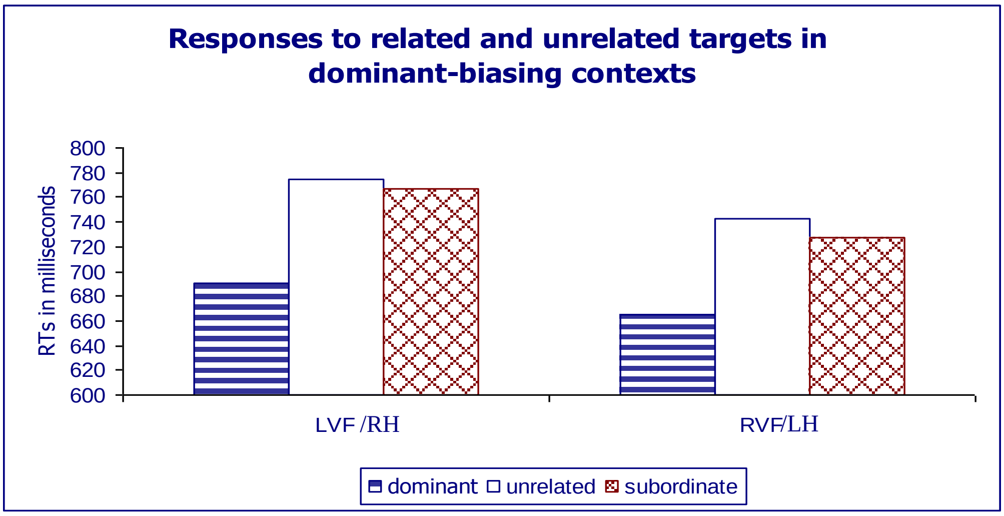

Figure 17: The dominant in the LVF and RVF are statistically significant (Human results)

Note that when context was biased toward the dominant meaning of the ambiguous word, the subordinate meaning was not accessed at all.

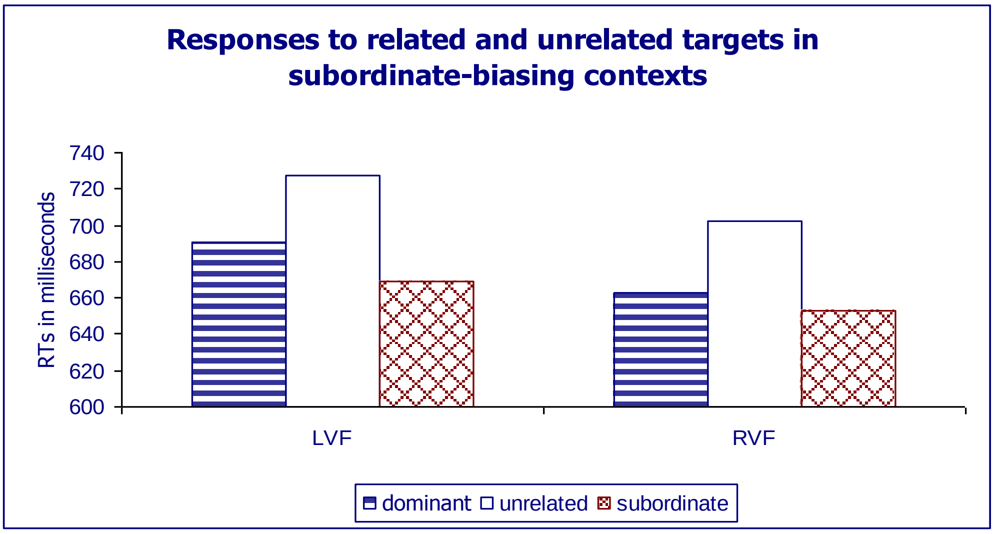

Figure 18: The dominant and the subordinate in LVF and RVF are statistically significant (Human results)

When context was biased toward the subordinate meaning of the ambiguous word, both the contextually compatible subordinate meaning and the contextually incompatible dominant meaning were accessed.

5.4.2 Simulations

Table 2 summarizes the number of iterations needed for the saturation of all units of homographs in the LH and in the RH networks when neither a dominant context, a subordinate context, nor any context is presented.

Context	Homograph LH	Homograph RH
No	14.91	19.37
Dominant	7.42	8.36
Subordinate	13.24	14.27

Table 2: Iteration to 100% Saturation of Entire Vector

Note 9.72% of the non-context homograph did not converge in the RH. All other situation converged

When homographs were presented without a biasing context, only the dominant meaning was accessed in both networks. Meanings were accessed faster in the LH network. This is consistent with LH advantage for lexical processing reported in psychological literature (see Figure 11). In the biasing context condition, there was no significant difference between the two hemispheres.

Table 3 summarizes the time needed to saturate units in the phonological and meaning sub-vectors in the LH and RH networks when neither a dominant context, a subordinate context, nor any context was presented.

	Homograph LH		Homograph RH	
Context	phono	sem	phono	sem
No	8.53	14.09	14.69	18.35
Dominant	6.15	6.19	7.19	6.71
Subordinate	6.85	10.67	9.16	10.45

phono=phonological sub-vector sem=semantic sub-vector

Table 3: Iteration to 90% Saturation of Sub-Vector

In the LH network, phonological information guides early stages of meaning activation. As predicted, phonological disambiguation preceded meaning disambiguation (see Table 3). Contrastingly, in the RH network, difference between phonological and semantics are less pronounced (see Table 3). In the RH, orthographic and semantic sources of information exert their influence earlier than phonological information. Moreover, in the case of no bias, there were many more non-convergent situations consistent with the interpretation that the RH kept both possibilities open longer than the LH (refer to the following section).

When homographs were presented with a biasing context, only the contextually compatible meaning was accessed in both networks. Additionally, dominant meanings in dominant contexts were accessed faster than subordinate meanings in subordinate contexts (see Table 2). This is consistent with empirical results demonstrating that inappropriate meanings are still activated when context is biased toward the subordinate meaning (see Figure 18), but not when context was biased toward the dominant meaning (see Figure 17)

5.5 Conclusions

Overall, the different structure of the two networks produced processing asymmetries comparable to those found in behavioral studies. In these tests, structural differences in the RH resulted in different behaviors in the RH network compared to the LH network. The RH network displayed less accuracy compared to the LH network and needed more iterations to converge.

When presented with a word, the LH network was quicker to converge. The ambiguous resolution was quick. In terms of activities, the network maintained both meanings of a word for a relatively short time and quickly suppressed the subordinate meaning. The RH was slow to converge. In terms of activities, the RH network maintained both meanings for a longer period of time.

In the LH network, orthographic units are directly related to both phonological and semantic units. As a result, lexical access was faster and more accurate. Since orthography is more systematically related to phonology than to semantics, the phonological computation of orthographic representations were faster than the semantic computation of these same representations. Consequently, meaning activation in the LH was initially influenced, primarily, by phonology.

In the RH network, phonological codes are not directly related to orthographic codes and are activated indirectly via semantic codes. This organization resulted in a different sequential ordering of events where the phonological computation of orthographic representations began later than the semantic computation of these same representations. As a result, lexical access in the RH was initially more influenced by orthography.

6 Why We Need Two Hemispheres: Building a Connection Between the Two Hemispheres

6.1 Interactions of the Two Hemispheres

Given the results of the above experiments, the natural question to ask is: What is the computational advantage of this dual processing? More simply, how might an interaction between the two networks in different hemispheres (presumably via the Corpus Callosum) affect the results computationally? Rather than attempt to solve this physiologically or by examining anatomical connections, a more preliminary and abstract question will be addressed: Is there a way, or an example of a circumstance where information from one of the networks might, naturally, be used in the other hemisphere?

6.2 Earlier Models

In S.A.Weems, J.A.Reggia's [27] paper three different network structures (see Figure 19) were used to mimic human behaviors regarding word recognition. They concluded the following:

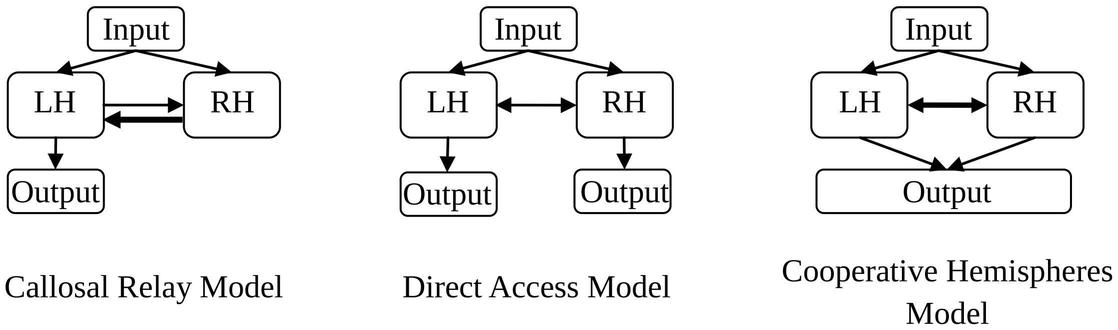

Figure 19: Three computational models are compared: the Callosal Relay Model; the Direct Access Model, and the Cooperative Hemispheres Model. The Callosal Relay Model has strong right to left, but minimal left to right connectivity. There is response output only from the LH. The Direct Access Model has minimal connectivity between the hemispheres, and separate LH and RH outputs. Finally, the Cooperative Hemispheres Model has strong interhemispheric connections, as well as a single output receiving connections from both LH and RH association areas.

S.A.Weems, J.A.Reggia's [27] lateralization of the cooperative hemispheres and Callosal Relay Model led to different outcomes: the Callosal Relay Model showed faster recognition for both words and non-words for RVF trials, whereas the Cooperative Hemisphere Model showed a RVF response latency advantage only for words. The Cooperative Hemisphere Model was the only one to mimic behavioral results in both accuracy and latency performance (i.e., the only model that showed a RVF advantage in both response measures for words only). Contrastingly, the successful performance of the Direct Access Model, regardless of which visual field the stimuli were presented, suggests that this model allowed each hemisphere to become a strong word recognizer.

Monaghan, P. & Pollmann, S. show in their paper [22] that when stimuli have to be matched in a complex task (such as whether two letters have the same name), performance is better when stimuli are presented across the hemispheres of the brain. Furthermore, they argue that for simpler tasks (such as whether two letters have the same shape), better performance is achieved when stimuli are presented unilaterally. The authors show that this bilateral distribution advantage effect emerged spontaneously in a neural network model learning to solve simple and complex tasks with separate input layers and separate, but interconnected, resources in a hidden layer. The authors show that relating computational models to behavioral and imaging data helps to understand hemispheric processing and generating testable hypotheses.

6.3 *Simulation of Collaboration Between the Two Hemispheric Networks*

From this, as well as additional literature on this subject an experiment was planned involving the collaborative interactions of the two cerebral hemispheres. This method was used to help to gain a better understanding of the disambiguation processes of word comprehension between the two hemispheric structures, as each hemisphere has different behaviors and processes to distribute information to units. This model was used to mimic a human reaction of word recognition during the silent reading of words. In doing so, the words' perceptual meaning was controlled using clues after initial recognition and understanding of a word had been reached.

To judge what the possibilities might be, at the time course of activation level of each of the possible resolutions of the lexical ambiguity was used. Figure 20 and Figure 16 gives the graph of both the dominant and subordinate meanings during the time course in both the RH and LH in these models. In the asymptote, the subordinate meaning disappeared in both hemispheres. The time course was different in each.

The subordinate activation in the LH increased more sharply and to a higher degree. It then fell more sharply. This was interpreted to mean that the secondary possibility remained available for a longer period in the RH. In terms of artificial neural network dynamics, this means that during a period (in grey in Figure 20) the RH, while dynamically on its way to the "attractor" corresponding to the dominant meaning, is less "deep" in the attractor well.

To test this hypothesis, the following scenario was conjured. First, the networks would commence with the orthography. If there was no priming, they would start the dynamics toward convergence to the dominant attractor. If there was subordinate priming, then it would start the dynamics towards convergence to the subordinate attractor.

Assuming that during the time indicated by the grey area in Figure 20 additional information was given to the network that the other attractor is appropriate. For example, during reading, this might occur by information coming from the end of a sentence. In the artificial network, this was modeled by assuming that there is new input to the semantical units of the model that biases the results. This is similar to "priming," i.e. giving clues to the network used above. The difference is that this is not done at the initial stage of the network, but during the time course of processing.[3]

[3] There are different technical possibilities as to how the clues should be given; e.g., what percentage of semantical units are primed, whether the priming is one-shot or whether it is continual during the remaining processing, and whether or not to "reset" the other semantical units. We exhaustively investigated the

It was then asked how the networks behave under different conditions. First, as a baseline, how fast the networks react individually under the same external "clues" from the initial position, and how they reacted from the temporal point inside the grey area was observed. This can also be compared with the reaction of the LH, assuming that the LH received information from the RH. Tests focused on the LH because it has an earlier time course, and thus it was anticipated that during the grey area window of time the RH information would help the LH. This was modeled by replacing the values in the LH with values from the RH.

Figure 21 shows the course of activation when clues were presented at times indicated by the arrow.

	Error | Non-convergence (out of 288) | Speed of convergence (Iteration)											
Method	LH only			LH+RH (prior)			LH+RH+Noise			LH+RH (post)		
Dominant to Subordinate	49	129	40.75±3.9	9	81	34.74±6.68	0	24	26.17±8.06	0	0	22.38±4.67
Subordinate to Dominant	0	60	26.7±5.3	0	60	26.48±5.12	0	58	20.85±6.69	0	55	18.1±6.04

Table 4: The network first converged to the dominant meaning and then was given clues towards the subordinate disambiguation, and vice versa

Table 4 shows the results of changes from dominant to subordinate disambiguation, and vice versa. The two rows have similar qualitative results. The number of "errors" and the average time convergence are shown. These are words that did not converge and words that converged to the wrong disambiguation.

For both errors and iterations, the first column shows when the LH received clues to the opposite meaning and did not receive information from the RH. The second column indicates when the external clues were presented to the network *after* transferring the values of the RH and only the specific percentage of semantical units were presented. The values indicated are for four out of eight units being given clues. The use of different numbers of clues changed the values of the errors and the iteration times. The use of different numbers did not change the qualitative order of the columns. The third column shows results of the same tests conducted for column two with the additional uses of noise, with was not present in column two experiment. The fourth column shows the results when orthography and phonology information arrived from the RH to when clues were given to some units and random values given to the rest of the units.

The distinction between column two and four is that in column two the LH received clues to the counter the meaning *after* transferring information from the RH. In column four, the information from the RH is accompanied by the addition of noise[4]. In column two, the LH's values in the "non-clued" semantical units are the same as in the RH prior to the clues arriving. In column four, non-clued units are randomized.

Results in column 1 show that without information from the RH, the LH has substantially worse performance, both in the number of iterations to convergence and in the number of “errors." This justifies having two distinct networks. Column three's

differences, but the bottom line is that qualitatively it does change our results, although there is some difference quantitatively. This is described further below.

[4] Noise is a function that choose small random values between 0.005 to - 0.005 and added that values to the activation level

performance was better than column two's performance. The addition of noise enabled the neurons to be less biased toward the incorrect attractor in column two, whereas in column three there was noise. Technically, column four performed better than column two. This was expected, because non-clued neurons were more biased toward the incorrect attractor in column two, whereas in column four they were neutral.

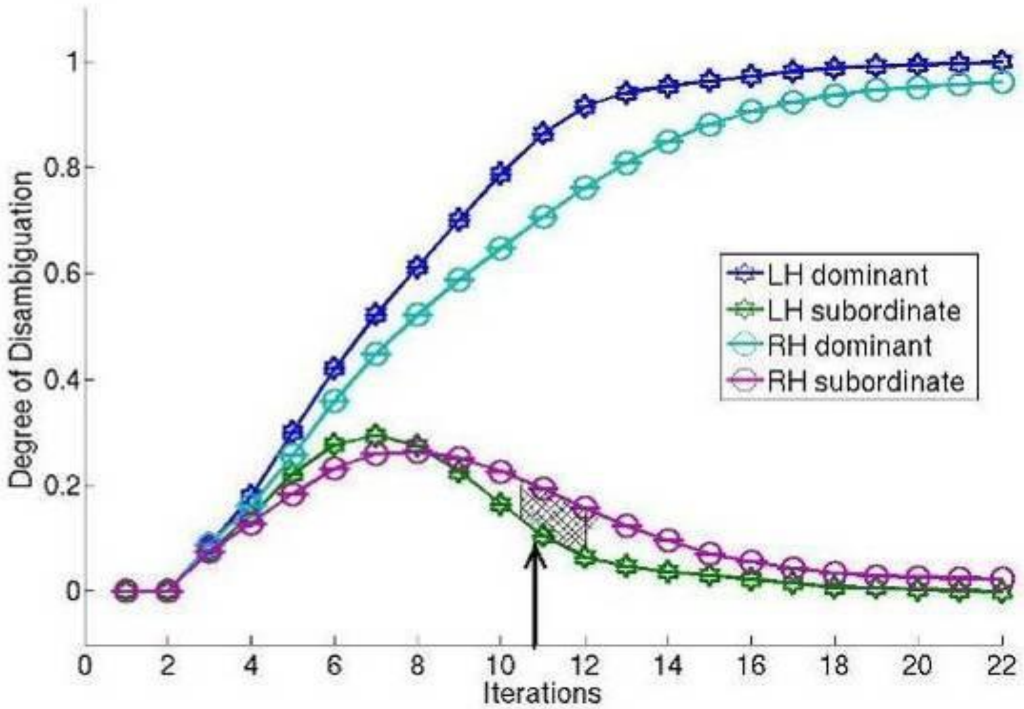

Figure 20: Time Course of Disambiguation No clues. Note grey area – See text.

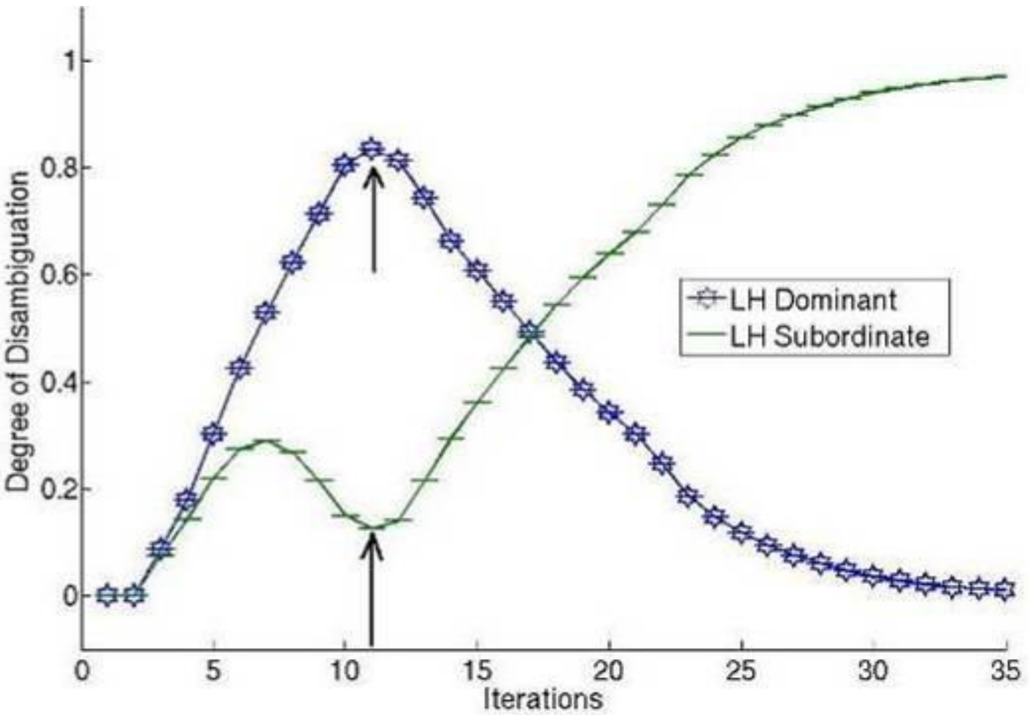

Figure 21: Time course of disambiguation of a homograph when clues were applied at the time of arrow.

6.4 Differences in Disambiguating Homophone vs. Heterophones Words - Examples

When comparing the activation levels during the disambiguation process between homophonic to heterophonic words, the heterophonic words were hard to recover compared to the homophonic words. Unlike the LH the homophonic words, recovered without any difficulties (see Figure 23). With the heterophonic words, the process was much harder (see Figure 22).

Figure 22 shows a successful recovery of both hemispheres on heterophones words. RH successfully recovers only with the clues help. On the other hand, the LH recovers with the clues and the information from the RH. The graph also shows the difference in the activation level of the two hemispheres prior to the clues and the RH information influence, note that in the LH current level of activation without help of the RH the LH was much less chance to recover.

Figure 23 shows a successful recovery of both hemispheres of homophonic words. In this case, information from the RH was helpful and there were no difficulties in the recovery stage.

Figure 24 shows that the LH received information from the RH, but that the information was not sufficient to "change the mind" of the LH. Thus, the LH recovered from the information that it received from the RH and continued to prefer the dominant meaning. This situation was caused from a lack of clues, or from a strong polarity between the ratios of the dominant to subordinate meanings. This situation could have also resulted from RH already deciding on the dominant meaning. If this were the case, the transfer of information from the RH to the LH would have no effect. It appears that clues had more of an influence on homophonic words than heterophonic words.

Figure 25 and Figure 26 show the LH's successful recovery using information from the RH. The clues were not enough to help the RH recover from the dominant meaning (Figure 25 on homophonic words, Figure 26 on heterophonic words). Again, homophonic words were easier to recover compared to heterophonic words.

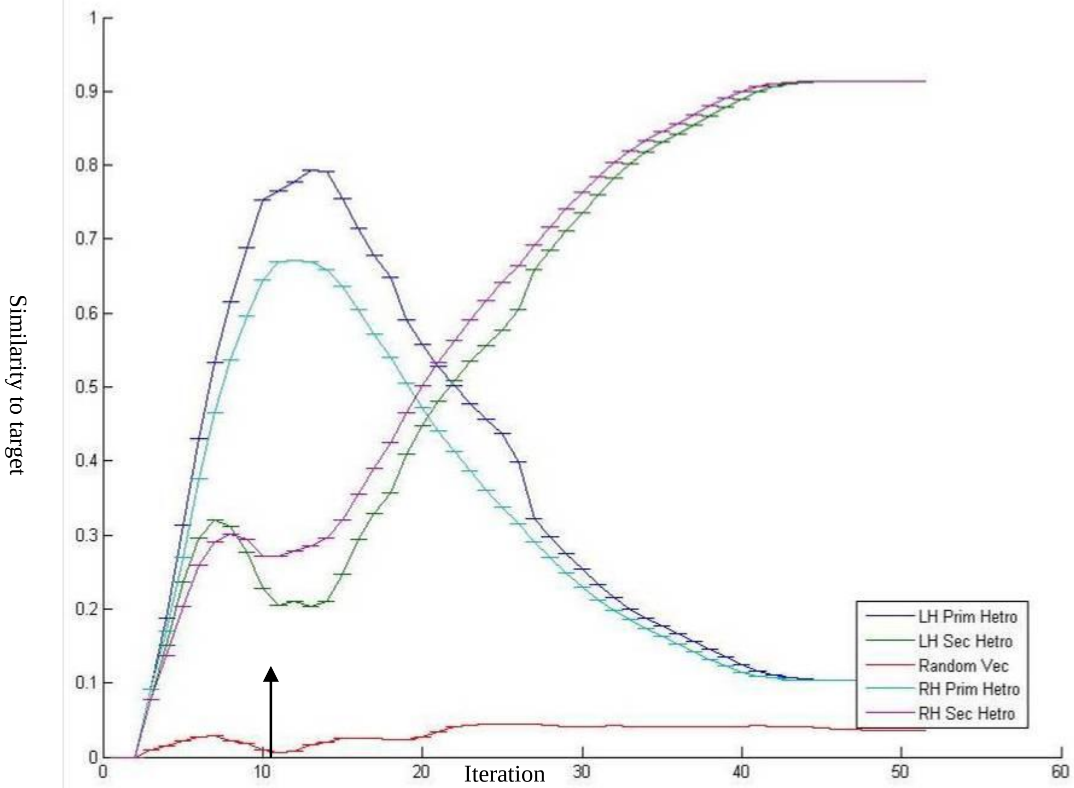

Figure 22: Time course of disambiguation with heterophonic words in the LH and RH when clues were applied at the time of arrow.

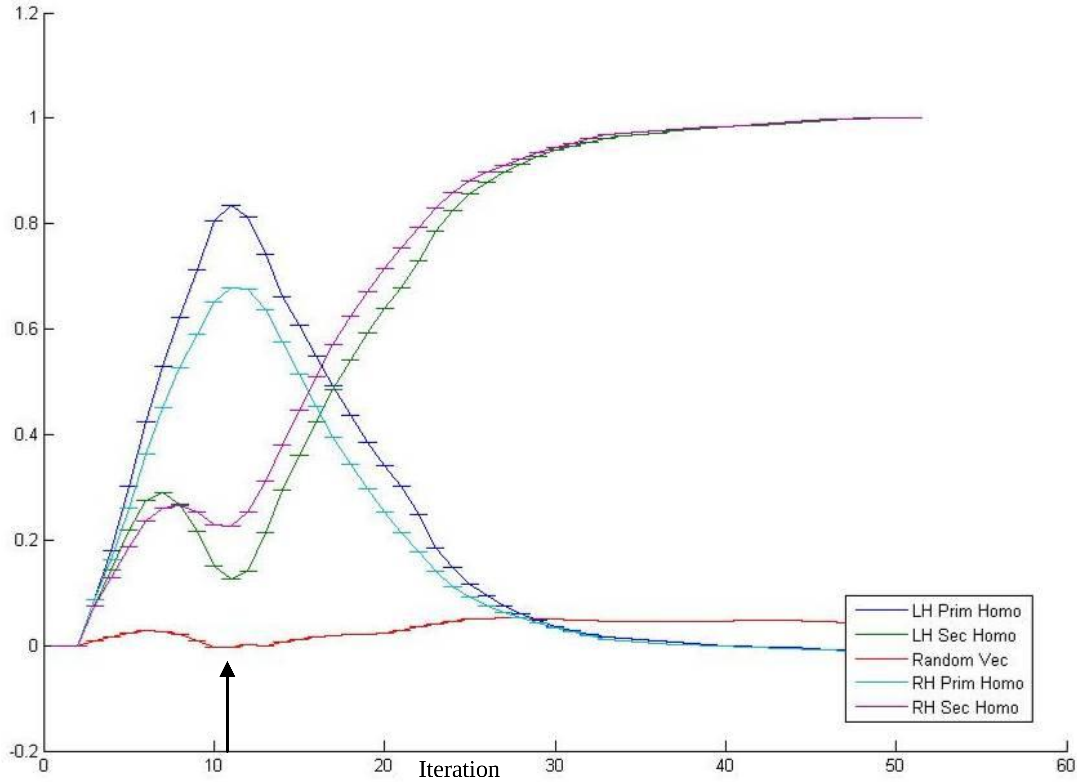

Figure 23: Time course of disambiguation with homophonic words in the LH and RH when clues were applied at the time of arrow.

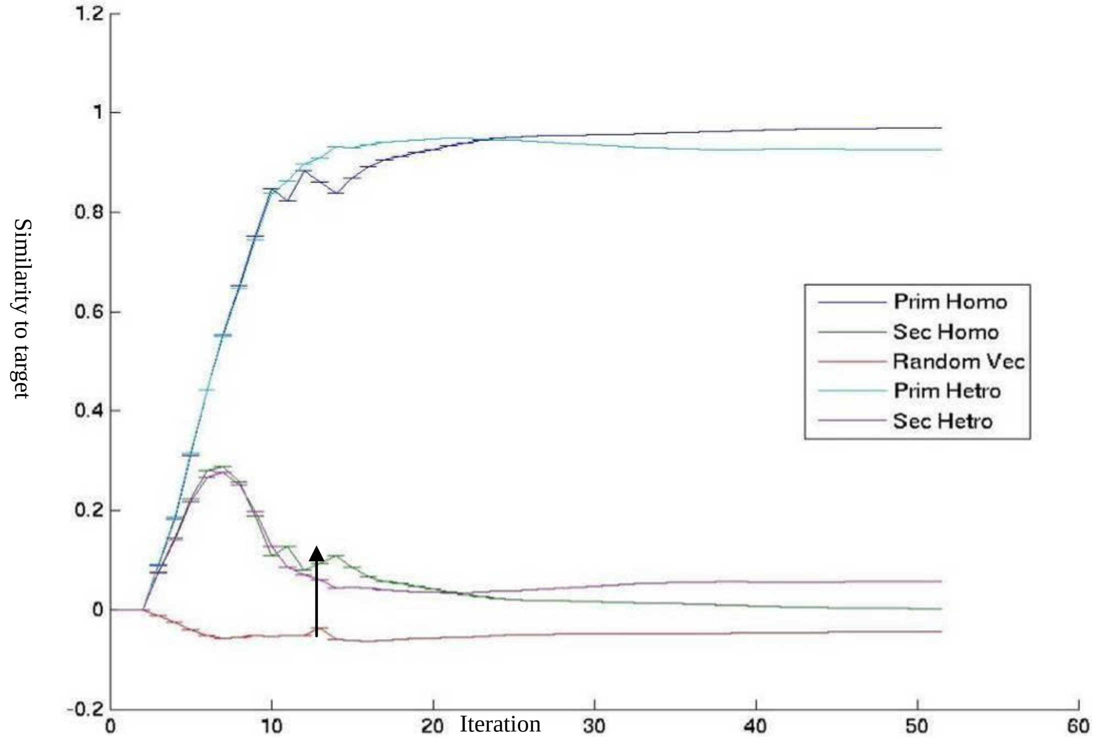

Figure 24: the LH failed to recover from the dominant meaning

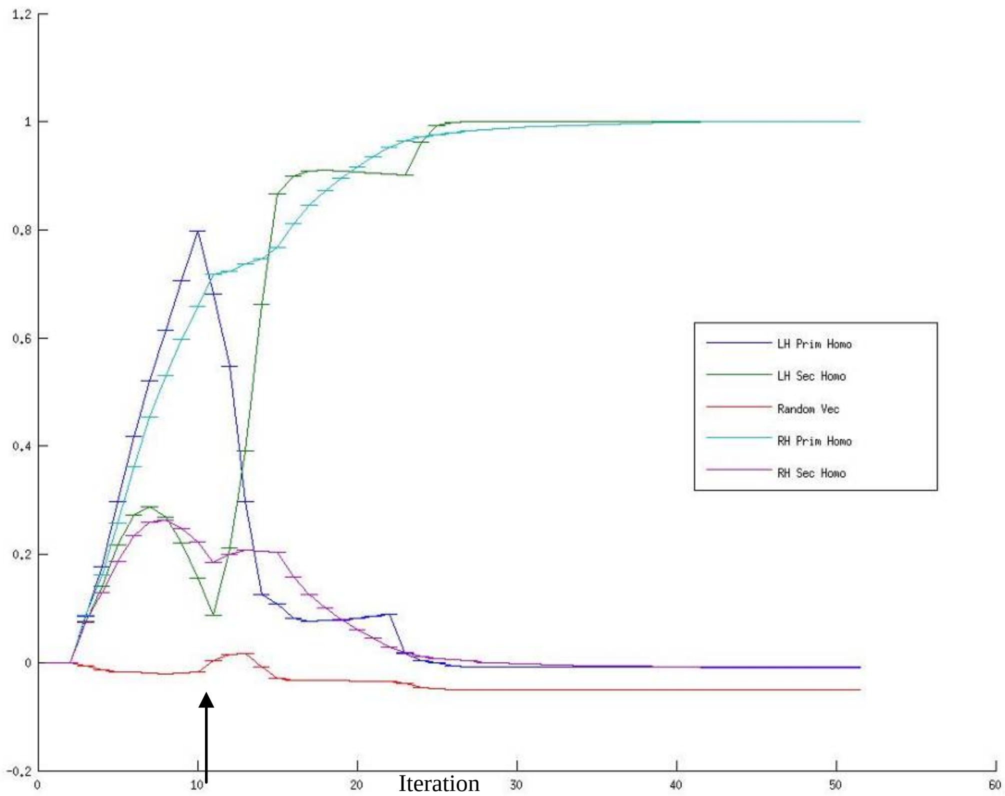

Figure 25: Using homophonic words, the LH successfully recovered using information from the RH. The RH failed to recover from the dominant meaning. The clues were not enough to enable recovery.

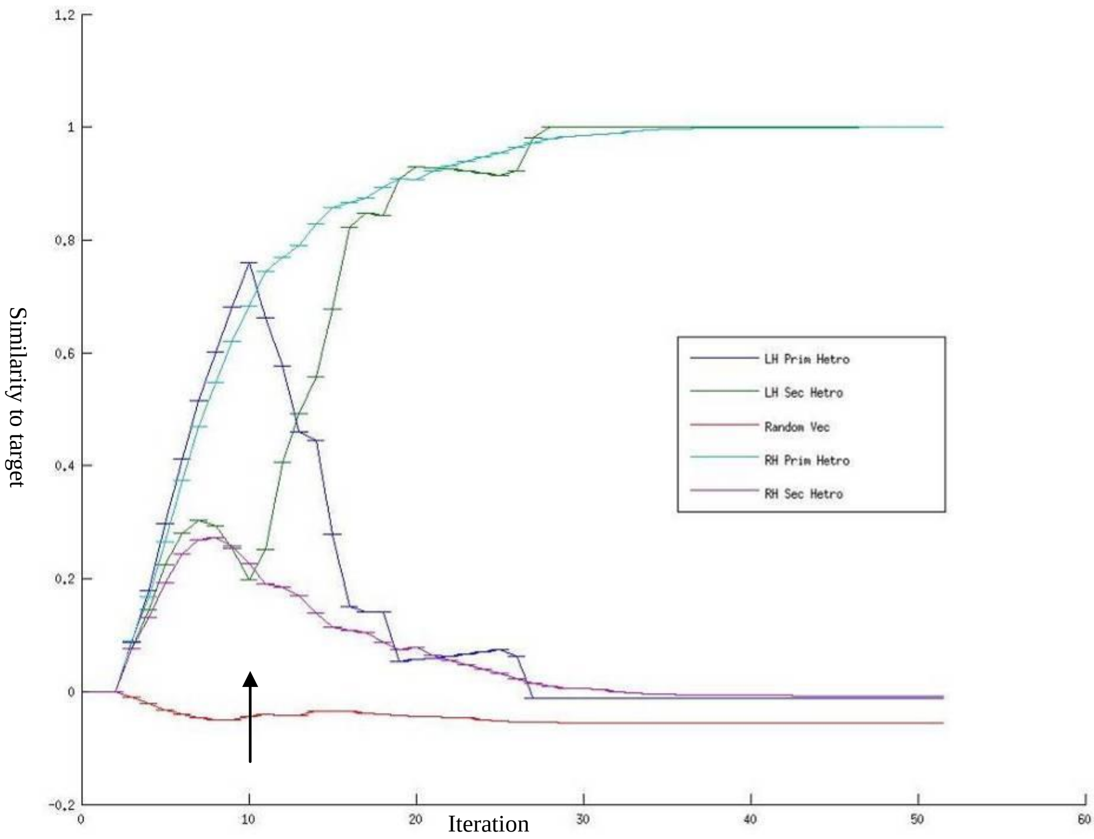

Figure 26: Using heterophonic words, the LH successfully recovered using information from the RH. The RH failed to recover from the dominant meaning. The clues were not enough to enable recovery.

7 Summary

These results have important implications for the role phonology plays in accessing the meaning of words in silent reading. One class of models suggests that printed words activate orthographic codes that are directly related to meanings in semantic memory. Another class of models asserts that access to meaning is mediated by phonology [for reviews See Frost [10]; Van Orden and Kloos [33]]. The results of these tests support the idea that in the LH words are read more phonologically (from orthography to phonology to meaning), whereas in the RH, words are read more visually (from orthography to meaning).

Overall, the two networks produce processing asymmetries comparable to those found in behavioral studies. In the LH network, orthographic units are directly related to both phonological and semantic units. However, because orthography is more systematically related to phonology than to semantics, the phonological computation of orthographic representations is faster than the semantic computation of these same representations. As a result, meaning activation in the LH is initially influenced primarily by phonology. In the RH network, phonological codes are not directly related to orthographic codes and are activated indirectly via semantic codes. This organization results in a different sequential ordering of events in which the phonological computation of orthographic representations begins later than the semantic computation of these same representations. Therefore, lexical access in the RH is initially influenced more by orthography and by semantic.

There is a computational advantage to having these two different networks, such as when a network needs to change after substantial convergence. Based on the results of this tests it is suggested that the LH can converge more quickly than the RH, but at the price of the loss of information when the network "changes its mind." Fortunately, the different time course in the RH allows the LH to recover by copying its information into its network and then proceeding under the LH structure.

8 Recent Results and Improvement on the Networks

In analyzing network training and its results, it was concluded that the training algorithm, in some instances caused the network to over train some words. Each group required a different amount of training. The results of the global repetition time of training led to extra training, and had a negative effect for groups that required less training. The unnecessary extra training of groups led to these groups being over trained. For example, it was found that homophonic words were more difficult, compared to heterophonic words, for the network to learn Similarly, heterophonic words were more difficult for the network to learn than normal words. This information prompted the improvement of methods being used to train the network. The previous set of tests needed 1300 repetitions of words to train the network. By implementing the new algorithm, presented below, training homophonic words required an average of 932 repetitions, heterophonic words required an average of 848 repetitions, and normal words required an average of 802 repetitions. The standard deviation in addition to difficulties associated with the runtime in some of the networks was problematic in the assessment of results.

In the next section, the improved learning algorithm process is described. Another change that will be elaborated on is the lexicon. The new lexicon presented here was changed to contain normal words in order to give a base of comparison during word resolution.

8.1 The New Learning Algorithm

Initializations of the weights between the units start at small, randomly chosen, values (between 0.001 to -0.001). The weights between the units could also initialize at zero to get the same results. After running a series of tests it was concluded that the small random values gave the network a more flexible and dynamic movement from the local minimum in addition to more speed and accuracy in the results. The learning phase contained three steps:

A. First, sets of words were created in random order from the lexicon in homogeneous, uniformed, random order.

B. Second, 48 lines from the sets of words were given to the network and updated the weights between the units accordingly.

C. Third, after training the network with 48 words, from the training set, the network was tested according to the criteria described below to see if more training was needed. If more training was needed, the experiment continued, training the network with new words in groups of 48. If no further training was needed, the experiment proceeded to the testing phase.

8.1.1 Step A:

A training set was created with a repetition of 30,000 entries chosen from the words in the lexicon for each subject. In creating 30,000 entries for the training set, the ratio between the dominant and subordinate words was maintained. The ratio of 5 to 3 was used. This means that where the subordinate words were chosen 3 times, the equivalent dominant words were chosen 5 times. The random choosing of a word needed to be homogenous. Therefore, words could not be repeated until all of the words were used. For example, after choosing one of the dominant words from the training set this word could not be chosen that word again until all of the remaining dominant words were used. Similarly, all of the subordinate words had to be used before a word could be repeated. Had this system not been used there would have been an imbalance in the weights. This imbalance would have given a bias to certain words. According to the ratio, dominant words were chosen more often than the subordinate words.

8.1.2 Step B:

Forty-eight words from the training set were given to the network to study. The learning was made possible by giving the input to all the units and changing the weights, using Equation 1. The training scheme used was an "error-correction" algorithm based on an approach developed by Kawamato [e.g. Kawamoto [15], Rosenblatt [26], and Widrow & Hoff [36]]. To train the network, one of the 48 lines from the lexical entry was presented to the network. This activated the corresponding units in the network and set the activation level to the appropriate value: +1.0 if the feature was present or -1.0 if the feature was absent.

The networks' learning process was achieved by altering the weights between the units in the network. The strengths of the connections to a unit were computed by modifying, to minimize (i.e., correct), the error between its activation level, determined by the environment and the network input. This adjustment in the connection strength from a given unit j to a unit i, ΔW_{ij}, can be expressed in terms of Equation 1

Equation 1: Learning function by changing weights

$$\Delta W_{i,j} = \eta\left(t_i - i_i\right)t_j \text{, where } i_i = \sum_j W_{ij} t_j$$

η Is a scalar learning constant. It is set to 0.00003. (A further elaboration of η is provided in the appendix.) t_j and t_i are the target activation levels of units i and j. i_i is the sum of all other units outputs to unit i. The magnitude of the change in connection strength was determined by the magnitude of the learning constant and the magnitude of the error $t_i - i_i$. The network learned a vector (i.e. one entry) from the training set by inputting the values of the vector into the input of the units and simultaneously adjusting all the weights between the units using Equation 1. This process was used for every vector in the training set.

This learning algorithm was sensitive not only to the input during a particular learning trial, but also to the frequency at which the inputs were presented during the training. Accordingly, the network would receive greater exposure to words that are more frequent in the language. This was achieved by presenting random, homogenous entries from the lexicon, such that the probability that a given entry was presented to the network during the training phase varied according to its relative frequency of occurrence as represented in Table 1: Networks learned during the training phase. The training set was created using this lexicon.

8.1.3 Step C:

The networks learned until the end of the learning set or until they, the units, knew all the words in the lexicon. Additionally, the following two conditions had to be fulfilled, without mistakes, in order for the network to complete the training process.

a) When presented with the orthography part of a word, the network needed to, successfully, choose the dominant word.

b) When presented with the orthography part of a word and with a clue[5], the network needed to successfully choose the correct word, fitting the presented orthography and clue.

Using the 48 words from the training set, the network was checked to see which one of the categories (homophonic, heterophonic, or normal words) of the lexicon succeeded in fulfilling the aforementioned two conditions. If the two conditions were fulfilled by the network in the three categories, the training process was stop and the network was ready to continue to the testing phase. If the network passed without mistakes, in a specific category, it was unnecessary for the network to continue training in that category. Training was stopped using the words of a category the network was successful in. If after the test, it was found that one of the categories that the network had previously been successful in, and consequently stopped training, was found to make a mistake in, failing to meet the aforementioned two conditions; training was resumed in the category it failed, until it successfully fulfilled the two conditions. If after training the network, the network failed to meet, the two conditions training was restarted from the beginning (see 8.1The New Learning Algorithm).

8.2 New Lexicon Representation in the Two Networks:

Each network learned the lexicon consisting of: 16 unambiguous words; 16 pairs of Homophonic Words; 16 pairs of Heterophonic Words, and three-letter words (see Table 5). Each entry was represented as a vector of 256 binary-valued features. The first 48 features represented its spelling. The second 80 features represented its pronunciation. The last 128 features represented its meaning. A vector of 16 bipolar bits represented each character (-1, 1). Combined with different methods of representation, these selections caused the size of the network to be proportional to the vector being taught. For every bit of encoding, a unit would receive and process that bit. After receiving and processing, the values from the units would spread the bit throughout the network (see Figure 6).

Three sets of 27 sub-vectors were created. Each sub-vector corresponded to a letter of the Hebrew Alphabet (see appendix on encoding). Each sub-vector was chosen using a hamming distance function. This function measured the distance between the sub-vectors. This process created an equal distance between all of the sub-vectors. This ensured that each sub-vector would be equally similar, or different, to all other sub-vectors. Three sets of codes were made that represented the three parts of the input and the network. We created a lexicon of words by encoding the characters of each word using the proper encoding tables.

[5] "Clues" were implemented by giving external stimulus to some of the semantics units. See Figure 10 at page 20

These tables were created for orthography, phonology, and meaning. In each part of the vector, a different sub-vector was assigned (see Figure 27). The lexicon represented each letter in a like group utilizing a specific character (see the character ‘e’ or ‘s’ at pronunciation or the character ‘ש’ in orthography in Figure 27), however the same letter was represented in a different group using a different character. (See the character ‘מ’ at orthography and meaning in Figure 27).

The lexicon contained two kinds of words: ambiguous and unambiguous. Each word had a ratio that represented the frequency of the meaning, in ambiguous words, or all parts of a word, in unambiguous words. In the learning phase, we used this lexicon to create a training set that used the frequency as a ratio between the dominant and the subordinate words. Using this ratio in the training set (see Table 5) dominant words appeared more often than subordinate words. In the training set the lexicon learned that the meaning of dominant words were of greater significance than subordinate words.

Note: reducing the size of the vector from 288 to 256 led to the corresponding size reduction of the size of the network.

	Orthography	Pronunciation	Meaning	Frequency
Homophonic Homographe	שיח	ci$ax	עצקטנגנה	5
	שיח	ci$ax	שיחהדיונ	3
	חלל	xalal	חלולאויר	5
	חלל	xalal	נפלבמלחמ	3
	שאף	s$aaf	חמצוננשם	5
	שאף	s$aaf	מטרהרצונ	3
	הגה	$hege	אוטונוהג	5
	הגה	$hege	השמיעקול	3
	בית	bai$t	גרימדירה	5
	בית	bai$t	חלקשלשיר	3
	רגל	regel	הליכהאבר	5
	רגל	regel	אורךמדיד	3
	חבק	xevel	לקשורחוט	5
	חבק	xevel	אזורבארץ	3
	שאל	saa$l	תשובהרצה	5
	שאל	saa$l	ביקשלקבל	3
Heterophonic Homographe	בור	bor	חולחפירה	5
	בור	bur	רפהשכלמח	3
	יוד	yu$d$	פצעטיפול	5
	יוד	yo$d$	אותהקליד	3
	אלף	e$lef	מספרגדול	5
	אלף	a$lef	אותראשונ	3
	שמש	semes	חמהורקיץ	5
	שמש	samas	עובדכיתה	3
	שוק	su$k$	דוכנירקת	5
	שוק	so$k$	ירדשלעופ	3
	דבר	davar	חפצאובקט	5
	דבר	dever	מגפהמחלה	3
	אתר	a$tar	ימבומבקר	5
	אתר	e$ter	מחומרמרד	3
	אמנ	a$man	יוצראמנו	5
	אמנ	a$men	אמרותפלה	3
Normal Words	איל	ayale	חמדברקרנ	5
	סלט	salat	ירקותטעם	3
	קיץ	kaitz	עונהחמהמ	5
	דשא	deshe	עשבירוקח	3
	צבע	zevha	אשלושסוג	5
	קדש	kadsh	סופשענבם	3
	עפר	hafar	אדמהחזרב	5
	פחם	pexam	שחורמלכל	3
	מים	mayim	יסודהחים	5
	כנף	kanaf	ציפורעפה	3
	נפט	nefat	זהבשחורנ	5
	טעם	taham	אחדמהחוש	3
	שקד	shked	פריעץבחג	5
	כבש	kvese	חיהשעירה	3
	דלת	delet	כנסהלחדר	5
	תכף	texhf	עתידקרוב	3

Table 5: Networks learned during the training phase. The training set was created using this lexicon

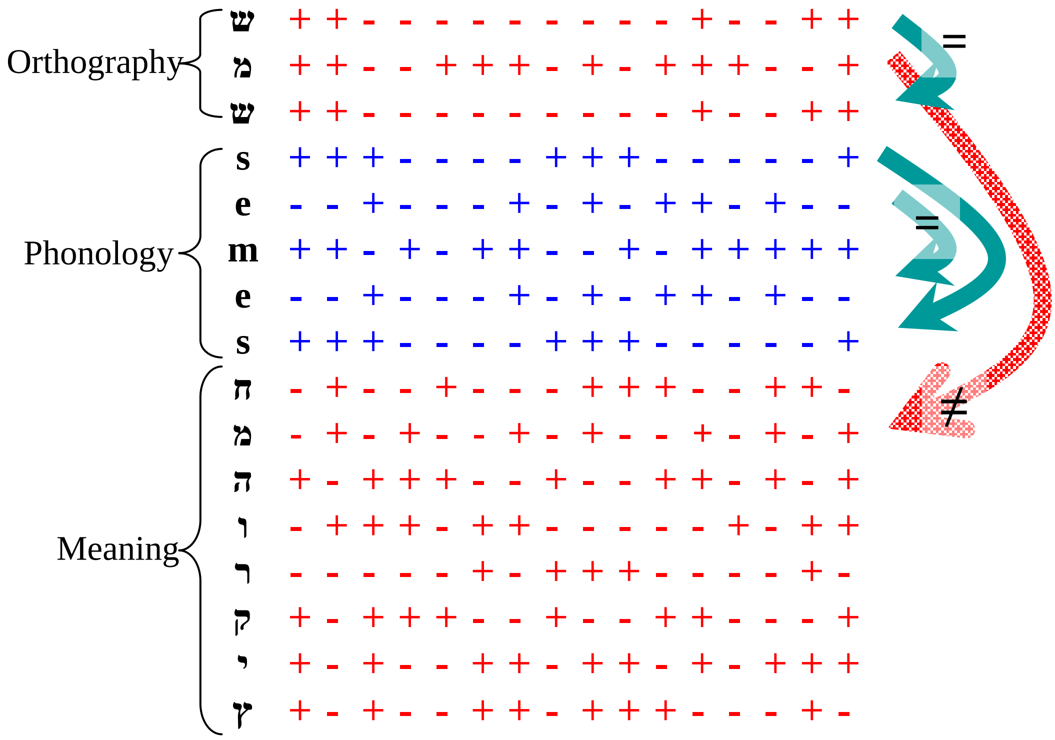

Figure 27: Representation of a vector, from one entry from the lexicon. Each part has a different encoding. Different vector encoding is illustrated by a red, or grey, arrow. Equal encoding is shown using green, or solid, arrows. All encoding was carefully chosen to be equi-distant one from the other (red or grey arrow). See appendix for further details.

8.3 Supplement to the Running / Testing Each of the Networks Separately

An improvement was added to the running algorithm, described above (see 3.5 Running / Testing each of the Networks Separately). This improvement is seen in Equation 2:

Equation 2:
Activation function

$$a(t+1) = \text{Limit}\left[\delta a(t) + \left[\sum_j w_{ij}(t)\, a_j(t)\right] + s_i(t)\right]$$

Previously δ is a decay that was constant and set to 0.7. Now it is changed to be a value based on the progress of the network, which functioned to either strengthen or weaken the influence of the unit's previous values in its current iteration. As the network progressed in processing the information over the iterations, the decay value increased. This caused the network to be influenced by the unit's value from the previous iteration. These results regarding the unit's remembering of the previous value steered the network as a whole. As the network progressed in processing the information the decay value increased. In doing so, the network was strengthened from the previous value and had more memory of previous progress. This concept will be further explained in the appendix.

The decay rate (δ) is necessary not only to help the network get out of an undesired situation (i.e. local minima) but also to help it converge in a more dynamic way. After running a number of tests, which will be explained in more detail in the appendix, we decided to make the decay value dynamic. We discovered that the information flow of the network's structure caused the values of those units lacking input from the external environment to be dependent on the information from units that did receive input. Thus, in the first iterations, the decay rate was 0.6. This meant that within the current iteration, unit i received 60% of its value from the previous iteration. This caused the network to be more dynamic and less fixed in one direction. As the iterations continued, we increased the value of (δ) by relative steps until the end point was reached. At the endpoint, the decay rate value would be one.

8.4 Results of the Recent Empirical Results on 1000 SOA

The following chart displays the disambiguation of homophonic and heterophonic homographs when presented after a context biased toward the subordinate meaning.

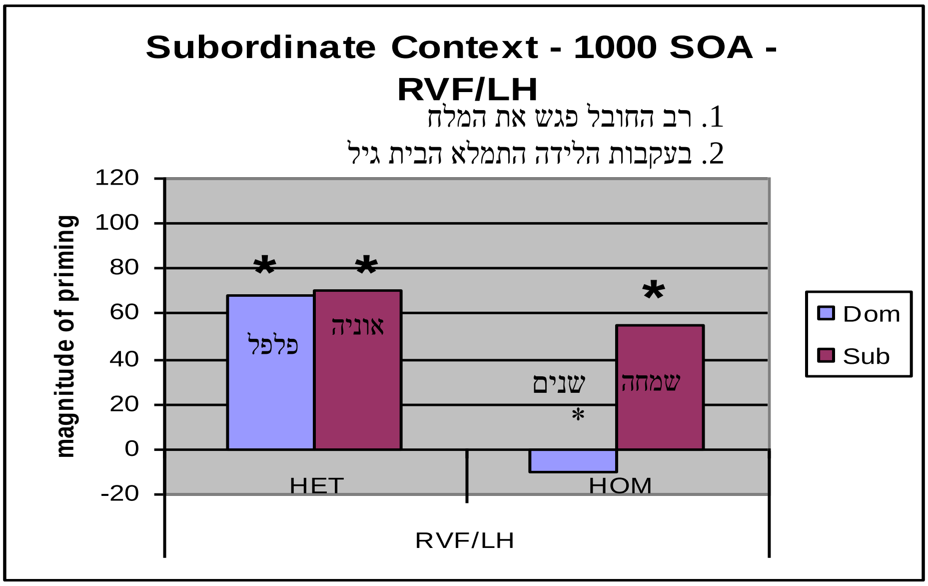

Figure 28 : Human results, priming of activation when subordinate context was presented. The dominant and subordinate meaning of the heterophonic words were activated, in contrast to homophonic words where only the subordinate meaning was activated
Magnitude of priming in milliseconds. – Human results [Reaction Time (RT) unrelated minus RT related]. *= significant difference p<0.05

The most interesting result when a biasing context was given occurred in the subordinate condition at a longer 1000 ms SOA [Peleg, O., Eviater, Z 25]. A significant interaction between VF (RVF/LH Figure 28 vs. LVF/RH Figure 29) and homograph type (heterophones vs. homophones) revealed that heterophones and homophones were disambiguated differently in the two cerebral hemispheres when presented after a sentential context biased toward the subordinate meaning. For heterophones, the dominant but contextually inappropriate meaning was still activated in the LH 1000 ms after the onset of the ambiguous prime, together with the subordinate contextually appropriate meaning. In the RH, however, only the contextually appropriate meaning was activated at the same time. Alternatively, for homophones the reverse pattern was found: The inappropriate dominant meaning was still activated in the RH but not in the LH.

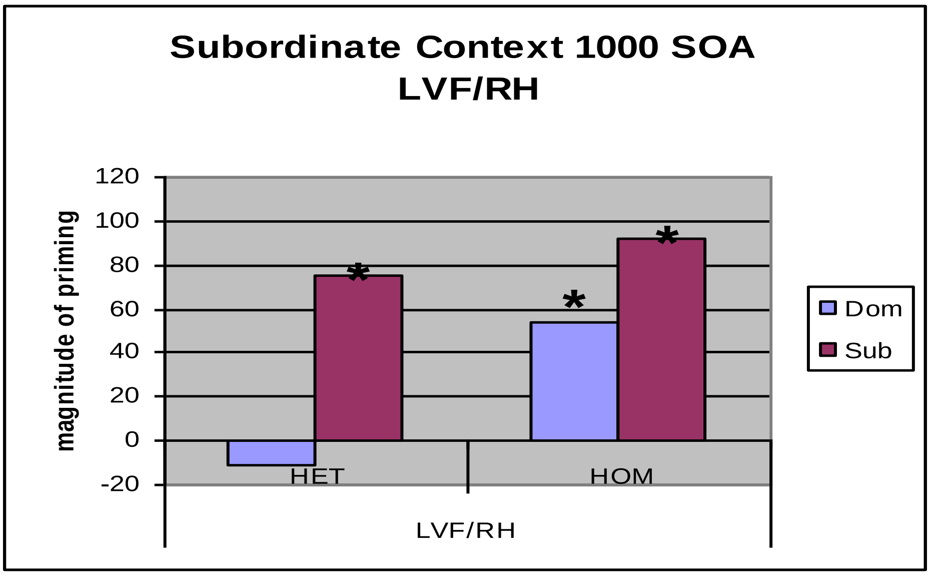

Figure 29: Human results, priming of activation when subordinate context was presented. The dominant and subordinate meaning of the homophonic words were activated, in contrast to heterophonic words where only the subordinate meaning was activated
Magnitude of priming in milliseconds. – Human results [Reaction Time (RT) unrelated minus RT related]. *= significant difference p<0.05

8.5 Results of the Computational Simulation

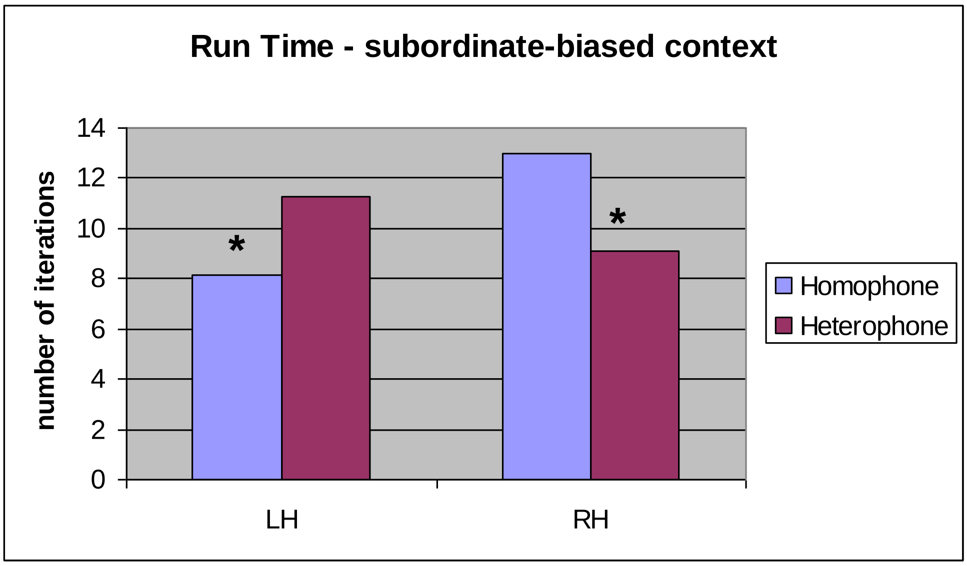

Figure 30: Run time of the simulated RH network. The subordinate heterophone was selected faster compared to the subordinate homophone. Both the homophonic and heterophonic dominant words were selected in roughly the same amount of time. (result on the simulated network)

In the simulation, (see Figure 30) our results were consistent with the empirical results. When the context was biased toward the subordinate meaning, the contextually appropriate subordinate meaning was chosen in most cases (no errors in the LH network and 6.25% errors in the RH network). However, the run time was different for heterophones and homophones as a function of the network's architecture. Heterophones were easier to disambiguate in the RH, relative to homophones, and homophones were easier in the LH relative to heterophones. Under the assumption that total run time is longer because of a longer period of competition between the two alternatives, these results mirror empirical results.

8.6 Additional Results of the Computational Simulation

Corresponding to the results in section 6.4 the recent results on differences in disambiguating homophone vs. heterophones words are similar to the previous one. However, the recent results are substantially and smaller standard deviation in the run time and similarity measurements.

Figure 31 shows the time course of activation in the RH with the new lexicon. Normal words were chosen first to compare to homographs, and there is not much difference between homophone words to heterophones words in the activation level. Eventually the heterophones were chosen before homophones

Figure 32 shows the time course of activation in the LH with the new lexicon. Normal words were chosen first to compare to homographs, the difference between homophone words to heterophones words in the activation level is that the homophones words were chosen before heterophones words

Figure 33 and Figure 34 shows the activation level with the standard deviation in comparison between LH and RH on homophone and heterophone words correspondingly. It is shown that the RH is slower in choosing meaning compare to the LH, and the standard deviation is higher for heterophones words because of the bigger struggle between the two meanings of one heterophone.

Figure 35 shows a comparison between LH and RH in the activation level of all homographs with the standard deviation for the activation. Here the standard deviation is much smaller compared to previous figures because it is the results of all homographs (heterophones and homophones).

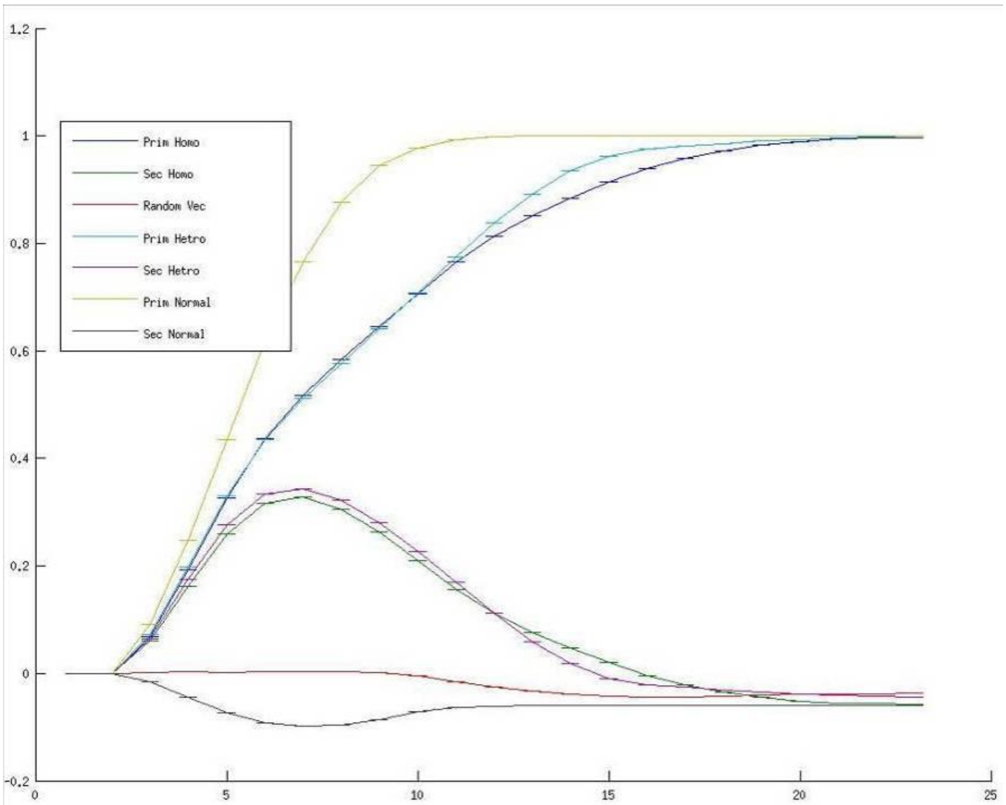

Figure 31: Time course of activation in the RH on all categories using the new lexicon. (result on the simulated network)

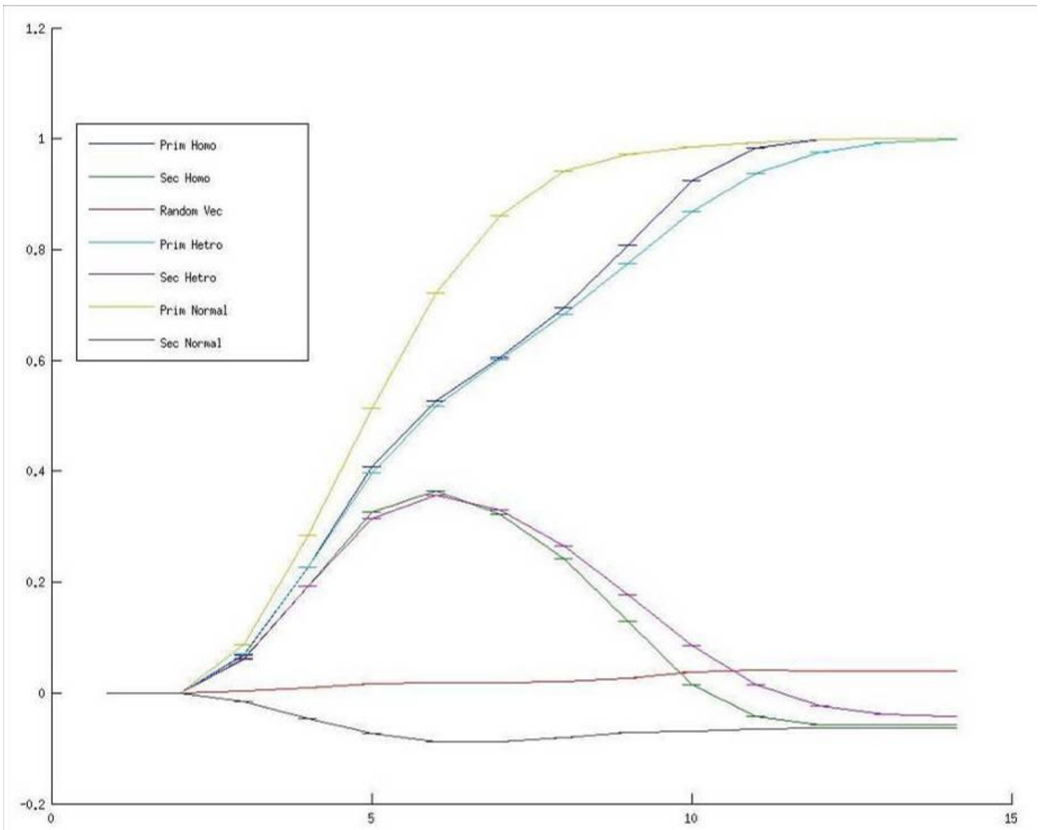

Figure 32: Time course of activation in the LH on all categories using the new lexicon. (result on the simulated network)

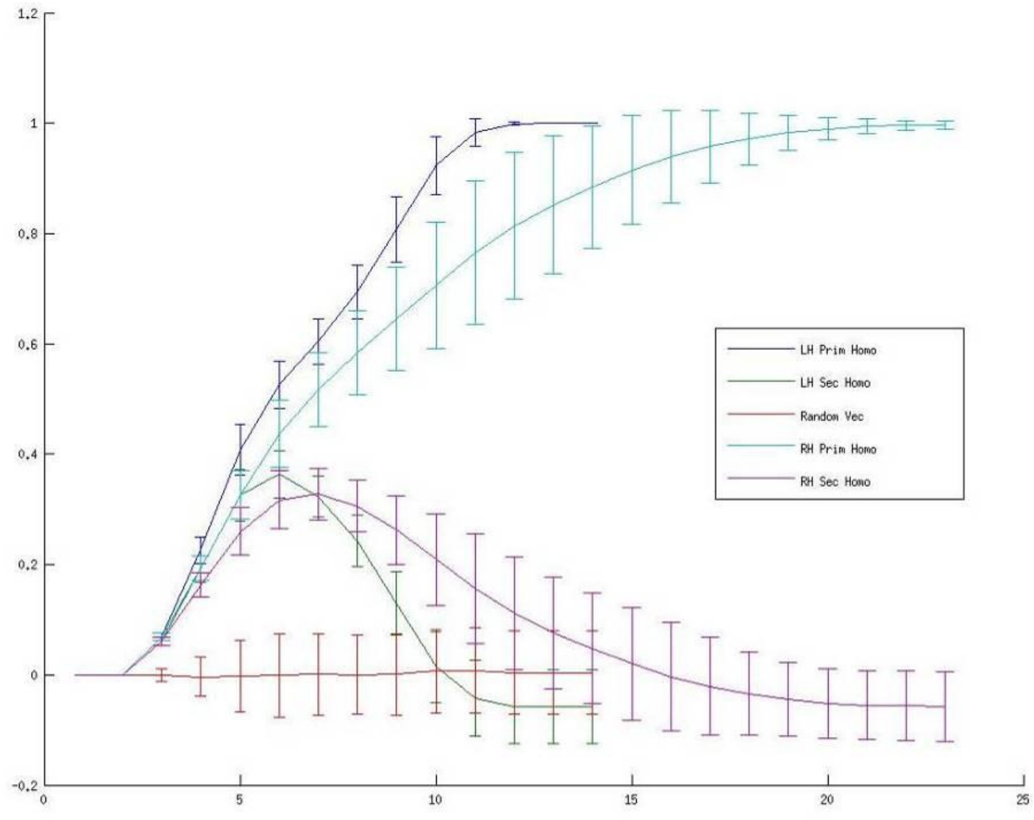

Figure 33: The graph shows the standard deviation for the activation in all subjects. The graph compares the level of activation of homophonic words between the LH to RH. (result on the simulated network)

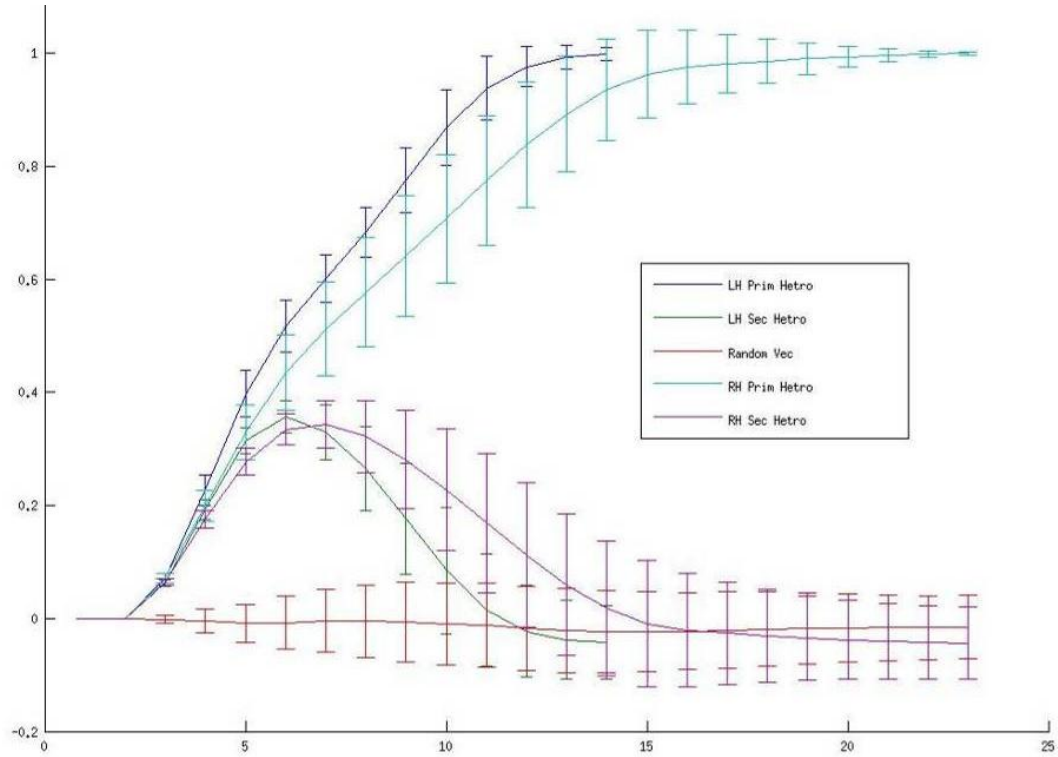

Figure 34: The graph shows the standard deviation for the activation in all subjects. The graph compares the level of activation of heterophonic words between the LH to RH. (result on the simulated network)

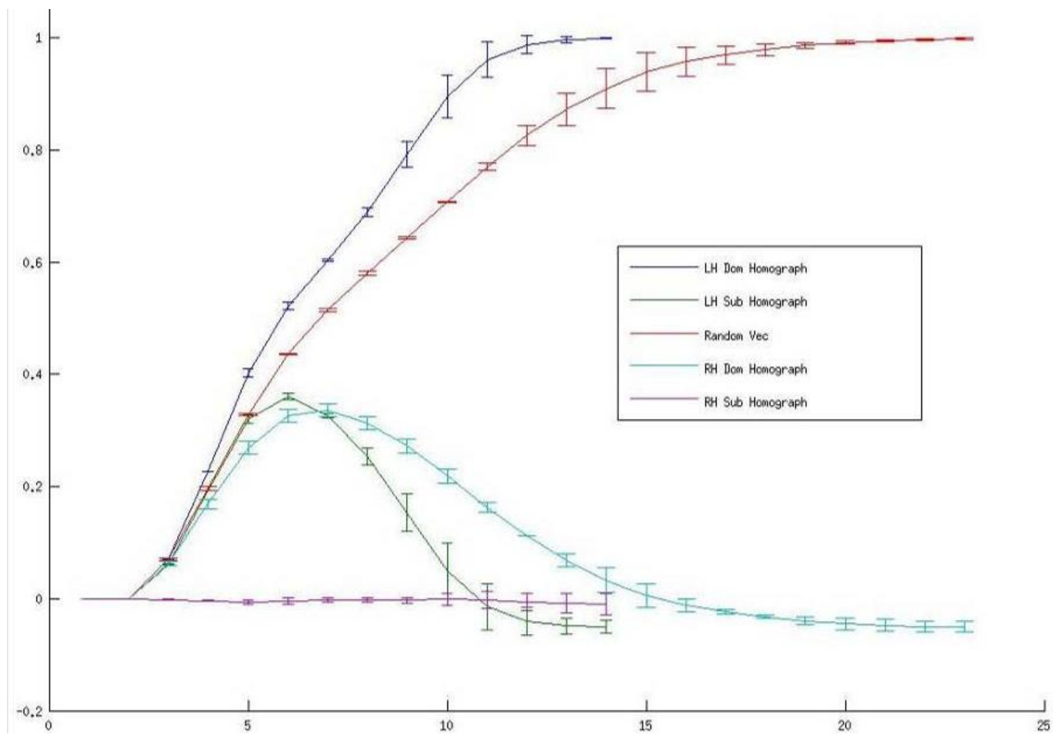

Figure 35: The graph shows the standard deviation for the activation in all subjects. The graph compares the level of activation of homographic words between the LH to RH. (result on the simulated network)

8.7 *Additional Results of the Computational Simulation for which no Comparable Human Experiments have yet been preformed.*

Here show the results of the simulated network for which no comparable human experiments have yet been preformed

Figure 36 and Figure 37 shows the results of the LH and RH networks when dominant or subordinate context are presented to the network. In those results the LH, show an obvious superiority on the RH for homograph in term of runtime. The heterophone in the LH are shown a difference between the dominant meaning to the subordinate meaning, in the contrast to the behavior on the RH network that shows no differences in the runtime of the heterophone. In terms of standard deviation, the RH has more deviation, which indicates that the runtime iteration is fluctuating and the LH is more accurate. The standard deviation that presented in those graphs is the standard deviation.

Figure 38 and Figure 39 shows a different perspective on the results of the LH and RH networks when no context, dominant or subordinate context are presented to the network. The aim this graphs is to compare between the homophone and the heterophone in the same task (i.e. when no context as presented, dominant, or subordinate contest as presented) and to show the differences. It shown as before the superiority of the LH on the RH in term of runtime in all terms of context, and that the homophone was first to select in all tasks. In a contrast the RH show that the homophone was slower in the dominant context but faster when subordinate contest or no context were presented to the network. The standard deviation that presented in those graphs is the standard deviation between the homographs in the same task.

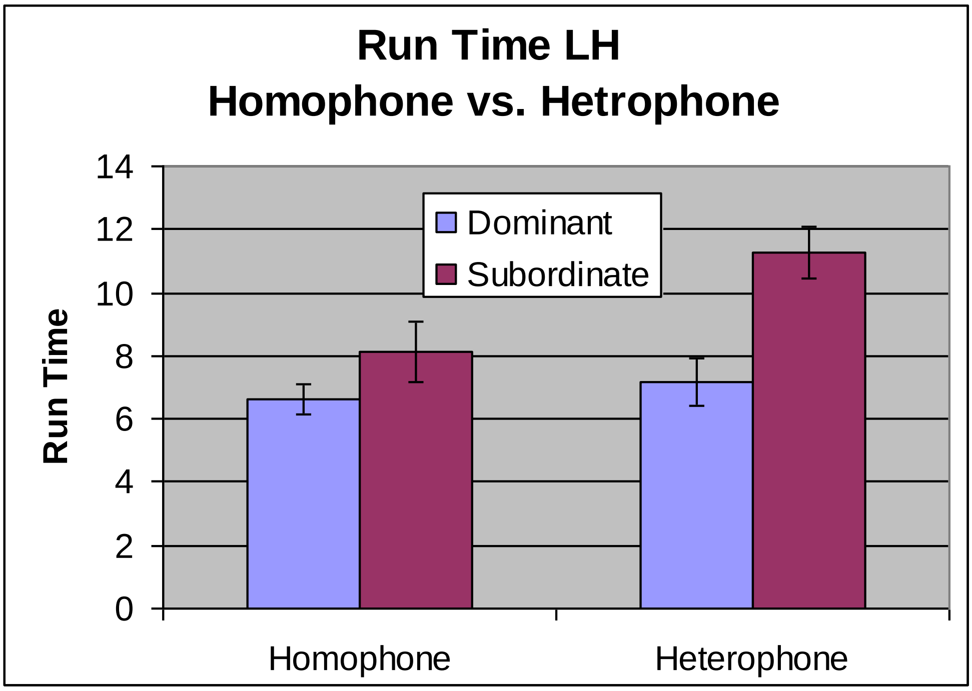

Figure 36: Run time of the LH network

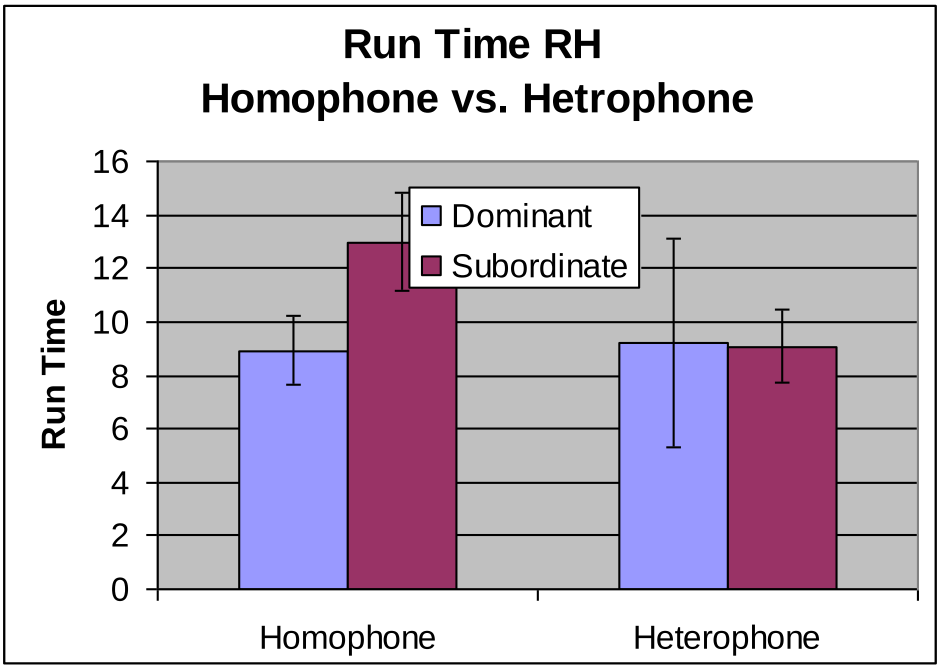

Figure 37: Run time of the RH network.

There results are consistent with what in known from human experiments although this has not been tested at 1000SOA

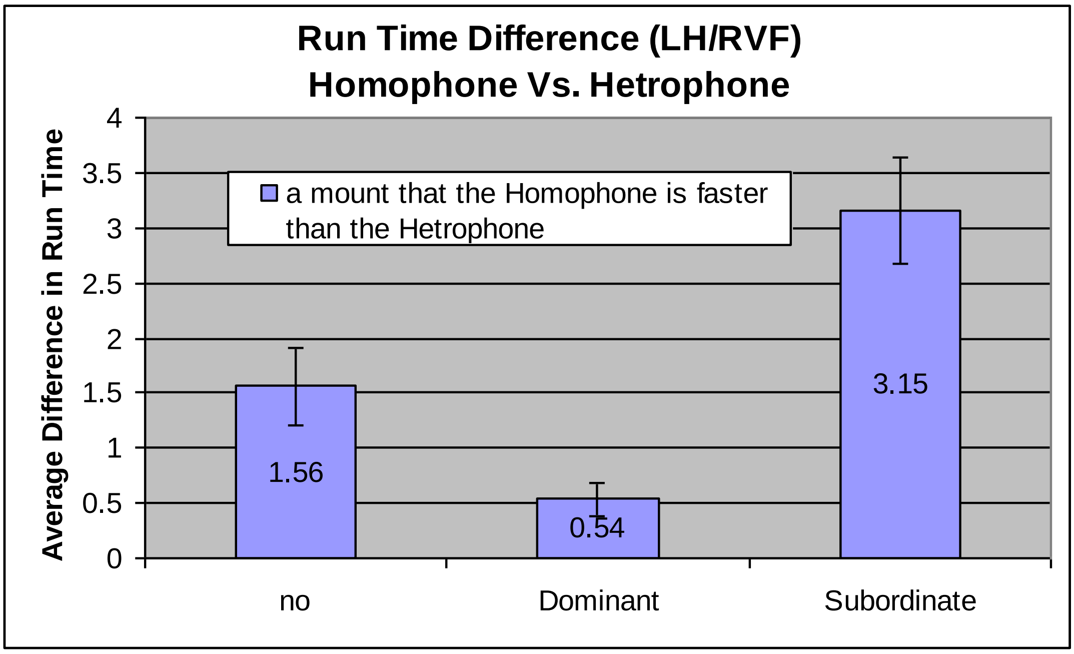

Figure 38: Only the differences in the runtime in the LH network, Homophone was faster in all cases/

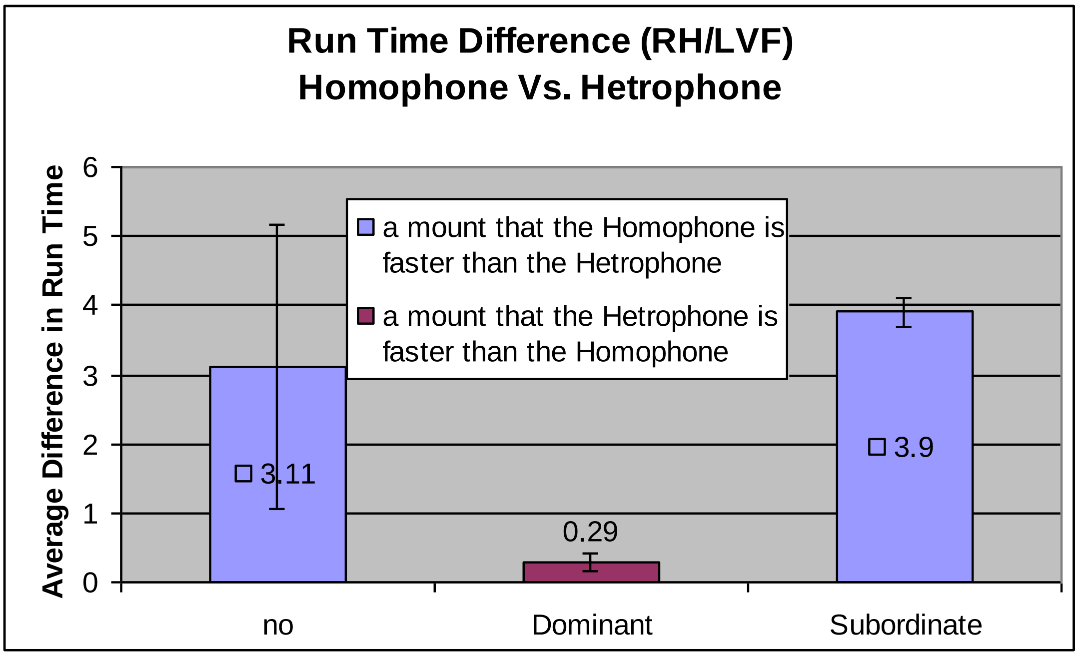

Figure 39: Only the differences in the runtime in the RH network, heterophone was faster only in the dominant

The subordinate results are contestant with what in known from human experiments at 1000SOA, the rest of the results are contestants with what in known from human experiments although this has not been tested at 1000SOA

References

1. Beeman M 1998: Coarse semantic coding and discourse comprehension. In: Beeman M, Chiarello C, editors. *Right hemisphere language comprehension: Perspectives from cognitive neuroscience*. Mahwah (N.J.): Lawrence Erlbaum Associates. 255-284.
2. Beeman, M., Friedman, R., Garfman, J., Perez, E., Diamond, S., & Lindsay, M. (1994). Summation priming and coarse semantic coding in the right hemisphere. *Journal of Cognitive Neuroscience*, 6, 26–45
3. Berent, I., & Perfetti, C. A. 1995: A rose is a REEZE: The two cycles of phonology assembly in reading English. *Psychological Review, 102*, 146-184.
4. Burgess, C. & Lund, K. 1998. Modeling cerebral asymmetries in high dimensional space. In M. Beeman & C. Chiarello (Eds.), Right hemisphere language comprehension: Perspectives from cognitive neuroscience (pp. 215-244). Mahwah, NJ: Lawrence Erlbaum Associates
5. Burgess, C. & Simpson, G. B., 1988: Cerebral hemispheric mechanisms in the retrieval of ambiguous word meanings. *Brain and Language, 33*, 86-103.
6. Chenery, H.I., Copland, D. A., & Murdoch, B.E. (2002). Complex language function and subcortical mechanisms: Evidence from Huntington's disease and patients with non-thalamic subcortical lesions. *International Journal of Language and Communication Disorders*, 37(4), 459-474.
7. Faust, M., & Chiarello, C. (1998). Sentence context and lexical ambiguity resolution by the two hemispheres. *Neuropsychologia, 36*, 827-836.
8. Faust, M.E., & Gernsbacher, M.A. (1996). Cerebral mechanisms for suppression of inappropriate information during sentence comprehension. *Brain and Language*, *53*, 234-259.
9. Frost, R. & Bentin, S. 1992: Processing phonological and semantic ambiguity: Evidence from semantic priming at different SOAs. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18,* 58-68.
10. Frost, R. 1998: Toward a strong phonological theory of visual word recognition: True issues and false trails. *Psychological Bulletin, 123*, 71-99.
11. Grindrod, C., & Baum, S. (2003). Sensitivity to local sentence context information in lexical ambiguity resolution: Evidence from left-right hemisphere damaged individuals, *Brain & Language*, 85, 502-523.
12. Harm M. W. & M. S. Seidenberg (in press). Computing the meanings of words in reading: cooperative division of labor between visual and phonological processes. *Psychological Review*.
13. Hopfield J. J 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Science, USA, 79,* 2554-2558.

14. Iacoboni M, Zaidel E (1996) Hemispheric independence in word recognition: evidence from unilateral and bilateral presentations. *Brain Lang* 53:121–140

15. Kawamoto, A. H. 1993: Nonlinear dynamics in the resolution of lexical ambiguity: A parallel distributed processing account. *Journal of Memory and Language, 32,* 474-516.

16. Lavidor, M. & Ellis, A. W. 2003: Orthographic and phonological priming in the two cerebral hemispheres. Laterality, 8, 201-223.

17. Lukatela, G., & Turvey, M.T. 1994a: Visual access is initially phonological. 1: Evidence from associative priming by words, homophones, and pseudohomophones. *Journal of Experimental Psychology: General, 123,* 107-128.

18. Lukatela, G., & Turvey, M.T. 1994b: Visual access is initially phonological. 2: Evidence from associative priming by homophones, and pseudohomophones. *Journal of Experimental Psychology: General, 123,* 331-353.

19. Manevitz, L.M., & Zemach, Y. (1997). *Assigning meaning to data: Using sparse distributed memory for multilevel cognitive tasks*. Neurocomputing, 14, 15-39.

20. Marsolek, C. J., Kosslyn, S. M., & Squire, L. R. 1992: Form specific visual priming in the right cerebral hemisphere. Journal of Experimental Psychology: Learning, Memory, and Cognition, 18, 492–508.

21. Marsolek, C. J., Schacter, D. L., & Nicholas, C. D. 1996: Form specific visual priming for new associations in the right cerebral hemisphere. *Memory and Cognition*, 24, 539–556.

22. Monaghan, P. & Pollmann, S. (2003). Division of labour between the hemispheres for complex but not simple tasks: An implemented connectionist model. Journal of *Experimental Psychology: General*, 132, 379-399.

23. Peleg, O., Giora, R., & Fein, O. 2001: Salience and context effects: Two are better than one. *Metaphor and Symbol, 16,* 173-192

24. Peleg, O., Giora, R., & Fein, O. 2004. Contextual strength: The Whens and hows of context effects. In I. Noveck & D. Sperber (Eds.), *Experimental Pragmatics* (pp.172-186). Basingstoke: Pagrave.

25. Peleg, O., Eviater, Z. 2007. Hemispheric sensitivities to lexical and contextual information: Evidence from lexical ambiguity resolution. *Brain and Language* (in Press).

26. Rosenblatt, F. (1962). Principles of Neurodynamics: Percep Irons and the Theory' of Brain Mechanisms. Washington: Spartan Press.

27. S.A. Weems, J.A. Reggia : Hemispheric specialization and independence for word recognition:A comparison of three computational models. Brain and Language 89 (2004) 554—568

28. Seidenberg, M.S., & McClelland J.L. 1989: A distributed developmental model of word recognition and naming. Psychological Review, 96, 523-568.

29. Swinney, D. & Love, T. 2002: Context effects on lexical processing during auditory sentence comprehension; on the time course and neurological bases of a basic

comprehension process. In: Witruk, Friederici, Lachmann (Eds.). *Basic Functions of Language, Reading and Reading Disability*, Kluwer Academic (Section 2, ch 1, pp25-40).

30. Thivierge, J.P., Titone, D., & Shultz, T.R. 2005: Simulating frontotemporal pathways involved in lexical ambiguity resolution. *Poster Proceedings of the Cognitive Science Society*, 2005.

31. Titone, D. A. 1998: Hemispheric differences in context sensitivity during lexical ambiguity resolution. *Brain and Language,* 65, 361-394.

32. Tompkins, C.A., Lehman, M.T., Wyatt, A.D. & Schulz, R. (1998). Functional outcome assessment of adults with right hemisphere brain damage. *Seminars in Speech and Language*, 19, 303-321.

33. Van Orden, G. C, & Kloos, H. 2005: The question of phonology and reading. In M. S. Snowling, C. Hulme, & M. Seidenberg (Eds.). *The science of reading: A handbook.* Blackwell Pub. 39-60.

34. Van Orden, G. C., Pennington, B. F., & Stone, G. O. 1990: Word identification in reading and the promise of subsymbolic psycholinguistics. *Psychological Review, 97*, 488-522.

35. Weems, S. A., & Reggia, J. A. (2004). Hemispheric specialization and independence for word recognition: A comparison of three computational models. *Brain and Language,* 89(3), 554-568.

36. Widrcw. G.. & Hoff. M. E. (1960). Adaptive switching circuits. Institute of Radio Electronic Show and Convention, Convention Record, Part 4, 96-104.

37. Zaidel, E 1982: Reading in the disconnected right hemisphere: An aphasiological perspective *Dyslexia: Neuronal, Cognitive and Linguistic Aspects Oxford, Pergamon Press* 35: 67-91.

38. Zaidel, E., & Peters, A. M. 1981: phonological encoding and ideographic reading by the disconnected right hemisphere: Two case Studies. *Brain & Language, 14*, 205- 2

39. Duffy, S.A., Morris, R.K., & Rayner, K. (1988). Lexical ambiguity and fixation times in reading. *Journal of Memory and Language, 27,* 429-446

9 Appendix A. Ambiguity Resolution of Homographs

9.1 *Repercussions of Processing Different Kinds of Homographs*

If meanings were accessed directly from the orthographic input, no difference would be found between processing homophonic and heterophonic homographs. If phonology mediated access to meaning, then different effects of context and frequency would be found.

If phonologic disambiguation precedes meaning activation and if it is ordered, meaning that the dominant representation is activated first, then the following would occur:

1. The dominant meanings of heterophonic homographs would be easier to access than the dominant meanings of homophonic heterographs. This is because the dominant phonological representation of heterophonic homographs is associated with one meaning, whereas the dominant phonological representation of homophonic homographs is associated with two competing meanings.
2. The subordinate meaning of heterophonic homographs would take longer to access than the subordinate meaning of homophonic homographs. This would append because the dominant phonological representation activates only the dominant meaning, whereas the dominant phonological representation of homophonic homographs activates both the dominant and the subordinate meanings of the ambiguous word.

9.2 *The Base Model*

The network described, developed by Kawamoto, was implemented with the following changes:

1. The learning rate was changed from 0.01 to 0.001. The constant decay was changed into a growing decay of the exponential rate starting at 0.1 and increasing until 0.95. These changes were made because the network did not converge using the original parameters.
2. The original 48 4-letters words were replaced with 48 patterns representing 24 pairs of polarized Hebrew 3-letter homographs, half-heterophonic and half-homophonic.
3. Thirty-six features (instead of 48) represented the words' spelling and 60 features (instead of 48) represented its pronunciation. This is because the pronunciation included the vowels that were omitted from the spelling. Similar to the Kawamoto Model, each entry was represented as a vector of 216 binary-valued features (18 characters × 12 bits = 216 bit vector).

Twelve identical networks were used to simulate 12 subjects in an experiment. The network was trained using 2000 learning trials. In each learning trial, an entry was selected randomly from the lexicon. Dominant meanings were selected more times than subordinated meanings.

Note: The learning and activation equations are discussed in detail in the learning of the LH and RH model section.

9.3 Running the Simple Model

After the networks were trained, they were tested by presenting the spelling part of the entry as the input. In order to assess lexical access, the number of iterations through the network for all the units in the spelling, pronunciation, or meaning fields to become saturated, was measured. A response was considered errors if the pattern of activity did not correspond with the input or if not all of the units were saturated after 50 iterations.

Kawamoto's test on his network used ambiguity resolution on regular and homographic words. His model was improved by changing his lexicon to include Hebrew encoding. Additionally, more words were added to the homograph in the lexicon. The first lexicon created was composed of homophonic homographs, heterophonic homographs, and normal words. The aim of these tests, which utilized additional words, was to see if a difference exists in ambiguity resolution of homophonic homograph words and heterophonic homographic words. In this approach, the goal was to identify if there is a difference in convergence between sub-vectors at the homograph in different tests (input: orthography only or orthography and clues). The conducted tests on the network (see Figure 10) were similar to the tests conducted on human subjects. It is important to note that there is no reference to tests like this in current psychological literature.

9.4 Results

Results indicate that when homographs are presented without a biasing context, only the dominant meaning is accessed. Lexical access took longer for heterophonic homographs, than for homophonic homographs (see Table 6: Time to convergence of each sub-vector). These results are consistent with heterophonic homographs disadvantage, which was observed in the LH during the human experiments.

Since these empirical findings reflected initial stages of lexical access (50 ms SOA), the activation of the meaning of homographs was also examined as a function of time.

The phonological status of the homograph (homophonic versus heterophonic) affect the time-course of activation. With heterophonic homographs, the dominant meaning was initially activated more than the subordinate meaning. With homophonic homographs, the dominant meaning is activated slightly more than the subordinate meaning. This data is consistent with the empirical results in where the subordinate meaning was initially more activated for homophonic homographs than for heterophonic homographs.

9.5 Conclusion of the First Goal

mean # of cycles	pronunciation	meaning
Homophonic	9.97	20.38
Heterophonic	19.6	22.54

Table 6: Time to convergence of each sub-vector

Heterophonic homographs and homophonic homographs are processed differently by Hebrew readers and by a fully recurrent connectionist network. Together, this data suggests that phonology plays an important role in silent reading. According to this model, when a word is viewed, orthographic units are the first to be activated. This activation spreads to phonological and semantic units. Since heterophonic homographs have different pronunciations, these homographs involve the mapping of a single orthographic code onto two phonological codes, in addition to the mapping of a single orthographic code onto two semantic codes. Because of this one to two relationship between orthographic and phonological codes, the speed of lexical access is slower for heterophonic homographs than for homophonic homographs when no context is presented (see Table 6). Additionally, the single phonological representation of homophonic homographs allows the subordinate meaning to, initially, be more activated. Although a multiple and parallel search of meaning occurs after the presentation of orthographic information, this search interacts with phonological information.

10 Appendix B. Differences Between LH and the Kawamoto Model

10.1 Introduction

As part of my thesis, I developed a program to simulate the two networks. The first network was the base model, taken from the Kawamoto Model [15]. I used the base model because of it is a fully recurrent connected network that uses simple units in its structure, moreover the similarities that in the experiment that Kawamoto did on his network. On the first attempt, I tried to recreate the Kawamoto's experiment. I then began modifying the model to fit our requirements. The first thing that I did was expand the encoding of the network allow additional character, which required in the Hebrew language. (Hebrew has 27 characters compared to English, which has 26 characters.) The second thing I did was eliminate the 'part of speech' as our tests did not require it. Both of these things were insignificant changes that did not affect the network's behaviors.

From here, we began to adapt the model to our experiment by changing other, larger, things in the base model. Now, we will elaborate on the changes we made, why we made these changes and the results that occurred in making these changes.

10.2 Changing the Encoding of a Character and Size of the Networks

The original lexicon was change into a Hebrew lexicon. The Hebrew lexicon included homophonic homograph words, heterophonic homograph words, and normal words. After conducting tests the original encoding of the network was change from a 12-bit per character to a 16-bit per character. This was done because of similarities in the lexicon entries. This ensured that the distance of each vector would not bias units toward a specific direction, thus leading to altered results. In doing this, the number of units in the network increased from 216 to 256 unit's proportion to the bit per characters (3+5+8) ×16=256-bit entire vector and units.

The maximum distance using 12-bits encoding is exactly 12 vectors. The difference between each exactly 50%. All the vectors that had a difference of 50% or more equaled 24 vectors. This led to the belief the encoding Kawamoto used would not suffice in the experiments. His encoding would most likely bias the results to undesired directions. More, different, vectors were needed. It was preferred that all of the vectors would have the same hamming distance from each other. After considering 'plotkin bound'[6, 7]and 'hamming bound'[7], it was found that there was a coarse bound number of large of n bits. An exhaustive search from 4-bit spaces to 17-bit spaces was conducted. It was found that 16-bits had enough vectors to represent 27 characters with same distance between all vectors (see Table 7). Table 7 shows that the 16-bit sector is the right choice for the encoding

[6] M.Plotkin.*Binary codes with specified minimum distance,* IRE Transactions on Information Theory, 6:445-450, 1960

[7] F.Jessie MacWilliams and Neil J.A.Sloane. *The Theory of Error-Correcting Codes*.North-Holland, Amsterdam, 1977.

n-bits	Number of vectors that diffracts by	
	Exactly 50%	Equal and more then 50%
4	4	8
5	1	4
6	2	8
7	1	8
8	8	16
9	1	6
10	2	12
11	1	12
12	**12**	**24**
13	1	8
14	2	16
15	1	16
16	**16**	**32**

Table 7: This table shows the number of vectors for each 'n' size

For each character, a vector of 16-bits was randomly created and checked against previous vectors to ensure that the distance between each vector was the same, and that there was no vector too close, or too far, from the others. For example, to create a vector for five characters with one, two, or three humming distances from each character, with 4-bits, it would look like Table 8. According to this table, create 27 characters, more than 4-bits would be needed, as the minimum distance that could create 4-bits would be four vectors.

Table 9 contains one of the vector tables used to map and compose characters for the vectors in these experiments to teach to the network. The distance between each vector was an eight hamming distance. This means that there were eight bits exchanged between each vector.

Character	Changes											
	1→25% difference				2→50% difference				3,4→100% difference			
A	0	0	0	0	0	0	0	0	0	0	0	0
B	0	0	0	**1**	0	0	**1**	**1**	**1**	**1**	**1**	**1**
C	0	0	**1**	0	0	**1**	**1**	0	-	-	-	-
D	0	**1**	0	0	0	**1**	0	**1**	-	-	-	-
E	**1**	0	0	0	-	-	-	-	-	-	-	-

Table 8: This table is a representation of a 4-bit character and a possible number of changes that could occur.

א	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
ב	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
ג	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
ד	0	0	1	1	0	0	1	1	1	1	0	0	1	1	0	0
ה	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
ו	0	0	1	1	1	1	0	0	1	1	0	0	0	0	1	1
ז	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
ח	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0
ט	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0
י	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1
כ	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
ך	0	1	1	0	0	1	1	0	1	0	0	1	1	0	0	1
ל	0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1
מ	0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0
ם	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1
נ	1	0	0	1	0	1	1	0	1	0	0	1	0	1	1	0
ן	1	0	0	1	1	0	0	1	0	1	1	0	0	1	1	0
ס	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
ע	1	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0
פ	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1
ף	1	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1
צ	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
ץ	1	1	0	0	0	0	1	1	0	0	1	1	1	1	0	0
ק	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1
ר	1	1	0	0	1	1	0	0	0	0	1	1	0	0	1	1
ש	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
ת	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
$	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0

Table 9: This table is a representation of a set of characters.

10.3 Explanation on the Dot Product Method Used to Plot the Graph

It is necessary to track the networks' progress during the ambiguous process, to do this, the current activation level of the network need to compare to the two possible goals vector and to be measure as a distance function. Because the networks used -1 and +1 as the end values of a vector, the values used during the process were between -1 and +1. The current activation level of the network was measured as a dot product between the status of the network and the target vector (i.e. dominant vector or the subordinate vector). Dot product is an operation, which takes two vectors over the real numbers, and returns a real-valued scalar quantity. It is the standard inner product of the Euclidean space. For example:

#	Vector
A	1 1 1
B	0 0 0
C	-1 -1 -1

The result of the dot product on thus three vectors is:

$$\text{(A to B) is } \begin{bmatrix}1 & 1 & 1\end{bmatrix} \bullet \begin{matrix}0\\0\\0\end{matrix} = \frac{0}{3} = \mathbf{0} \qquad \text{(C to B) is } \begin{bmatrix}-1 & -1 & -1\end{bmatrix} \bullet \begin{matrix}0\\0\\0\end{matrix} = \frac{0}{3} = \mathbf{0}$$

$$\text{(A to A) or (B to B) or (C to C) is } \begin{bmatrix}1 & 1 & 1\end{bmatrix} \bullet \begin{matrix}1\\1\\1\end{matrix} = 3 \rightarrow \frac{3}{3} = \mathbf{1 = match}$$

$$\text{(A to C) is } \begin{bmatrix}1 & 1 & 1\end{bmatrix} \bullet \begin{matrix}-1\\-1\\-1\end{matrix} = -3 \rightarrow \frac{-3}{3} = \mathbf{-1} = \textbf{Totally different}$$

Examples of compression between vectors, to demonstrate the dot product value

The reason that in our reported graphs is no value is near -1 is because of the method (shown on page 57), that always had two distinct encoding vectors with 50% similarity (50% humming distance) equal to 0 on the inner products. Vector that composed using sub-vectors that far from each other by at least 50%. The entire vector should be at least 50% from the other vectors. Thus, 50% distance is equal to 0.

In conclusion: the graphs in this project could go between -1 as a total different, and +1 as a complete match, but a complete correlation to a particular meaning will be measure as 0 from the other one.

10.4 Fixed Decay to Variable Decay

In the original model, the decay value was a constant value. The purpose of the decay value is to give the current unit an influence in the current situation aside from the environment and input of the rest of the units. The status of the units was very important for the output of the units in an iteration. To understand more about the extent of the influence the decay value plays in the units' choosing a response, two experiments were conducted:

1. All units were given a decay value of zero. This means that the units were not influenced from the previous iteration.
2. All units were given a decay value of one. This means that 100% of the previous value of the prior iteration, input from the environment, and sum of all outputs of the rest of the units influenced the unit.

In conducting these two experiments, the following was found:

When the decay value was zero, the output of the units was completely dependent on the input from the environment and the sum of all outputs of the rest of the units from the previous iteration. Therefore, if the decay value was low, less 0.5, and the clues that the network received from the external environment were not enough (i.e. they did not reach the critical mass that made the network jump to the target) the network would fluctuate and continue to change without becoming saturated. This happened because the units did not remember, or were less influenced, by past experiences and more influenced by the output of other units and external inputs. This caused the network to miss its target. The network would miss its target because the units did not possess enough momentum to keep going to the right target. The decay value was the momentum of the units' movement.

When the decay value was one, the outputs of the units were dependent on information from the first steps in the network. As the steps progressed, the unit ignored the influence of other units and the environment. In doing this, the units acted in a stubborn manner, using only their own information from a previous iteration. If the value was stronger (near 1), the units did not forget their past experiences and would either miss the target completely or continue working toward a target that was not within the training set despite other units' information which was capable of steering the unit in the right direction. These units would have too much momentum and would overshoot the target causing the units to miss their targets.

From the two experiments, the decay rate (δ) was necessary to help the network get out of an undesired situation (i.e. local minima) and to help it converge in a more dynamic way. The decay rate (δ) was set to be 0.6 in the firs iteration. This meant that within the current iteration, unit i received 60% of its value from the previous iteration. This caused the network to be more dynamic and less fixed in one direction. As the iterations continued, the value of (δ) was increased using relative steps until the endpoint when the value was 1. When (δ) reached 1 the current iteration unit i would receive 100% of its value from the previous iteration causing the network to be less dynamic and more fixed in its current direction towards its goal. From this information, it was concluded that the units were dependent on their past experiences, as well as information from other units that received environmental input. Coupled with the units influence on each other, the units were directed toward their goals.

10.5 Learning Constant and Training Number

The purpose of η was to correct the value of error between two units. The chosen value of η was very delicate and needed a lot of testing. During the learning phase the algorithm, take one entry from the training-set and inputting it to the network, adjusted the errors from the current weights to the desired value. Every entry that the network learned adjusted the weights to the right value by η. The adjustment process would have a constant value, and the adjustment would come from the lexicon and its frequency. The value of η would not be too small or not too large. The correct adjustment needed to come from the learning process, in which the network learned words. For example, If the value of η was too large the correction error function overestimated the 'target' and consequently missed. Similarly, if the value of η was too small the correction error function required a significant amount of corrections in order to reach the desired 'target'. When this happened, additional training sets were needed to achieve the desired result.

10.6 Creating a Training-Set by Selecting Random Homogeneous Words from the Lexicon

During the training process, an entry from the lexicon was chosen randomly with consideration of frequency (dominant or subordinate). This entry was then input to the units. It was found that it was important to choose entries using a homogeneous and random order. If this would not done, the network would develop a preference to certain words. Using the normal random function that was a standard program within Matlab did not assure that the networks would learn without 'prejudice'. Below, is the part of the program that aided in the choosing of random training sets from the lexicon:

```
while max(LearnTimes)>0

  if max(LearnTimesII)==0
    if max(LearnTimes)>=dom
      LearnTimesII(1:2:maxLines)=dom;
    else
      LearnTimesII(1:2:maxLines)=max(LearnTimes);
    end
    if min(LearnTimes)>=sub
      LearnTimesII(2:2:maxLines)=sub;
    else
      LearnTimesII(2:2:maxLines)=min(LearnTimes);
    end
    LearnTimes=LearnTimes-LearnTimesII;
  end
  LearnTimesIII(1:2:maxLines)=(2*rand(1,maxLines/2));
  LearnTimesIII(2:2:maxLines)=(rand(1,maxLines/2));
  LearnTimesIII(LearnTimesIII>1)=2;
  LearnTimesIII(LearnTimesIII>0.5&LearnTimesIII<1)=1;
  LearnTimesIII(LearnTimesIII<=0.5)=0;
  LearnTimesII=LearnTimesII-LearnTimesIII;
  LearnTimesIII(LearnTimesII<0)=0;
  LearnTimesII(LearnTimesII<0)=0;

  while max(LearnTimesIII)~=0
    lineInVec=ceil(rand*maxLines);
    if (lineInVec==0), continue, end

    if LearnTimesIII(lineInVec)>0
      LearnTimesIII(lineInVec)=LearnTimesIII(lineInVec)-1;
      outputVector(counter,1)=lineInVec;
      counter=counter+1;
    end
  end
end
```

Figure 40: This table shows how the random homogenous choosing took place. It has two loops. One loop is for the entire process, which stopped when it finished composing the list that needed to be taught. The second loop is the general loop. This loop created the small tables that made the boundaries of the range.

10.7 Adding Noise to the Learning Phase and the Activation of the Network

During the learning phase and the activation phase, it was found that adding small random values as noise to the input of the network during both processes made it easier on the network to generalize the learning process or for faster saturation. When small random values were added to the collaboration model, where the RH shared information with the LH, the added noise accompanied by the exchange of information made the processes more accurate and resilient to information released by the RH to the LH. The noise was small random values relative to the value of η, because it was supposed to be smaller than η. The ratio to 10 of η.

10.8 Differences between Kawamoto's Network and the Hopfield Network and the New Model

In this section, a comparison made by R.Ferber, in his paper, A Remark[8] on Alan H. Kawamoto. This paper helped in the understanding of some of the difficulties encountered when recreating the Kawamoto Model. Furthermore, it helped in the understanding of the weaknesses of the Kawamoto Model. After understanding this, it was clearer as to how the model could be improved to teach the Hebrew lexicon.

The network defined by Kawamoto differs from the Hopfield Network (Hopfield, 1982) in several ways. These differences lead to different dynamic behaviors:

1. In the Hopfield Network, there are only two possible activity values, 0 or 1.
2. Hopfield used a threshold function to calculate the new activation value of a unit.
3. In the Hopfield Network, only one unit is updated per time step. This is known as sequential updating.
4. In the Hopfield Network, the strength of learning is 1 and not a small value like 0.0003.
5. In the Hopfield Network, the vectors that need to be taught do not need to be far from each other. They can even be in one hamming distance.

These changes lead to fundamental differences in the models, for example, Hopfield Network is proved to converge on any input, when Kawamoto in rare cases can reach infinite number of iteration without convergence. Kawamoto Model can learn frequency features while Hopfield network cannot.

In these remarks on the Kawamoto Model, the author explains that it is possible to change the Kawamoto Model to the Hopfield Network. However, he also explains that some of Kawamoto’s work was done using very specifics parameters. For example, it is possible to change the Kawamoto activation function using the limit bounds (see Equation 2) to a Hopfield Network like the threshold function:

Equation 4: Hopfield Threshold Function

$$a_i\left(t+1\right)=\begin{cases}1 & \text{if } \sum_{k=1}^{N} W_{ik}a_k\left(t\right)>0\\ 0 & \text{otherwise}\end{cases}$$

The updating of the units will come to non-increasing, finally constant sequence. The existence of activity values different from 0 and 1 in the Kawamoto Network allows the increasing energy sequences. Another way is to choose the learning parameter η appropriately, or to construct the training patterns representing the words appropriately.

In the new model, presented in this paper, both options were used, finding the right value of η to a proper learning and choosing an entry from the learning set using the homogeneous random function.

[8] R.Ferber (1994): A Remark on Alan H. Kawamoto: Nonlinear Dynamics in the Resolution of Lexical Ambiguity: A Parallel Distributed Processing Account, unpublished. On the internet on the author’s home page: http://information-retrieval.de/ferber/homepage/english-html/publist.html

Other differences between the Hopfield Network and the network used by Kawamoto are Local Minima, and Stable States. In the Hopfield Network, a pattern of activities may have many possible successors in the development of the system due to the fact, that the next unit for processing is selected at random. All these possible successors differ at most in the activity of one unit since the activity of only one unit is changed. Assuming the Hamming distance exists in the space of activity patterns, (i.e. the distance between two patterns is the number of units in which the activities are different) all successors of an activity pattern lie in the immediate neighborhood since they have at most distance 1. If all patterns with distance 1 from *'a'* pattern a have a higher energy, this pattern *'a'* is a local minimum of the energy. In this case, it is a stable state because the next state would have to be identical or have distance 1, which it cannot because the energy cannot increase. The Hamming distance of the two patterns is maximal (The activity of both units' changes).

In the case of the Kawamoto Network, things are quite different. On the one hand, a pattern has a unique successor since there is no random selection of a unit, but all units are updated parallel to each other. This means that many units can change their activity. The Hamming distance to a successor is not bounded by 1 but the system can ‘jump around” over large distances.

Kawamoto used parallel updating. If only one unit (chosen by chance) would be updated at a time, the system would either converge to the activity pattern *'a'* (if unit 2 were chosen first) or converge to the activity pattern *'b'* (if unit 1 were chosen first). Both patterns are stable states of the system.

In conclusion, this paper demonstrates some problems in the Kawamoto Model that needed to be accounted for during this research when building the two models created from the Kawamoto Model. The goal is to measure the delicate differences in the information exchange and processing of close patterns like Homographs.

11 Appendix C. User Manual for the Network Simulation Program

11.1 *Introduction*

The program source code is implemented in Matlab version R2007b and runs on the Linux Machine with an AMD 64-bit dual processor's at 2.4-GHz with 16-Gigabytes of internal memory.

11.2 *File List and Description*

The files below need to be in the same tree directory as shown below. Each file has a small description that elaborates on its purpose and general parameters. More details on the main files will be elaborated on in the section following the table.

<table>
<tr><td>| ActivationStapByStap.m</td><td>Activation of the network one iteration. One cycle</td></tr>
<tr><td>| activeNeuronNetSilent.m</td><td>Run the network on one entry from the test set</td></tr>
<tr><td>| avg.m</td><td>Return average on vectors inside matrix</td></tr>
<tr><td>| avg2.m</td><td>Return average on vectors inside matrix without calculating zeros</td></tr>
<tr><td>| Begin_network.m</td><td>Main procedure of creating, and teaching the network.</td></tr>
<tr><td>| computesWords.m</td><td>Count how many words are homophonic, hetrophonic, or normal words</td></tr>
<tr><td>| convert.m</td><td>Convert the lexicon from ASCII characters to there equivalent vectors that created by ' creatRandomLetters.m'</td></tr>
<tr><td>| convertInputTable.m</td><td>Read the lexicon file to the memory.</td></tr>
<tr><td>| convetTable.m</td><td>Replace the Hebrew characters with English or unique characters.</td></tr>
<tr><td>| createaxes.m</td><td>Create bar figure to the iteration time</td></tr>
<tr><td>| CreateLearingTable.m</td><td>Creating of the training set sequence, using entries from the lexicon in homogenous random order.</td></tr>
<tr><td>| creatRandomLetters.m</td><td>This program is initiated in the creation of the network to create the sequence vectors to represent every character.</td></tr>
<tr><td>| creatFefureActivationGraph.m</td><td>Program that creates a figure of the activities of the homophone or heterophone with the appropriate color and legend</td></tr>
<tr><td>| creatFefureActivationGraphAll.m</td><td>Program that creates a figure of the activities of the hemisphere with the appropriate color and legend</td></tr>
<tr><td>| creatFefureActivationGraphAllLHRH.m</td><td>Program that creates a figure of the activities of both hemispheres with the appropriate color and legend</td></tr>
</table>

| EngtoHeb.m	Convert characters from English to Hebrew
| filemining.m	Independent program, for summering numerous results to one big report
| inifiles.m	Read the parameters from the ini file
| insainaty.m	Independent program, for insanity checks. To check similarities between all vectors.
| insainatyForLetters.m	Independent program, for insanity checks. To check similarities between all characters.
| lim.m	Limit function to limit x to be between +1 to -1
| loadDataFile.m	Load the Lexicon from file to memory
| loadIniFile.m	Load all parameters from file 'test.ini'
| main.m	The Main Program
| matchActivation.m	Find similarities(humming distance) on activities of one subject in LH or RH through all the tests
| matchActivationAll.m	Find similarities(humming distance) on activities of all subjects in LH or RH through all the tests
| matchActivationAllDotProdact.m	Find similarities(dot product distance) on activities of all subjects in LH or RH through all the tests
| matchActivationAllDotProdactRHLH.m	Find similarities(dot product distance) on activities between LH and RH of all the subjects
| matchActivationDotProdact.m	Find similarities(dot product distance) on activities of one subject in LH or RH through all the tests
| matches.m	Make a comparison between the vectors from the current state of the network and the desire vector, return the mistakes and non-conversion.
| netAnalasys.m	The main procedure to start the analysis on the log of activation of the network
| PreRuning.m	During the learning process this program test the currant state of the network if it learn enough, it return the error of the network if have.
| recognizeSilent.m	Convert bit vector to a string from the lexicon.
| PutAndGetFromBuffer.m	Program that help to the Corpus Callosum to exchange information between hemispheres
| reshpeVector.m	Create a bigger or smaller vector according to reshape algorithm in graphics
| SaveAllActivity.m	Save all active variables to file, before cleaning the memory
| saveReport.m	Save report on another way of statistics on the iteration time course.

| stdv.m	Return the standard deviation on matrix on each vector from a matrix
| tempNetParametrs.mat	Temporarily file that create a program during running
| Testing_Network.m	The main procedure that start the all testing phase after finish the learning phase
| trainNet.m	Train the net with the training set
|	
+---data	All the Parameters for the networks
| inputRH-15-02-07.txt	Backup RH Hebrew Lexicon
| convet hebrew.txt	Table of Hebrew and English characters to conversion
| inputLH.txt	Active LH testing Lexicon. Contain characters that spread in homogeneous sequence
| test.ini	All Parameters of the network
| inputRH.txt	Active RH testing Lexicon. Contain characters that spread in homogenizes sequence
| inputLH-15-02-07.txt	Backup LH Hebrew Lexicon
| NetworkRandomTable.mat	Temporary file that the program creates
|	
\---results	Here all the program outputs of the experiment are saved

11.2.1 main.m

This is the main program that runs the entire simulation. This is the starting point of the entire simulation. From here the network stared to load the parameters learning, testing, and in the end analyzing the results.

11.2.2 data/test.ini

This is an external file, containing all of the global parameters to the program. It defines the number of iterations of all tests and the constant values throughout the program. It also defines how many clues would be given in the tests.

11.2.3 data/inputRH.txt and inputLH.txt

The networks learned this input table. 'inputRH.txt' is the input for the RH network, and 'inputLH.txt' is the input for the LH network. In these files, there is the lexicon that is shown in Table 1 and in Table 5.

11.2.4 Begin_network.m

This is the main subsection of the program where at the end of the sub program, the network and all of the simulated 'subjects' are ready to be tested.

11.2.5 Testing_Network.m

This is the main subsection of the main program that take all the trained networks from the previous subprogram and in the end, all of tests are finished and the results are in the loges files and ready analyze.

11.2.6 netAnalasys.m

Subsection of the main program that start analyzes all the logs files, prepare a summery and statistics reports, and prepare graphs of the activation level and comparison between hemispheres

11.3 Structure of the Program

Main.m
loadIniFile.m

Begin_network.m
loadDataFile.m
filemining.m
inifiles.m
computesWords.m
convertInputTable.m
creatRandomLetters.m
CreateLearingTable.m
EngtoHeb.m
trainNet.m
PreRuning.m

Data
inputRH-15-02-07.txt
convet hebrew.txt
inputLH.txt
test.ini
inputRH.txt
inputLH-15-02-07.txt
NetworkRandomTable.mat

Testing_Network.m
choseTestVectorForTests.m
activeNeuronNetSilent.m
recognizeSilent.m
ActivationStapByStap.m
lim.m
comper.m
SaveAllActivity.m
PutAndGetFromBuffer.m

netAnalasys.m
anova.m
avg.m
matches.m
creatFefureActivationGraph*.m
matchActivation*.m
stdv.m
saveReport.m

בחירת משמעות בקריאה דמומה של מילה רב משמעית ברשת נוירונים: הבדלים ושיתוף פעולה בין המודל הימני והמודל השמאלי של המוח

חננאל חזן

תקציר

מיומנות הקריאה הדמומה נרכשת בגיל צעיר, במהלך קריאה דמומה אנו משלבים מגוון רחב של מקורות מידע בכדי לפענח את הסימנים החזותיים שאנו רואים (לדוגמה : תכונות הגיית המילה וצלילי ההברות, רמת השימוש והמוכרות שלנו מהמשמעות המילה או רמזים שנקראו בעבר). למרות מספר המחקרים שנעשו בתחום בשנים האחרונות, תהליך קריאה הדמומה וצורת פענוח והעברה המידע הנלווה מהצורה החזותית של המילה עדיין לא מובן דיו. מחקרים מראים ששני חלקי המוח האנושי משתתפים במאמץ הפענוח, וכל אחד מהם עושה זאת בצורה שונה לדוגמה, חלק השמאלי של המוח מהיר יותר בעיבוד משמעות המילה ומדייק יותר בבחירתו במשמעות המילה מהחלק הימני. החלק השמאלי מושפע מאוד מהסימנים הפונולוגיים של המילה הכתובה (ההיגוי של ההברות) , לעומת זאת החלק הימני של המוח רגיש יותר לצורה החזותית של המילה. בנוסף, הצד השמאלי של המוח בוחר במהירות את המשמעות הדומיננטית של המילה לעומת החלק הימני שמשאיר את המשמעיות הן הדומיננטית והן המשנית במצב של כמעט בחירה לתקופה ארוכה יותר מאשר בשמאלי. מטרת המחקר היא להציע מודל מתמטי של רשת נוירונים, על פי המחקרים שנעשו בחצויי מוח ובתכונות האופניות לכל אחד מחציי המוח ובדיקתו. המודל היה בנוי משני תתי מודלים, מודל הרשת השמאלית היה מורכב בצורה המופשטת ביותר והמודל של הרשת הימנית היה העתק של המודל השמאלי עם שינוי קטן ככול האפשר שיסביר את השינויים והתפקודים השונים של שני חלקי המוח כפי שמדווח בספרות. על פי המחקרים הסקנו שהחלקים הקשורים לפענוח הסימנים החזותיים בחלק השמאלי של המוח מחוברים גם לחלקים האחראיים על פענוח הגוי של המילה וגם לחלקים שאחראיים להבנת המשמעות של המילה. לדוגמה, למרות שפענוח התכונות החזותיות הנוצריות מהמילה הכתובה מיחסות יותר למשמעות המילה, חישוב ופענוח הפונולוגי של הסימנים החזותיים (המילה הכתובה) מהירים יותר מהחישוב ופענוח של המשמעות של אותה מילה. כתוצאה מזה, המשמעות הנגישה בחצי מוח השמאלי היא בראש ובראשונה משמעות הראשית שנוצרת מהפונולוגיה. לעומת זאת בחלק הימני של המוח, שבו החלקים האחראים לפענוח הסימנים הפונולוגיים אינם קשורים באופן ישיר לחלקים האחראים על פענוח החזותי של המילה הכתובה אבל קשורים בעקיפין דרך האזורים האחראיים על הבנת המשמעות של המילה הכתובה. דוגמה להבדל בעברת המידע בין היחידות השונות ובביצועי המבנה המתואר לעיל, השפעה על סדר על אירועים האחראים על הפענוח והעברת המידע בשני תתי המודל ובמודל עצמו יהיו שונים. כאשר מופיע גירוי חזותי לשני המודלים , המודל הימני יפענח את משמעות הגירוי החזותי לפני פענוח משמעות הפונולוגית, לעומת המודל השמאלי על אותו גירוי שבו סדר העברת המידע יהיה שונה ולכן גם סדר פענוח יהיה שונה.

בחירת משמעות בקריאה דמומה של מילה רב משמעית ברשת נוירונים: הבדלים ושיתוף פעולה בין המודל הימני והמודל השמאלי של המוח

מאת: חננאל חזן

בהנחיית: דוקטור לארי מנביץ.

דוקטור אורנה פלג.

עבודת גמר מחקרית (תיזה) המוגשת כמילוי חלק מהדרישות לקבלת התואר "מוסמך האוניברסיטה"

אוניברסיטת חיפה

הפקולטה למדעי החברה

החוג למדעי המחשב

תשרי תשס"ח

בחירת משמעות בקריאה דמומה של מילה רב משמעית ברשת נוירונים: הבדלים ושיתוף פעולה בין המודל הימני והמודל השמאלי של המוח

חננאל חזן

עבודת גמר מחקרית (תיזה) המוגשת כמילוי חלק מהדרישות
לקבלת התואר "מוסמך האוניברסיטה"

אוניברסיטת חיפה
הפקולטה למדעי החברה
החוג למדעי המחשב

אלול תשס"ח