

2013

# Computationally Modeling an Incremental Learning Account of Semantic Interference through Phonological Influence

Konstantinos Hatalis  
*Lehigh University*

Follow this and additional works at: <http://preserve.lehigh.edu/etd>



Part of the [Computer Sciences Commons](#)

---

## Recommended Citation

Hatalis, Konstantinos, "Computationally Modeling an Incremental Learning Account of Semantic Interference through Phonological Influence" (2013). *Theses and Dissertations*. Paper 1503.

This Thesis is brought to you for free and open access by Lehigh Preserve. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Lehigh Preserve. For more information, please contact [preserve@lehigh.edu](mailto:preserve@lehigh.edu).

COMPUTATIONALLY MODELING AN INCREMENTAL  
LEARNING ACCOUNT OF SEMANTIC INTERFERENCE  
THROUGH PHONOLOGICAL INFLUENCE

BY

KONSTANTINOS M. HATALIS

A Thesis

Presented to the Graduate and Research Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Computer Science

Lehigh University

September 2013

Copyright © 2013 Konstantinos Hatalis

All Rights Reserved

This thesis is accepted and approved in partial fulfillment of the requirements for the Master of Science.

---

Date

---

Thesis Advisor

---

Chairperson of Department

## **Acknowledgments**

I would like to express my gratitude to my adviser Prof. Hector Munoz-Avila for the useful comments, remarks, and engagement throughout the learning process of this master thesis. Furthermore I would like to thank Prof. Padraig O'Seaghda for introducing me to the topic of computational cognitive neuroscience, his comments, as well for his support. Also, I like to thank Prof. Almut Hupbach, Prof. Dominic Packer, and Kimberly Preusse who have willingly shared their time to join in meetings with me and provide additional pieces of advice for this project. I finally would like to thank my family, who has supported me throughout this entire process. I am eternally grateful for your love. This research was supported in part by a Lehigh CORE Grant.

# Table of Contents

Table of Figures .....	vii
Abstract .....	1
Chapter 1 – Introduction .....	2
1.1 Computer Models.....	2
1.2 Computer Models of Mind Processes .....	4
1.2.1 Picture Naming Paradigms .....	5
1.2.2 Properties of Semantic Interference.....	8
1.3 Artificial Neural Networks .....	10
1.4 Summary of Results .....	13
1.5 Outline of the Thesis.....	13
Chapter 2 – The Oppenheim Computer Model.....	15
2.1 General Description .....	15
2.2 Input Data.....	16
2.3 Propagation Schema.....	19
2.3.1 Lexical Activation.....	21
2.3.2 Lexical Selection.....	21
2.3.3 Learning .....	24
Chapter 3 – Phonological Model and Motivation.....	25
3.1 – Word Test Sets.....	26
3.2 – Human Experiments.....	29
3.3 – Phonological Model .....	32
Chapter 4 – Implementation Details .....	39
4.1 Simulation Test Sets .....	40
4.2 Code Structure .....	42
Chapter 5 – Simulations and Discussion .....	44
5.1 Simulation 1 – Using 2 Strongest Competitors.....	45
5.1.1 Simulation 1 Method.....	46
5.1.2 Simulation 1 Results .....	47
5.1.3 Simulation 1 Discussion .....	50
5.2 Simulation 2 – Using 5 Strongest Competitors.....	50
5.2.1 Simulation 2 Method.....	51

5.2.2 Simulation 2 Results .....	51
5.2.3 Simulation 2 Discussion .....	53
5.3 Simulation 3 – Adjusted Learning .....	54
5.3.1 Simulation 3 Method.....	55
5.3.2 Simulation 3 Results .....	55
5.3.3 Simulation 3 Discussion .....	57
Chapter 6 – Conclusion.....	58
6.1 Future Work .....	61
6.2 Final Remarks .....	62
References.....	63
Appendix.....	66
Simulation 1 Test Plots .....	66
Test 1.2.....	66
Test 1.3.....	68
Test 1.4.....	70
Simulation 2 Test Plots .....	72
Test 2.2.....	72
Test 2.3.....	74
Test 4.4.....	76
Vita.....	78

# Table of Figures

Figure 1. Two-layer feedforward neural network example. ....	11
Figure 2. Training and testing data for the Oppenheim model. ....	18
Figure 3. Sample demonstration on a subset of the Oppenheim model This figure is a slight modification from (Oppenheim et al., 2010). ....	20
Figure 4. Visual representation of the competitive boosting process used in (Oppenheim et al., 2010). ....	23
Figure 5. Homogeneous semantic and phonological sets with heterogeneous sets are presented in visual form used in human trails and in model simulations. Taken from Preusse (2013). ....	27
Figure 6. Homogeneous semantic and homogeneous phonological sets are presented in visual form used in human trails and in model simulations. Taken from Preusse (2013). ....	28
Figure 7. Mean response times for each cycle in human testing for the both condition. Taken from Preusse (2013). ....	29
Figure 8. Mean response times for each cycle in human testing for the semantic condition. Taken from Preusse (2013). ....	30
Figure 9. Mean picture naming latencies for each test condition. Taken from Preusse (2013). ....	31
Figure 10. Lexical activation levels with phonological facilitation to the target words and to the 2 strongest competitors. ....	33
Figure 11. Lexical activation levels with phonological facilitation to the target words and to the 5 strongest competitors. ....	34
Figure 12. Phonological abstract model with example. ....	37
Figure 13. Semantic condition simulation test sets. (a) homogeneous sets, (b) heterogeneous sets. ....	41
Figure 14. Both condition simulation test sets. (a) homogeneous sets, (b) heterogeneous sets. ....	42
Figure 15. Total mean selection times for each three conditions. ....	48



Figure 16. Semantic condition simulation test 1.1 (a) selection time per word, (b) mean selection time per cycle. ....	48
Figure 17. Both condition simulation test 1.1 (a) selection time per word, (b) mean selection time per cycle. ....	49
Figure 18. Phonological condition simulation test 1.1 (a) selection time per word, (b) mean selection time per cycle. ....	49
Figure 19. Total mean selection times for each three conditions. ....	51
Figure 20. Semantic condition simulation test 1. (a) selection time per word, (b) mean selection time per cycle. ....	52
Figure 21. Both condition simulation test 1. (a) selection time per word, (b) mean selection time per cycle. ....	52
Figure 22. Phonological condition simulation test 1. (a) selection time per word, (b) mean selection time per cycle. ....	53
Figure 23. Semantic condition simulation test 1. (a) selection time per word, (b) mean selection time per cycle. ....	55
Figure 24. Both condition simulation test 1. (a) selection time per word, (b) mean selection time per cycle. ....	56
Figure 25. Phonological condition simulation test 1. (a) selection time per word, (b) mean selection time per cycle. ....	56
Figure 26. Total mean selection times for each three conditions. ....	57

## **Abstract**

Computer models play a vital role in providing ways to effectively simulate complex systems and to test scientific theories and hypotheses. One major area of success for neural network models in particular has been in cognitive neuroscience for modeling semantic interference effects in memory. When a person sees a picture of an object such as a car multiple times, the memory of that object is primed so that it can be retrieved more effectively. When a picture of a similar object is seen, such as a truck, sharing semantic features with the primed object, then the primed memory of a car would interfere with the retrieval of a truck. This is known as semantic interference. A recent hypothesis by Preusse et al. (2013) puts forward that semantic interference is further increased by the sharing of phonemes among two words. In this thesis a new phonological computer model of lexical retrieval is developed based on this hypothesis using a two layer feedforward Artificial Neural Network (ANN). The new model can represent semantic interference effects through increased lexical activation by phonological features. Simulations were performed in a MATLAB environment each using a different variant of the phonological model. The simulations tested three conditions of activating semantic and phonological features. Results demonstrated that semantic interference is significantly increased when phonological features are activated alongside semantic features versus activating semantic features alone thus supporting the hypothesis by Preusse et al. (2013). The characteristics of the new ANN model could make it useful in studying other phenomena related to memory and learning.

# Chapter 1 – Introduction

## 1.1 Computer Models

Computer models have had a far reaching and profound effect in areas from meteorological and environmental research in modeling weather (Masters, 2011) and climate change (Colin et al., 1993) all the way to avionics in modeling wind resistance for aircraft (Mukherjee et al., 2000) and to VLSI circuit design in modeling the way that transistors behave at the nanometer level (Welch et al., 1990). Such models have been a critical factor in the way that scientific and engineering discoveries have been made in the twentieth century and are the defining hallmark of progress to be made further in the twenty first century. A computational or computer model is a high level theoretic system employing the use of mathematical, statistical, algorithms running on a computer or network in modeling phenomena's from scientific, engineering, and social disciplines (Hartmann, 2009).

Computer models themselves are at times intricate and may have little relation to the way the actual underlying system works - which brings to question of how can scientists trust them? The answer depends on how well a model can predict or replicate results of real systems. A computer model should be built on constraints of the data from the domain that it is from. This is necessary so that the model can be appropriately tested empirically and showcase results in a clear and concise manner even if they are wrong. It is also not necessary that be as complex as the underlying phenomenon.

Of particular interest for this thesis are models related to the operation of the human brain. If we could develop a model as complex as the human brain for instance, built using trillions of lines of code or thousands to millions of algorithms, then the model itself may not be graspable and could be just as confusing as the brain. It would be little to no use in aiding our understanding of the mind. So at often times, such high levels of complexity are not required and the principle of Ockham's razor is followed – keeping a model as simple as possible while incorporating as much data as possible (Myung & Pitt, 1997; Young, Peter, Parkinson, & Lees, 1996; Blumer, Anselm, 1987; Domingos, 1999; Burton & Obel, 1995).

One major area in a dire need of such models is in cognitive neuroscience. Cognitive neuroscience is a scientific field which studies more closely neural encodings of mental processes through overlapping theories from both psychology and neuroscience. The study of different parts of the brain is riddled with difficulty making it a challenging task in understanding when done by an unaided human. But this area is greatly enhanced through the use of high performance computing and more importantly of computer models such as artificial neural networks. Their use as a vehicle to enrich our understanding of different aspects of the mind has enabled us to study the brain in an easier, reliable, effective, and safe manner.

## 1.2 Computer Models of Mind Processes

This thesis explores the use of artificial neural networks to model processes of the mind which affect how we think and retrieve learned words from memory. When we see a picture of dog, for instance, we recognize that it is a dog, but some time before we do our minds are racing through our memory banks to try to locate the word dog. And while we are searching for the term we come across other words we have learned in our memory which may share similar semantic features with a dog such as a cat which shares feature similarities such as the fact that a cat is also a mammal, that it is a pet, has four legs, and has fur. This semantic similarity with a cat may cause our memory retrieval process to slow down before picking the word dog.

This phenomenon, arising from when similar concepts are retrieved from memory, is known as semantic interference (Oppenheim et al., 2007), also known as semantic blocking, and is a version of retrieval induced forgetting during speech production. It has been argued (Oppenheim et al., 2010) that this is a side effect or “dark side” of another phenomenon known as repetition priming where if a person was to see the picture of a dog, and then sometime later see it again they would be able to name it faster this time around. This is because the person may have seen the picture of the dog before and has primed it in his memory. This decreases the chance of misnaming the picture (reduces naming error) while increasing the speed of naming the picture. This retrieval priming process can have a negative effect when trying to retrieve words that share a

similar theme or semantic category, such as a car and a truck which are both road vehicles.

Similar to the cat and dog example, seeing a picture of a car twice for instance primes it for future retrieval. Then when seeing a related picture such as a truck, the memory retrieval process has to search through memories of a car first because it is primed in the same category as truck. So when thinking about what kind of vehicle the object in the photo is, the mind has to pick between choosing a more learned word such as car or the word truck. This negative effect of priming a word in memory as it is incrementally learned can distract or interfere with the lexical retrieval of a target word – this is what is known as semantic interference.

### **1.2.1 Picture Naming Paradigms**

The task of looking at a photo and attempting to semantically name the object in it is known as the picture naming task and is a critical component in cognitive psychology experiments in studies of semantic blocking because it ensures that responses are retrieved from memory (Schnur et al., 2009). Semantic blocking can be tested in many picture naming domains including object taxonomies, human actions, and facial identification. More specifically to this Thesis, we conduct simulations under a picture naming task that is known as the blocked-cyclic naming paradigm (Damian et al., 2001). In a psychology experiment with human subjects, a person can be given to name a small set of pictures, such as 3-6 pictures for instance. That set is then presented to the subject again, each picture presented one at a time with a small pause in between, but the

order of the pictures in the set may be randomized under the control of a computer program. When a picture is presented a person then says out loud the name of the picture and their response time is recorded. This is done to identify voice onset latencies.

A subject may cycle through the repeated set a small finite number of passes such as 4 to 7 times. All these cycles are considered a block. The set of photos in each block may consist of pictures with designated names that are either taxonomically related or unrelated. If they are related then this is known as a homogeneous condition with each picture in the set belonging to the same semantic category, such as a set of pictures of flowers. A set in another block may also consist of pictures which are not related by a semantic category. This is known as the mixed or heterogeneous condition, and every picture represents its own unique semantic group.

Another picture naming task is the continuous paradigm (Howard et al., 2006), but is not directly studied in this Thesis. With this paradigm pictures are continuously presented to a subject to name them but the pictures do not repeat and they do not cycle. Instead a series of pictures may be presented that could share a semantic category. This paradigm is used in other cumulative semantic interference studies because interference can be demonstrated through a continuous stream of semantically related words with each one taking a bit longer to name from the previous pictures.

Under the blocked-cyclic naming paradigm, response times of subjects are reported to be slower in the homogeneous naming condition (Belke et al., 2005). This effect also increases over time as more semantically similar pictures are presented to be named. This signifies that semantic interference may not be temporary but rather that it corresponds to sustained changes in memory. This makes it highly appropriate to investigate the properties of cumulative semantic interference through the use of complex computer models such as neural networks to simulate lexical access and retrieval in speech production.

Looking at a neural network as an abstract version of memory one could model the process of lexical retrieval (Oppenheim et al., 2010). For instance, from a high level perspective every time a picture is presented to be named input nodes are activated – these represent the semantic features of that object. The link from the corresponding knowledge to the memory of where the word is stored (in our case an output node in the neural network) is strengthened. This leads to adaptation of other mappings of unrelated words getting their links weakened. This model through this form of competition is fundamentally dynamic, with continuous adjustment of semantic concepts to word mappings. These properties make this form of a computer model a potent tool for the investigation of stable and dynamic meaning representations in semantic blocking since it is easy to implement various types of experiments with.



### **1.2.2 Properties of Semantic Interference**

Before diving in the exact details of how the model works and how simulations are setup (both of which are covered in later chapters), it is very important to first understand the fundamental principles that are required to induce cumulative semantic interference. Howard et al. (2006) and Oppenheim et al. (2010) present three principals or properties of lexical retrieval that when they interact with each other in a certain manner, produce interference in picture naming response time. The first is having a shared activation of a target word (word to be named) with competitor words (words with similar semantic features). In the network model, a shared activation means that multiple output word nodes are activated or chosen by the network activation algorithm. The second property is competitive selection which defines the competitive nature of semantic interference and is implemented computationally by a boosting process. The third property is known as priming which implements an incremental learning process which is the driving force behind inducing cumulative semantic interference.

Shared activation is a natural process arising from feature based semantic representations of a concept or word (McClelland et al., 2003). In other words, a homogeneous set will share common semantic features which would induce in the lexical retrieval process several lexical activations or words to be chosen in memory in anticipation for retrieval. For example, in a set of words such as: CAR, BUS, TRUCK, PLANE, TRAIN; all of them share the common semantic feature

that they are a vehicle and thus belong to that semantic category. When a subject is presented a picture of a TRUCK, the target word to be named, during the retrieval process the semantically related words such as CAR, BUS, PLANE, and TRAIN will be activated in memory along with the target word TRUCK. It should be noted that while they all, theoretically, are activated that does not imply that they are all the correct target word. The shared activation property, which could actually be seen as the first step in inducing cumulative semantic interference in the lexical retrieval process, only sets the stage for the competitive selection step by picking the “competitor words”.

The second step or property is competitive lexical selection (Howard et al., 2006). This is a process by which the target word is picked among all the activated competitor words. The key is to have competitor words with strong activation levels; this induces a competition process which increases the error rate of which the target word is chosen. The more non-target words are activated and the higher their activations, the harder it is to choose the target word. Oppenheim et al. (2010) provides the analogy that this process is similar to a sudden death race where athletic teams are competing to reach the end and where only one can win. Shared lexical activation can be viewed as several teams that are lined up based on the level of their activation.

In the vehicle set example, let's assume again that the target word is TRUCK and it is activated. If a competitor of it is activated such as BUS, this will slow down the selection process of choosing the word truck in memory. If BUS is

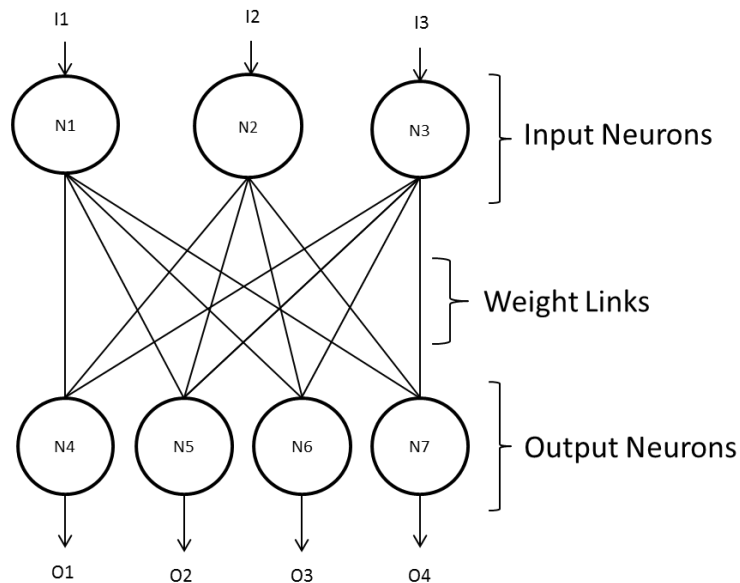
more often activated, i.e. has been seen often in the past, then the accuracy of choosing TRUCK would be decreased. Shared lexical activation, by activating related competitors, in combination with competitive lexical selection process slows the picking of the target word. This is what is meant by semantic interference – semantically similar words hindering the retrieval of a target word.

In order to define semantic interference as a cumulative process, a third property known as priming is required to carry the effects from the first two properties over to testing of other words. When a word is retrieved, it is primed in memory for future retrieval. A subject is learning that word and is able to recall it better the more often they see it. This makes the retrieval of a target word easier and faster and the retrieval of competitor words less likely. The effects of priming on interference are incrementally learned or accumulated during continuous experience driven mappings from semantics to words (Howard et al, 2006). Priming is also unaffected when a subject is presented with irrelevant or heterogeneous words (Damian et al., 2005). For instance, if the word CAR is primed, and then an unrelated picture of a flower is seen, the interference effects of CAR would be unchanged.

### **1.3 Artificial Neural Networks**

Artificial neural networks (ANNs) are a popular tool in pattern recognition and data mining. They also play a critical role in modeling semantic interference so in this section we dive into some of the details behind ANNs. Also known as perceptrons, their creation was inspired by the human brain and the way that

biological neurons are connected and interact with each other (Rosenblatt, 1958). Similar to a biological neurons sending activation signals to each other, an ANN also consists of neurons sending signals to each other. There is a wide variety of ANNs (Zhang, 2000) but in this Thesis we focus only on a type called a feed-forward neural network, which is a fairly common network type (Duda et al., 2012). We constructed Figure 1 to illustrate a two layer feed-forward neural network having input and output layers only. This is also known as a single-layer perceptron network and has a very simple architecture.



**Figure 1. Two-layer feedforward neural network example.**

In Figure 1 we illustrate an example with three input nodes and four output nodes. Input nodes are linked to output nodes by weight coefficients which represent the importance of the connection. In this architecture there are no links between input nodes or output nodes, and output nodes do not link back to input nodes. This is why this is called a feed forward network, because connections do

not form a cycle. An ANN is used to solve a classification problem by learning to linearly separate patterns. Inputs can be in the form of numbers or binary values. The sum  $x$  of the products of the weights and the input values is calculated from each input node. Sometimes a bias value is also added to the sum. To calculate an output, each output neuron has an activation function associated with it and the sum  $x$  is then entered into this function. Table 1 summarizes three popular activation functions (Bishop, 2006).

Name	Formula
Identity	$A(x) = x$
Step	$A(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$
Sigmoid	$A(x) = \frac{1}{1 + e^{-x}}$

**Table 1. Activation Functions (Bishop, 2006).**

Feed-forward ANNs are trained by a learning algorithm that adjusts weight values between neurons (Bishop, 2006). A very common learning algorithm is called the delta rule which is a gradient descent learning rule (Widrow et al., 1960) that calculates errors between output values of the network and desired output data, and uses this to update to the weights. Given a training set of inputs, with desired output values, a network will learn appropriate weight values so that it would correctly classify those inputs. One cycle through an entire training set is considered an epoch (Witten et al, 2005).

## **1.4 Summary of Results**

We implemented and tested a computational system for producing semantic interference effects based on semantic and phonological lexical features. Motivation for the system follows a theoretical model for phonological feedback and preparation on semantic interference in the blocked cyclic naming paradigm by Preusse et al (2013). This model extends the computer model of semantic interference of Oppenheim et al. (2010), from here referred to as just the Oppenheim model. Our new model incorporates new phonological features, changes to several methods in the Oppenheim model, and new test data. Under certain simulation conditions this model performs acceptably in replicating human test trails and is able to be used to further explain other facets of the phonological facilitation hypothesis. In our experiments we observed that semantic interference is significantly increased when phonological features are activated alongside semantic features versus activating semantic features alone. This confirms the Preusse et al (2013) hypothesis that phonology of words effects latency of response times.

## **1.5 Outline of the Thesis**

All three properties, shared activation, competitive selection, and priming, must be met in creating a computer model for cumulative semantic interference. Chapter 2 of this thesis reviews in detail the Oppenheim model. Chapter 3 explains in detail the model expansion to incorporate the introduction of phonological based facilitation for word production. Chapter 4 reviews the

implementation details of this model in Matlab as well as the input data. Chapter 5 showcases the various trail simulations conducted to evaluate the model and to test different conditions of the phonology facilitation hypothesis. Chapter six is concluding remarks and future work.

# Chapter 2 – The Oppenheim Computer Model

## 2.1 General Description

The model for cumulative semantic interference during speech production developed by Oppenheim et al. (2010), referred to as the Oppenheim model from here on out, features three robust methods for modeling shared lexical activation, competitive lexical selection, and priming through incremental learning. The model design was inspired by Howard et al. (2006) which implemented semantic activation in the form of semantic input nodes of a two layer neural network.

- **Shared Lexical Activation.** Shared Activation is a processes by activating lexical outputs that share a semantic feature with the activated target word.
- **Competitive Lexical Selection.** Competitive selection is modeled by having connections from every word to every other word as inhibitory where activating one word can inhibit other and competition arose from choosing a word whose activation surpassed a built in threshold.
- **Priming.** Priming is implemented by strengthening connections to the word chosen by the competitive process, which may or may not have been the target word. The connections of the word chosen are strengthened by increasing the value of the weights connecting the lexical (output) node of the network to the input semantic nodes that represent that chosen word.

Oppenheim enhanced the Howard model in three ways. First he updated the learning mechanism by which priming functioned so that priming was error



driven. This was done to show the effect of word errors on lexical access and to show the insensitivity of cumulative semantic interference to words that are unrelated over time. Oppenheim argued that this fashion of learning was closer to how many other cognitive theories explain learning. The second enhancement Oppenheim added was the utilization of a new competitive selection process known as boosting. This new method allows for lexical competition be able to be played out over time and better model response time of human subjects through lexical selection times. It was also argued that this new method better modeled competition in the brain. Lastly the Oppenheim model was created in a fashion so as to demonstrate that cumulative semantic interference could arise from error driven learning instead of competition.

## **2.2 Input Data**

The Oppenheim model demonstrated shared activation through a lexical activation method, competitive selection through lexical selection or boosting, and priming through the Widrow-Hoff learning rule customized for logistical activation. The whole model can be seen as a simple two layered feedforward neural network, as in Figure 1, with an added independent function to model competition. The network is first trained on a set of words. One run through the whole set is considered one training epoch. Training would consist of several epochs, each one going through the word set in a randomized order. The word set itself comprised of strings of binary values. One word is represented by a string of zeros with only two ones systematically placed somewhere in the string to

represent the semantic features of a word. For instance, the word cat can be represented by the following binary value:

10000100000000000000

The 1's in the first and 6<sup>th</sup> places are the two features that represent a cat, such as mammal and terrestrial. In the Oppenheim model a word is strictly represented by only two features and thus will have only two 1's in its input representation. These 1's are important because they are responsible for activating the two semantic feature nodes of the network when that word is fed into it. Design of the input array of words can be customized to suit any type of word but must adhere to certain conditions based on how tests or simulations are designed.

Figure 2 showcases an example set of 50 words used in the training and testing of several simulations in Oppenheim et al. (2010). Semantic features representing a word are highlighted in red. The first five words can be seen all sharing a feature in the first column but a different second feature. This represents that these five words all belong to homogeneous set or a group of words sharing a theme or category such as farm animals. The actual category of what the sets are, such as vehicles or animals, or what the actual words are is arbitrary in the construction of this and any kind of word set. We only use specific names of categories or words for demonstrating examples. What matters is that the design of the word sets conforms to representing separate homogeneous and heterogeneous groups.



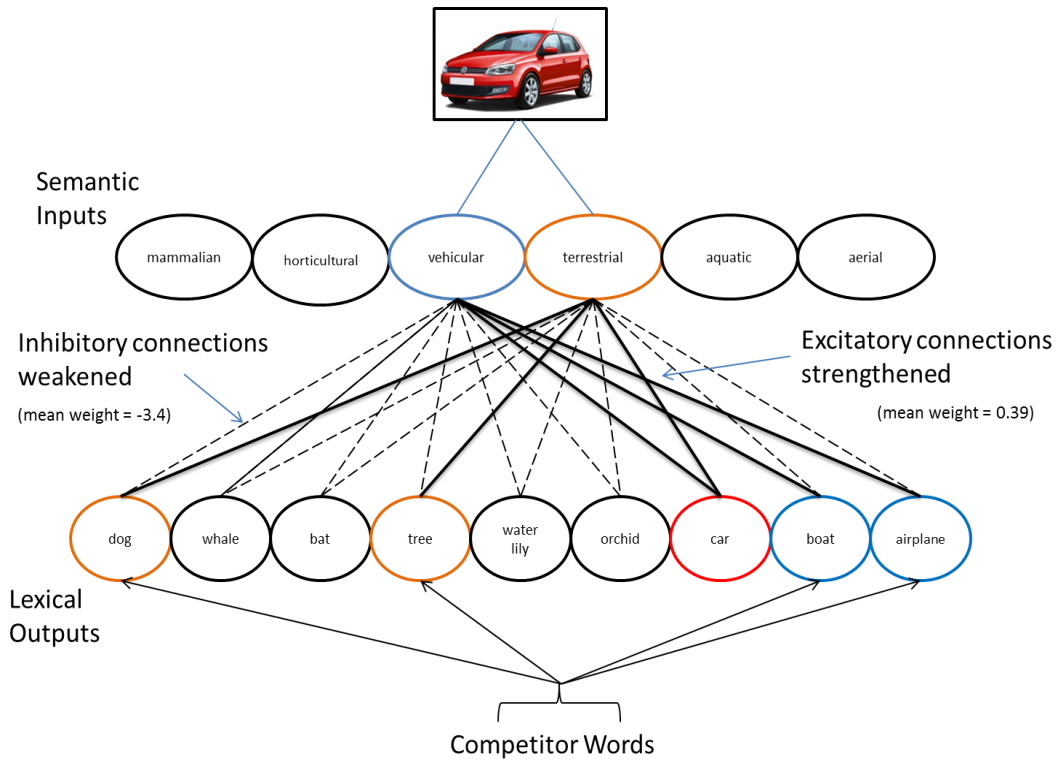
Figure 2. Training and testing data used in simulations in Oppenheim et al., (2010).

Training on a word set represents a person learning that word set. Initially all weights connecting all words are set to zero. In training with each pass through the set, each connection from a word to its features is strengthened representing a person becoming more familiar with that word. In training whole passes are conducted through the set, but in testing based on the conditions of a simulation only a subset of the words are used.

### **2.3 Propagation Schema**

As a word enters the model its two features activate the semantic nodes. Features that do not represent the word get a value of 0 and this do not become activated. Once this set of semantic features is activated, the output or lexical nodes are activated that correspond to those features. This is done through the shared activation method. Depending on the strength of the connection, multiple words may be activated. At this point the competitive selection or boosting mechanism starts and begins a process by amplifying the activations of the words. Words are continuously amplified until the strongest one passes a differential threshold. This then represents the selected target word. The number of times a word needs to be amplified to pass the threshold is analogous to response time. The word chosen may not actually be the target word and this represents semantic errors in the selection process. Lastly learning takes place which strengthens connections to the selected word and weakens connections to competitors in order to aid in or prime the selected word for future retrieval. It should be noted however that the competitive selection method does not affect learning. After the

boosting process, the amplified activations of the words are reduced back down to their pre-boosted levels.



**Figure 3. Sample demonstration on a subset of the Oppenheim model This figure is a modification from (Oppenheim et al., 2010).**

Figure 3 illustrates the basic functionality of the Oppenheim model using a small subset of the whole network. In a sample test run when picture of a CAR is presented to the network, which will actually be in the form of a binary value, the VEHICULAR and TERRESTRIAL semantic features are activated. Based on weight values from training, five lexical nodes will be activated. These are DOG, TREE, CAR, BOAT, and AIRPLANE. Aside from CAR, the other four are competitor words who share one feature with CAR. The activations of these words occurred because they have excitatory connections to one of the two

features of CAR. While all other words such as ORCHID or BAT have inhibitory connections (with a negative weight value) and thus have activation close to zero.

### 2.3.1 Lexical Activation

When semantic nodes become activated with an input the lexical activation method starts. The first step in choosing the values of the lexical activations is to sum the activations of each semantic node  $a_j$  multiplied by the weight from each node to each lexical node  $w_{ij}$ . The summation occurs in a net input:

$$net_i = \sum_j w_{ij} a_j \quad (\text{Eq. 1})$$

The letter  $i$  represents a lexical node while  $j$  represents a semantic node. After the net summation is calculated, it is inputted into a sigmoid function in order to calculate the lexical activations  $a_i$ :

$$a_i = \frac{1}{1 + e^{-(net_i + v)}} \quad (\text{Eq. 2})$$

A lexical activation will have a value between zero and one. In order to better simulate the brain, and how mental noise might distort lexical retrieval, a noise variable  $v$  is added to the net sum having a normally disturbed amount of noise with a mean of zero and a standard deviation of  $\theta$ .

### 2.3.2 Lexical Selection

The second step is lexical selection which induces a competitive process on the activated lexical node values in order to choose a single word. A booster mechanism is used which adds additional activations to the network, increasing the existing lexical activation values by a small amount until a word surpasses a

differential threshold. The booster process does not know which word is the target word and when it is complete and does not know if it has chosen the correct word. In this way it is an entirely naïve process. As the activations are increased, eventually one of them will surpass that of all the other activations. Competition is modeled in the boosting process by a differential threshold where a word is compared to the mean activations of all other words. Over time or several iterations a word will eventually pass the threshold. This can be represented as equation 3 below where in each time step  $t_n$  the lexical activation of node  $i$  is increased by a small factor  $\beta$  which is a constant value greater than one giving a new amplified activation  $a_{it_{n+1}}$ :

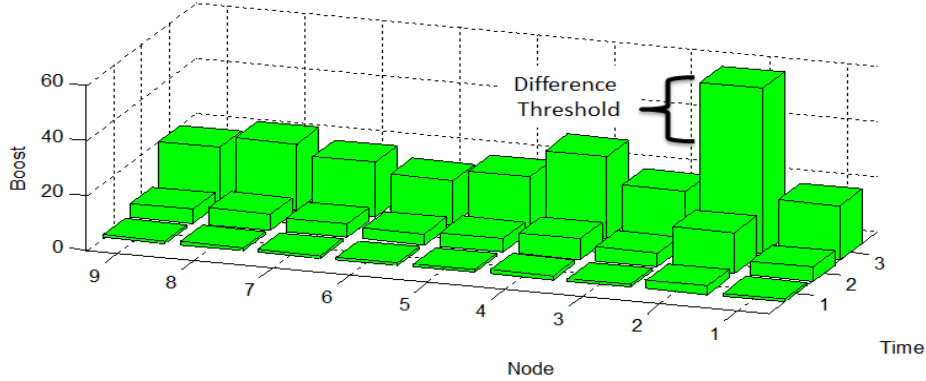
$$a_{it_{n+1}} = a_{it_n} \times \beta \quad (\text{Eq. 3})$$

A word is chosen when a boosted word activation surpasses the mean value of all other words  $a_{others\ t_n}$  of a constant threshold  $\tau$ :

$$\tau > (a_{it_n} - a_{others\ t_n}) \quad (\text{Eq. 4})$$

If an activation does not surpass the threshold, then the boosting process repeats until a winner is determined. To ensure that words are selected quickly, a word is omitted in the selection process if the iterations of boosting surpass a constant number of boosts  $\Omega$ . Oppenheim attributes this to a “wait and give up” theory. Oppenheim also admits that this boosting process lacks exact neural motivation but that with its discernible differential threshold it models semantic competition well enough for intended purposes. As a note, all activations return back down to

their pre boosting levels so that the learning process does not become affected by any biased activation.



**Figure 4. Visual representation of the competitive boosting process.**

In order to save time, and under the assumption that each repeated boost is not different (does not vary), a logarithmic function is used to represent the entire boosting process versus going through two loops to test each activation individually.  $t_{selection}$  represents the calculated selection time for the inputted activation  $a_{itn}$ :

$$t_{selection} = \log_{\beta} \left( \frac{\tau}{a_{itn} - a_{others\ tn}} \right) \quad (\text{Eq. 5})$$

Figure 4 represents a graphical interpretation of the boosting process. The x-axis represents the nodes in the output layer of the network, also known as the lexical layer, the y-axis represents the selection time or number of iterations of going through each boost, and the z-axis represents the boost level or activation level after amplification. Initially all activation levels would be substantially too small to notice any discernible differences. At each iteration each activation is



boosted by the factor  $\beta$  and eventually the difference by equation 4 would be passed. At that point the highest activation would be chosen as the selected or estimated target word. In the figure this would be word (node) two and this would have occurred in only three iterations signifying that the response time or selection time would be 3. Now Figure 4 shows a simplified case with a high beta to show how the difference threshold works.

### 2.3.3 Learning

Finally the learning process takes places which models priming by updating a target word to be more accessible or by making its competitors less accessible. The algorithm uses an error driven rule to update the weights of the connections from the lexical nodes to the semantic nodes. This rule is the Widrow-Hoff rule which has been adapted for the logistical activation function (Widrow et al., 1960).  $\eta$  represents a constant learning rate,  $a_i$  and  $a_j$  represents the lexical and semantic node activations,  $d_i$  is the desired lexical activation (the correct word activation for the semantic feature inputs), and  $w_{ij}$  is the weight change from the lexical to the semantic nodes:

$$\Delta w_{ij} = \eta(a_i(1 - a_i)(1 - d_i))a_j \quad (\text{Eq. 6})$$

This learning algorithm is error driven by the discrepancies between the desired activation  $d_i$  and the network lexical activation  $a_i$ . Utilizing this algorithm, connection weights of the target word to its semantic features are increased while weights of those same features to all other words are decreased.

## Chapter 3 – Phonological Model and Motivation

In this chapter we describe a high level view of our phonological model as well as provide motivation for its creation. We begin with describing the Preusse et al. (2013) hypothesis, the way they designed their tests, and results from human trails. We then dive in the details of the algorithms behind the phonological model as well as provide a few examples of how its processes work.

Oppenheim et al. (2010) provided a computer model to simulate cumulative semantic interference. This model provided a basis for explaining interference by activation levels of competing words. Semantic interference is also known as retrieval induced forgetting and is a product of when similar concepts are retrieved from memory (Belke et al., 2005). The significance of the Oppenheim model is that words which share meaning and form, when they are more activated will generate higher semantic interference. A hypothesis for interference is presented by Preusse et al. (2013) where retrieval induced forgetting can be increased by sound similarity or phonology of words.

The basic premise derives from the idea that form-based preparation may occur when attention to shared components pre-activates them. This results in facilitation in picture naming (Preusse et al., 2010). In other words, in the picture naming domain, when words are presented to a subject that have the same phonology such as PUFFIN and PIGEON which share the phonology /p/, will incur interference. A common phonology could further increase interference since





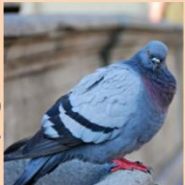


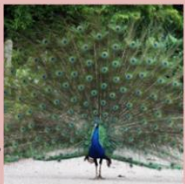

the hypothesis describes phonological onsets as feeding back to a word and increasing its activation.

Human experiments have been conducted in the blocked cyclic naming paradigm for phonological influence on interference. Since this theory is still in development, the purpose of this thesis was to explore the creation of a computer model of semantic interference based on phonology. The Oppenheim model was the perfect template for building up such a model since it already possessed robust methods for interference based on semantics in words. This chapter first describes the setup and experiments of the phonology hypothesis and discusses their significance. However an analysis of these experiments is beyond the purpose of this thesis as we only focus on computationally replicating human results in building a robust model to test alternative simulations for phonological interference. Then the phonological model and test sets used in simulation will be described followed by a high level view of the functionality of the model. Chapter 5 will describe the implementation details in Matlab and chapter 5 will describe the various iterations that took place to get a robust model as well as some basic simulations of the model showcasing phonological interference.

### **3.1 - Word Test Sets**

Human trials took place in the Language Production Lab at Lehigh University. Participants sat in front of a computer and were presented pictures of items one by one. As a picture popped up the participant would say the name of the picture and the computer would automatically record the time it took to say










the word and move on to the next picture. The software used for this was called E-Prime 2.0. All word sets were presented in random order and in each block there were seven naming cycles per set, with three words to a set with no repeats.

Homogeneous Semantic and Phonological Sets				
		Vegetables /k/	Birds /p/	Flowers /l/
Heterogeneous Controls	1	collards 	puffin 	lily 
	2	carrot 	pigeon 	laurel 
	3	cabbage 	peacock 	lilac 

**Figure 5. Homogeneous semantic and phonological sets with heterogeneous sets are presented in visual form used in human trails and in model simulations. Taken from Preusse et al. (2013).**

Four groups were designed, each containing three sets words and each set having three words. The four groups represented three homogeneous conditions and a control heterogeneous condition. All words picked for all sets were disyllabic nouns. Figure 5 represents the Heterogeneous and Homogeneous *both* sets; used in human testing. The heterogeneous group has three word sets; in each set no word shared a semantic or phonological feature. All three words are

independent. For instance set 1 has COLLARDS, PUFFIN, and LILY which have nothing in common. Figure 6 also shows the homogeneous both condition which are sets of words which share both a semantic feature and a phonological feature. For instance the first column shows the words COLLARDS, CARROT, CABBAGE. All three words belong to the same semantic category of vegetables and all three words have share the same phonology /k/.

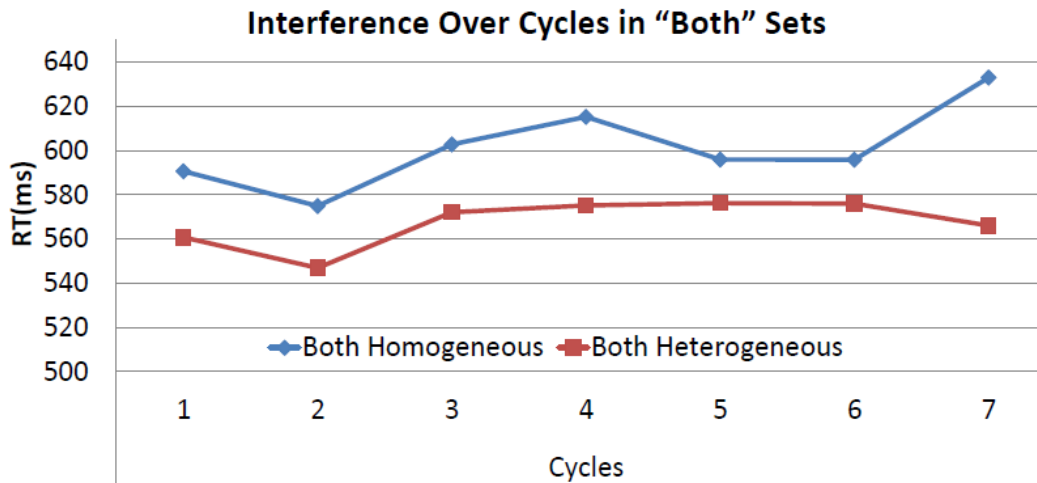
		Homogeneous Semantic Sets		
		Vegetables	Birds	Flowers
Homogeneous Phonological Sets	/p/	parsnip 	peacock 	poppy 
	/l/	lettuce 	lapwing 	lilac 
	/k/	cabbage 	condor 	crocus 

**Figure 6. Homogeneous semantic and homogeneous phonological sets are presented in visual form used in human trails and in model simulations. Taken from Preusse et al. (2013).**

Figure 6 shows two other word set groups, homogeneous semantic sets which shared only a semantic feature but not a phonological feature, and the homogeneous phonological sets where words in a set only shared a phonological

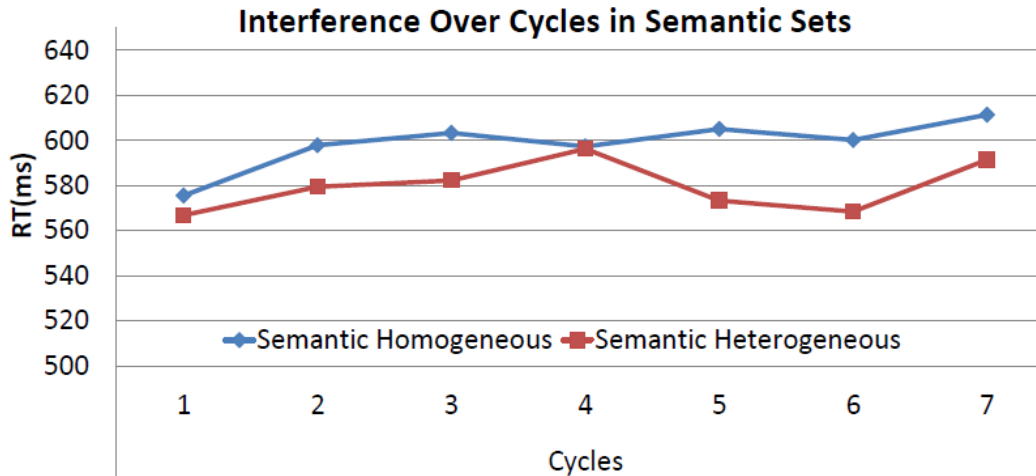
feature. These sets are essentially heterogeneous sets in the semantic sense where words had nothing in common except their phonology.

### 3.2 - Human Experiments



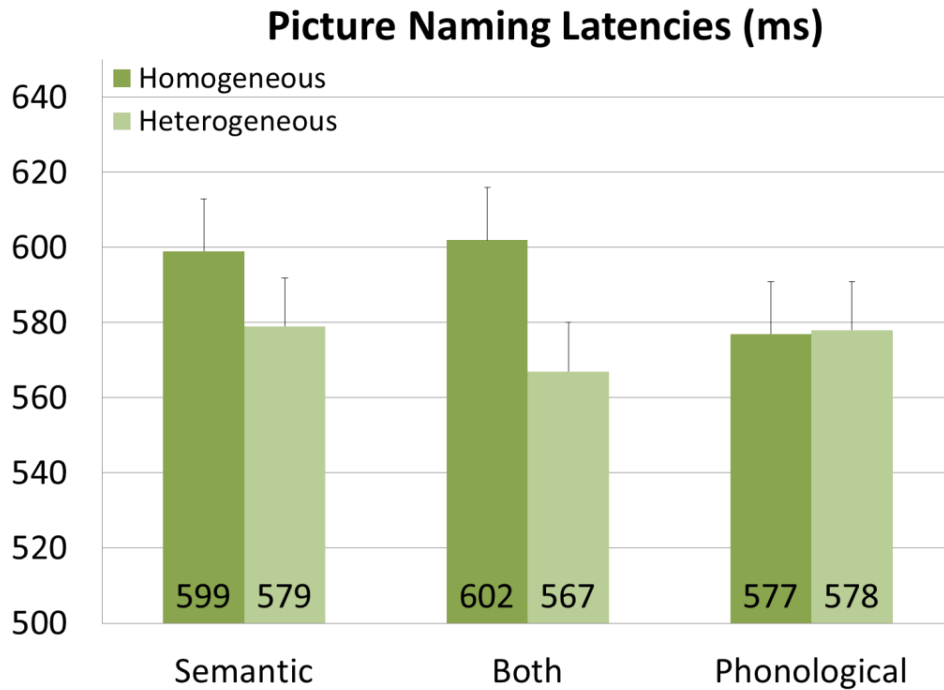
**Figure 7. Mean response times for each cycle in human testing for the both condition. Taken from Preusse et al. (2013).**

Three main conditions were tested under the blocked cyclic naming paradigm. In the “semantic condition”, participants were shown pictures from the homogeneous semantic sets and pictures of a heterogeneous control set. Response times for all participants were recorded and averaged over all trials. In the second and most significant “both condition”, participants were shown pictures from the homogeneous *both* sets and pictures of a matching heterogeneous control set and again had all their response times recorded and averaged. The third “phonological condition” and participants were shown pictures from the homogeneous phonological sets and then pictures of a heterogeneous control set.



**Figure 8. Mean response times for each cycle in human testing for the semantic condition. Taken from Preusse et al. (2013).**

Figure 7 shows the response time measured in milliseconds of how long it took participants to name pictures in the both condition. Response times for each cycle, the time it took to name each of the three words, were averaged. Across all seven cycles, interference is present as difference in response times between the homogeneous and heterogeneous word sets. When comparing results with Figure 8 we see mean response times in the semantic only condition, interference appears to be more consistent across the cycles in the both condition. In addition, in the semantic condition interference is weaker than in Figure 8. This signifies that phonological similarity of words when combined with semantic similarity between words increases naming latencies.



**Figure 9. Mean picture naming latencies for each test condition. Taken from Preusse et al. (2013).**

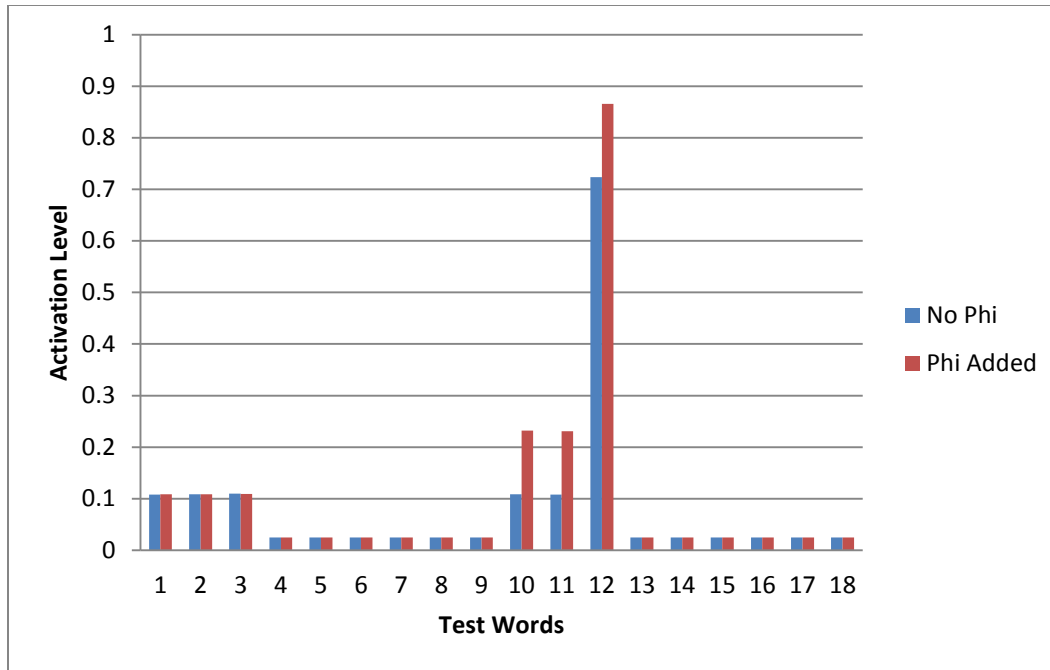
In Figure 9, picture naming latencies are averaged over all cycles to present total interference affects. Homogeneous sets for the *semantic* and *both* conditions show interference effects over heterogeneous sets. For the *both* condition we see a significant latency between homogeneous and heterogeneous sets. This interference in the *both* condition is about twice the size of the semantic condition. However, in the phonological condition, where the homogeneous sets only share a phonological feature and are otherwise equivalent to a heterogeneous set, we do not see any interference. More research is needed to explain this lack of phonological influence in non-semantic similar sets (Preusse et al., 2013).



### **3.3 – Phonological Model**

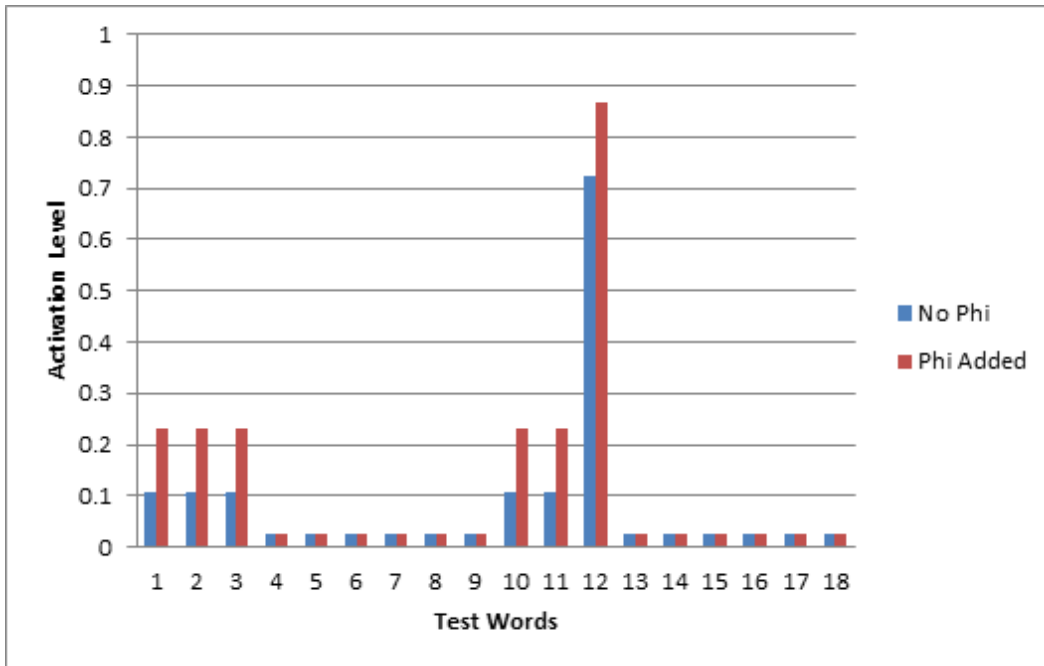
Experimental trails of the phonological facilitation hypothesis yielded significant results. In order to further validate the hypothesis a computer model would be needed. Oppenheim’s model provides a suitable template for adding additional methods to accommodate phonological similarity between word sets. Oppenheim’s model has already been shown to explain cumulative semantic interference through competition and incremental learning for semantically similar words using robust methods. This new phonological model was used to replicate human trails in the form of simulations. Chapter four outlines simulations done in replicating the semantic and both conditions to see how well this model would perform in predicting appropriate outcomes. However, simulations done with the phonological condition were not studied too deeply since initial trials with humans by Preusse et al. (2013) did not yield significant results that could be compared to.

The addition of phonological input would add several new features to Oppenheim’s model. The first enhancement adds phonological feature nodes to the neural network. These input nodes are similar to the semantic input nodes where they are triggered through a binary string, 1 for activation and 0 for off. However, these nodes are inactive during training. They receive an input of 0 for all training inputs. They are only activated during testing, particularly when phonological similarity is present in the word set. There can be multiple phonological feature nodes, each representing a different phoneme.



**Figure 10. Lexical activation levels with phonological activation to the target words and to the 2 strongest competitors.**

When a phonological feature node is activated during a simulation, lexical activation levels of words with that phonology are increased. Under the hypothesis, there must be increased activation from shared phonological onsets that would feed back to the word level and result in increased semantic interference (Preusse et al., 2013). Since a phonological feature is not activated during training, it cannot have a learned weight linking the input phonology node  $p$  to an output lexical activation node  $i$ . In place of a dynamic weight coefficient which is calculated during training a constant weight  $\phi$  is used to directly affect lexical activations.



**Figure 11. Lexical activation levels with phonological facilitation to the target words and to the 5 strongest competitors.**

A logical if-else statement is used to apply  $\phi$  directly into the net summation of activated inputs, equation 1. As the net sum is fed into the lexical activation function, equation 2, activation levels will fluctuate proportionally for words sharing a phonological feature. As will be explained in chapter 4, implementation of a test word sets were created to match as closely as possible human trail word sets, Figures 5 and 6. During a simulation, from all the trained possible words, three words could be inputted into the network, one by one, which may share a single phoneme such as PUFFIN, PIGEON, and PEACOCK. As a word enters the network, that word is considered as the desired target word, for instance PUFFIN. Note, in this example all words share both a semantic and phonological feature. PUFFIN would clearly have the highest activation level.

Figure 10 represents the activation levels of all the words (after training) when PUFFIN is the input modeled as word 12 with a lexical activation level just over 0.70. Activation levels for words can be zero or greater but cannot be negative. The left blue side bar shows activation levels for all the words when  $\varphi$  is not applied to the net summation and the red or right bar shows activations levels when  $\varphi$  is applied. Equation 7 shows the addition of  $\varphi$  to the net sum.  $\varphi$  is added only to the two strongest competitors and to the target word. It is not added for any other word  $i$ .

$$net_i = \left( \sum_j w_{ij} a_j \right) + \varphi \quad (\text{Eq. 7})$$

$\left| \begin{array}{l} \text{if a phono feature is activated } \varphi \text{ is on.} \\ \text{else } \varphi \text{ is off.} \end{array} \right.$

In Figure 11 we illustrate proportional increase in activation levels to the three words PUFFIN, PIGEON, and PEACOCK (words 10, 11, and 12). In equation 7, in order to have a proportional increase in activation levels, the constant  $\varphi$  is set to 0.9 when a word is a strong competitor word (sharing the same semantic and phonological feature) and  $\varphi$  is set to 0.09 so as to give a proportional increase for the target word level. This is actually a similar effect as if activations levels were increased by adding  $\varphi$  directly to the outcome activation of equation 2. For all other words  $\varphi$  is set to zero. This alteration in the value of  $\varphi$  makes it dynamic, however it is a systematic change in level and not one based on incremental learning of how often a word is seen. However, this increased level in activations due to  $\varphi$  does impact selection times and learning, explained in chapter 5.

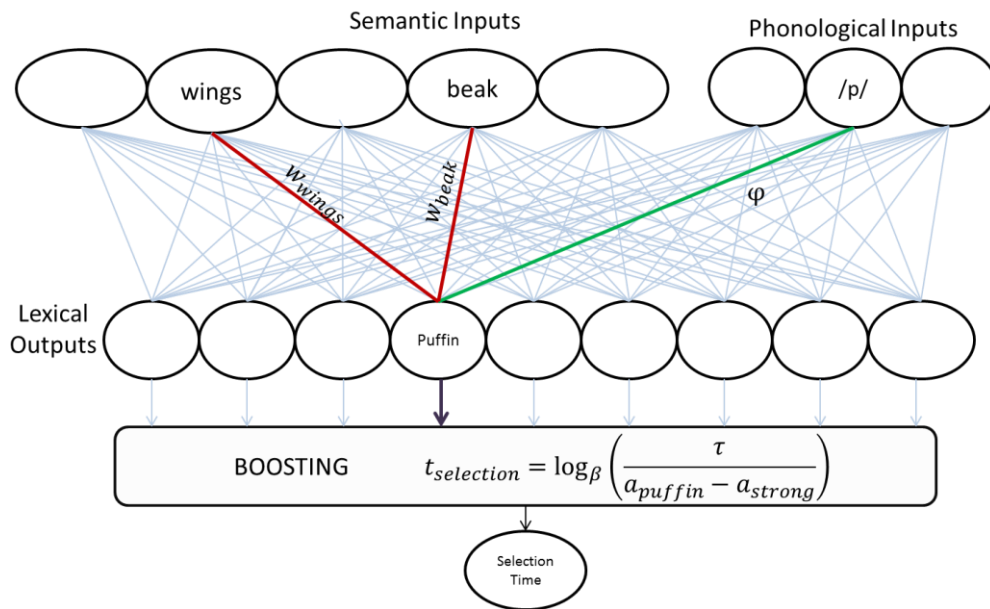
During simulation testing in chapter 5, two types of competitors were used to evaluate the effectiveness of the model under certain conditions in the competitive selection process. This is a change in the competition process in Oppenheim’s model of averaging the activation of all lexical outputs instead of just close competitors. This is done so as to see the effect on selection times. In simulation 1 we use only the 2 strongest competitor’s as described in equation 6. In simulation 2, when using the 5 strongest competitors,  $\phi$  is applied not only to the words in the current test set but also to the three words in the second most similar set (sharing a semantic feature). This phonological facilitation is seen in Figure 12 with words 1, 2, 3, 10, 11, and 12 all having their lexical activations increased.

The last major addition to Oppenheim’s model is an alteration of the logarithmic boosting function:

$$t_{selection} = \log_{\beta} \left( \frac{\tau}{a_{it1} - a_{strongest\ competitors}} \right) \quad (\text{Eq. 8})$$

$a_{strongest\ competitors}$  in equation 8 is the mean activation level of the strongest competitors from the test set. Simulations were done using two and five strongest competitors. So for instance in using two competitors this would be words 10 and 11 for the set in Figure 11. Using the strongest competitors versus the mean activation of the entire vocabulary set should create increased interference for the both condition. The reason why this could work is because the competitive selection process is not biased by other words from the entire word set which will have varying activation levels due to the number of times they’re activated and if

they share a semantic feature. Unlike the boosting function in (Oppenheim et al., 2010) which used the mean of all test activations for the differential threshold, in the phonological model it is more prudent to use the strongest competitors since the test set, described in chapter 4, is considerably larger. By focusing only on the strongest competitors we can get cleaner results because the mean difference between target word and competitor words is more stable - representing more closely competitor's impact on selection time.



**Figure 12. Phonological abstract model with example.**

In Figure 12 we illustrate a high level abstract version of the phonological model. As a word like PUFFIN enters the network the semantic features WINGS and BEAK are activated along with the phonology /p/. Net summation of  $w_{wings}$  and  $w_{beaks}$  along with  $\phi$  through equation 6 is inputted to the activation function, equation 2. Before learning, the boosting process fires up and a selection time is

calculated for the word PUFFIN from its activation difference with the mean of its two strongest competitors. Lexical activation levels are then returned to pre-boosting levels and the learning process takes place to update the weights  $w_{wings}$  and  $w_{beaks}$ . The model can be further extended to have varying  $\phi$  values for different phonemes. Though in simulations,  $\phi$  was kept the same among all phonologies.

## Chapter 4 – Implementation Details

The phonological computer model described in chapter 3 was implemented entirely in Matlab code. Matlab, which stands for matrix laboratory, is a high level programming language for numerical computation, visualization, and application development complete with user interface constructs, mathematical operators, data structures, user-defined-functions, etc. It is a coding language used primarily in the fields of math, engineering, science, and economics in academia and industry due to its easy algorithmic prototyping abilities and its built in mathematical functions. Matlab also has an array of toolboxes which provide additional computational support in specific areas from econometrics to neural networks. Matlab also provides support with integrating programs with other languages such as C, Java, .NET, and Microsoft Excel.

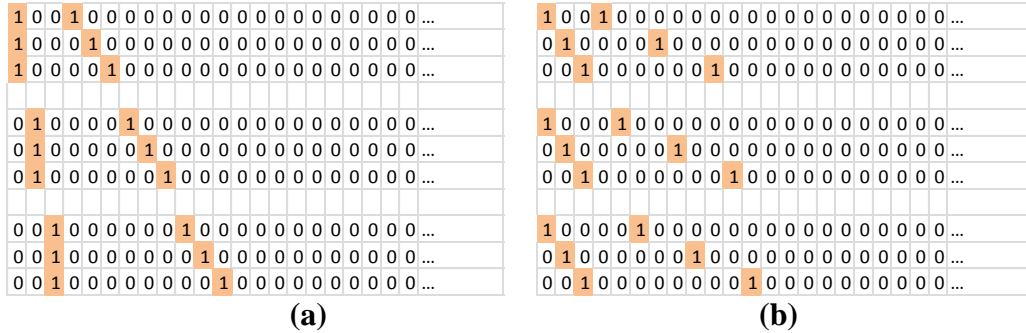
Variables in Matlab can be assigned without declaration types, so a variable such as “y” can be defined as a string by simple equating it to a string value:  $y = \text{'string'}$ , or it can be set as an integer or as a double:  $y = 1$  or  $y = 1.2$ . A powerful feature in Matlab is vector and matrix manipulation. It has the capability to assign matrices of any dimensions and to easily perform operations with them such as addition and multiplication and can easily reference positions in the matrix. A 2x2 dimensional array or matrix can easily be assigned as such:  $X = [1 \ 2; \ 3 \ 4]$ . While Matlab has classes, typically a user would code up functions instead that can represent classes. Functions can also act like methods and can accept any form of input from objects to string elements.



## 4.1 Simulation Test Sets

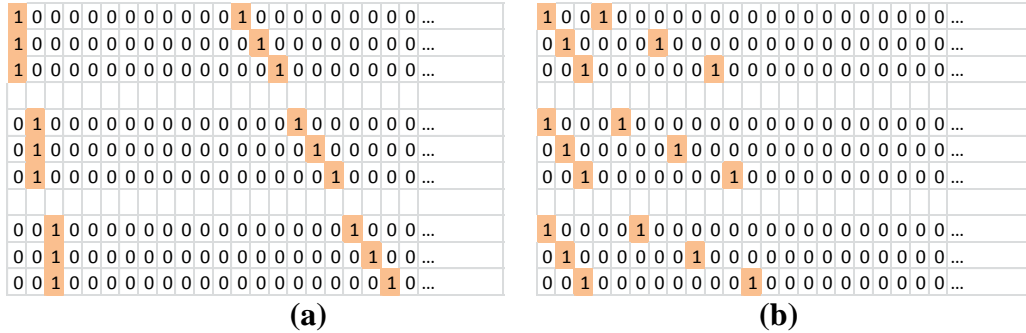
Similar to the simulation word set in (Oppenheim et al., 2010), shown in chapter 2, the test sets devised to evaluate the performance of the phonological model also consisted of binary values to represent features. There are 216 words in the whole vocabulary used for training, with 72 features. Words ordered 1 to 18 and 106 to 126 were used to create the *semantic*, *both*, and *phonological* condition test sets. All other words were not used but were needed during training to ensure that every feature of every word was seen in each epoch at least once in order to prime them for testing. Phonological feature activations are not present in the training or test sets but are rather hard coded to be activated when the phonology simulation sets are inputted to the model.

Simulation sets were devised to model the test conditions in human trails described in chapter 3.1. Similar to those trails, three conditions were tested in simulations: semantic, both, and phonological, each composed of homogeneous and heterogeneous word sets. In addition, simulations further used two kinds of heterogeneous sets. In replicating test sets from the human trails, heterogeneous sets for each condition were composed of words with no overlapping semantic features but the words were a reordering of those used in the homogeneous sets. A separate heterogeneous set for each condition was also devised that shared no words with the homogeneous sets. This second set with no common words to the homogeneous sets more closely resemble tests sets used in simulations trails in (Oppenheim et al., 2010).



**Figure 13. Semantic condition simulation test sets. (a) homogeneous sets, (b) heterogeneous sets.**

Figure 13 shows the test simulations for the semantic condition. Not all features are shown. Figure 13 (a) represents the three homogeneous word sets, each containing three words. Each word in one of the three sets shares the same category or one semantic feature with the other words. But no words shared the same category among sets. Figure 13 (b) shows the three word sets for the semantic heterogeneous condition. Each is a reordering of words from the homogeneous sets. No word within a set shared any features. The separate heterogeneous condition used for testing which had no overlapping words with the homogeneous sets were similar to Figure 13 (b) but all features were shifted over by 36 positions. Figure 14 shows the style of the simulation sets for the both condition. Figure 14 (a) shows the homogeneous condition where each set shares a similar category with the semantic homogeneous sets. Heterogeneous sets in Figure 14 (b) were devised in a similar manner to the semantic condition, where they are simply a reordering of words from the homogenous condition.



**Figure 14. Both condition simulation test sets. (a) homogeneous sets, (b) heterogeneous sets.**

## 4.2 Code Structure

All methods were programmed in Matlab and saved as m files. A full simulation consists of four files: *features\_set.m*, *training.m*, *SemanticNet.m*, and *simulation.m*. The first file is the *features\_set.m* which has the entire training vocabulary hardcoded as a large two by two matrix of all 216 training words. It is also from this file that test words are extracted as inputs to the phonological model during test simulations. The *training.m* file is responsible for training the model and setting appropriate weights in the network from input semantic nodes to output lexical nodes as seen in Figure 10. During training 100 epochs take place where in each epoch a full randomized cycle of the entire feature set is seen.

When a word during training or testing is seen, the *SemanticNet.m* file is activated. This file is essentially the entire network coded and is the most important file. As described in chapter 3, three steps are required to make the phonological model work. These are the lexical activation phase, the lexical selection phase, and the learning phase. All three phases are modeled as three Matlab functions with the same names. The lexical activation function receives as

an input the input word activation, a trained weights matrix, a noise parameter and five other parameters which determine if the method is being used during training or testing and what condition is being tested, for instance semantic vs. phonological activation. During lexical activation equation 6 and 2 are calculated with output being a matrix of size  $i$  of the lexical activation levels. The lexical selection function implements the new boosting method described in chapter 3. Boosting parameters such as boosting rate, threshold, omega, and activation level matrix are inputted and a boosting matrix is outputted for each lexical node. As in the Oppenheim model, if the total number of boosts for a word surpasses the omega value (i.e. it's taking too many boosts to reach the threshold  $\tau$ ) then no selection time value is reported for that word. Finally the learning phase takes place where weight values are updated after the lexical activation phase takes place using equation 6.

The file *simulation.m* is responsible for setting all the testing parameters. In this file all network variables are set and all simulation word sets are set as well. This file then sets up each simulation condition such as the semantic test condition or the both condition. Depending on the condition tested, parameters and methods can be customized in this file. Every other file is set so it does not need any modification during different simulations.

## Chapter 5 – Simulations and Discussion

The phonological model adds two main new additions to the Oppenheim lexical learning model: adding a phonological similarity feature to the lexical input and using the strongest competitors during competitive lexical selection. The goal of this model is to account for phonological effects on semantic interference. To ensure that, we conduct several experiments that have been designed to match human test trail conditions described in chapter 3. However, since this thesis is not a cognitive neuroscience one, we only evaluate the phonological neural network model to the extent that it can replicate human results.

We use the vocabulary described in chapter 4 for training and testing. When a target picture is inputted to the model we assume that its semantic features are activated appropriately. Phonological activations are hard coded to correspond with appropriate word test sets. For all simulations we test the blocked cyclic picture naming paradigm using seven cycles for each word set. Each simulation consists of a training and test phase. As stated in (Oppenheim et al., 2010), the training phase simulates a subject learning the set of vocabulary words, acquiring lexical semantic knowledge.

Three simulations were carried out to assess the phonological model. Simulation 1 evaluates the model when applying increased phonological lexical activation to two competitors as well as using only those competitors for calculating lexical selection time during the boosting process. Four tests were

conducted for simulation 1, each test altered the order of the semantic and both conditions as they were presented to the model and another two tests were conducted using different heterogeneous word sets. These tests were also designed to evaluate carried over priming effects of learned words. Simulation 2 replicated simulation 1 and its four tests by using five competitors in order to see broaden effects of competition in lexical selection and the boosting process. Simulation 3 replicated test 1.1 from simulation 1 but had its learning algorithm modified slightly to update weight links of competitor words and features versus just that of the target word. This was done to see if the incremental learning process could be enhanced and yield higher interference levels that would match that of human trails.

### **5.1 Simulation 1 – Using 2 Strongest Competitors**

Here we test the semantic and both conditions using only the two strongest competitors in calculating selection times as described in chapter 3. Oppenheim et al. (2010) reported increased cumulative interference effects with each intervening word during simulations of the blocked cyclic naming paradigm. Our model, like the Oppenheim model implements all three properties, lexical selection, competitive selection, and priming, and should also exhibit incremental increase in selection times. A major finding in this simulation is that the phonological model does not experience cumulative semantic interference although total interference effects were significantly greater in the both condition. While all three conditions are tested, semantic, both, and phonological, only the

semantic and both conditions are examined closely. Due to ambiguous results from human trails (Preusse et al., 2013), it is not yet known the exact influence of interference effects from phonological similarity alone without semantic similarities already present in word sets.

A constraint on the model is that it can test word sets in randomized order. During one test, the semantic condition is tested first then the both condition. A second test is conducted with the two conditions having switched order. This is done to analyze the effect of how ordering of conditions may result in carry over priming effects of already seen words. Two more test were conducted, for a total of four tests, where the same conditions are tested as with the first two tests except that in the semantic and both heterogeneous sets, words do not overlap with words in homogeneous sets.

### 5.1.1 Simulation 1 Method

The network was trained with 100 randomly oriented epochs using the vocabulary set described in chapter 4 for each of the four testing phases. Parameters for each test in this simulation are summarized in Table 2.

Parameter	Value
Boosting Rate ( $\beta$ )	1.01
Threshold ( $\tau$ )	1
Deadline ( $\Omega$ )	100
Learning Rate ( $\eta$ )	0.75
Activation Noise ( $\theta$ )	0.5
Phonological Rate ( $\phi$ )	0.90

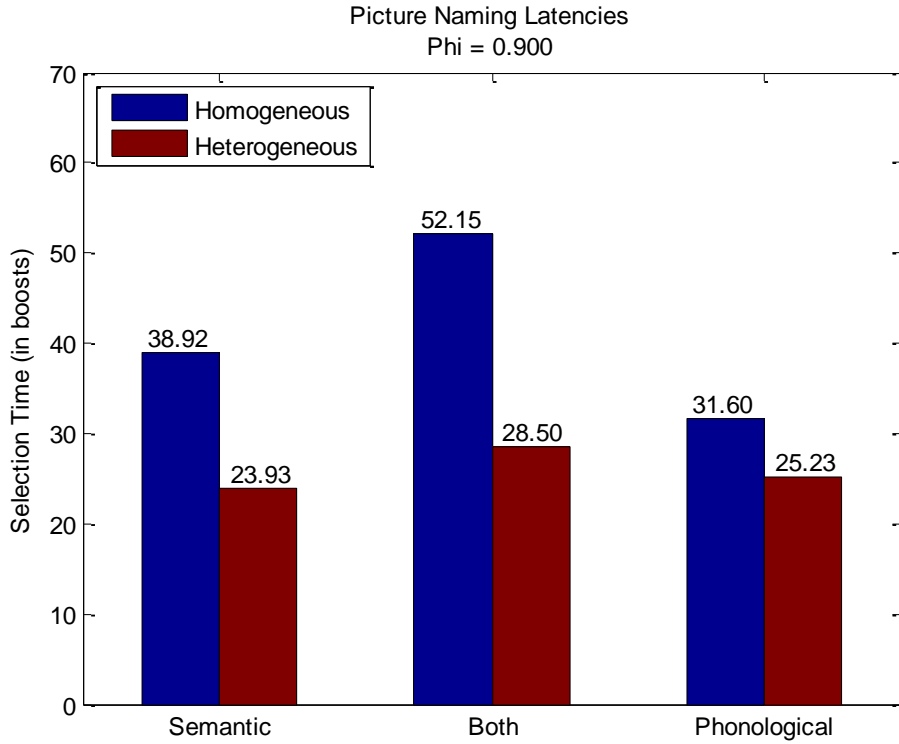
**Table 2. Model parameters for simulations.**

Under the blocked cyclic naming paradigm, sets of three words were cycled seven times for a total of 21 cycles. In each cycle, the three words were presented to the network in a random order. A total of three word sets were used in each homogeneous and heterogeneous group for each condition. Each homogeneous set belonged to a single semantic and/or phonological category. Words in a heterogeneous set all belonged to different categories. All three word set results were averaged after each test for each condition. Selection times are reported as the mean number of boosts.

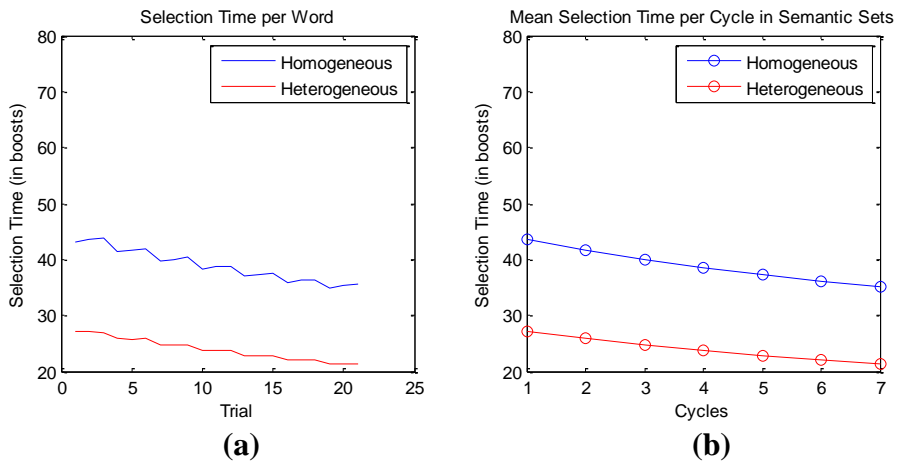
### **5.1.2 Simulation 1 Results**

Here we show results for test 1.1 where the semantic condition was tested first, followed by the both condition, and last by the phonological condition. Heterogeneous sets contained words which are a reordering of words from the homogeneous sets. Tests 1.2, 1.3, and 1.4 are shown in the Appendix under simulation 1 test plots. Test 1.2 switched the ordering of which the semantic and both conditions were presented to the model and tests 1.3 and 1.4 replicated the first two tests but used different heterogeneous word sets which did not share any words with the homogeneous sets.. Selection time plots for each word and mean selection time plots derived from the first plot are conducted for each condition, giving a total of six plots. For instance, Figure 16 (a) shows the selection per word in each cycle, and (b) shows the mean selection times for each cycle. Lastly, we show a picture naming latency plot that averages the total selection times over all seven cycles for all three conditions.

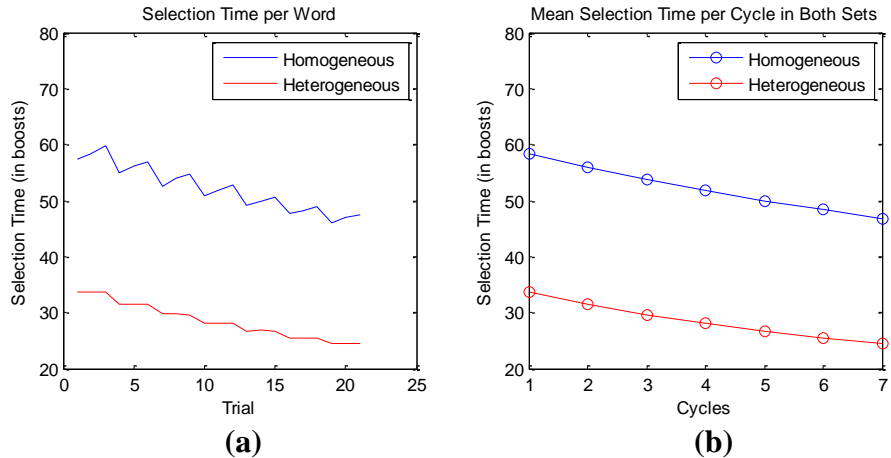




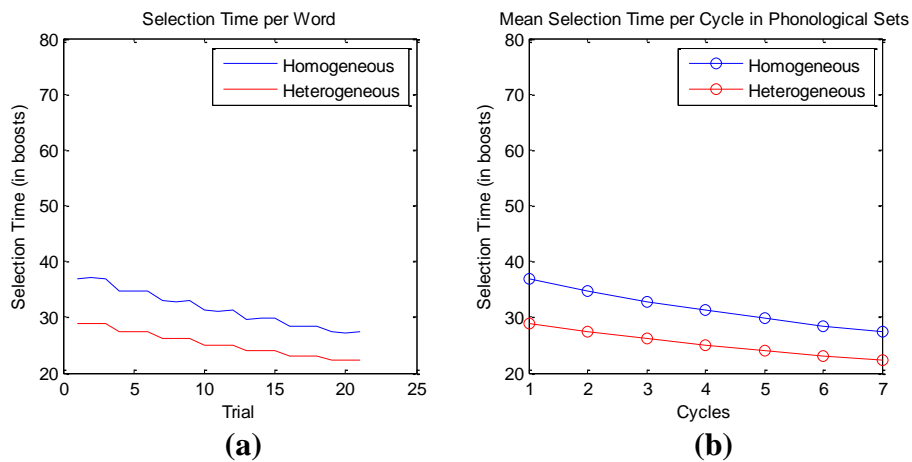
**Figure 15. Total mean selection times for each of three conditions.**



**Figure 16. Semantic condition simulation test 1.1 (a) selection time per word, (b) mean selection time per cycle.**



**Figure 17. Both condition simulation test 1.1 (a) selection time per word, (b) mean selection time per cycle.**



**Figure 18. Phonological condition simulation test 1.1 (a) selection time per word, (b) mean selection time per cycle.**

Figures 15 to 18 show results of test 1.1. Tests 1.2 to 1.4 are shown in the appendix. A major find in this simulation was that there were no cumulative semantic interference effects with each cycle in the homogeneous sets. Repetition priming effects are however present, where after each time a word is seen its selection time in the next cycle is smaller signifying that it was selected faster. Incremental learning has facilitated future lexical retrieval but did not increase selection times with each cycle to create cumulative effects.

### **5.1.3 Simulation 1 Discussion**

Several more interesting findings are inferred from the results. Interference is already present in the homogeneous sets at the start of the cycle trials. This interference effect along with the lack of a cumulative effect over trials can only be explained by the design of the test sets. In tests 1.3 and 1.4, found in the appendix, we use distinct heterogeneous groups which resulted in slightly decreased interference effects as well as small cumulative effects at the start of the cycles. This shows that using mixed sets with overlapping words from the homogeneous sets will have carried over priming effects.

In addition, when analyzing all tests, ordering of the semantic and *both* conditions when using two competitors does not have a large impact on interference when applying phonological facilitation to only two competitors. When also using none overlapping heterogeneous word set, we see in the picture naming latency plots of tests 1.3 and 1.4 in the appendix a larger interference difference between the semantic and both conditions than when using overlapping mixed sets seen in Figure 15.

### **5.2 Simulation 2 – Using 5 Strongest Competitors**

In this simulation instead of applying phonological facilitation to only the 2 strongest competitors, and then using only those 2 competitors in calculating selection time in the boosting process, we replicate simulation 1 by using the 5 strongest competitors. By extending out to more competitors we can evaluate the

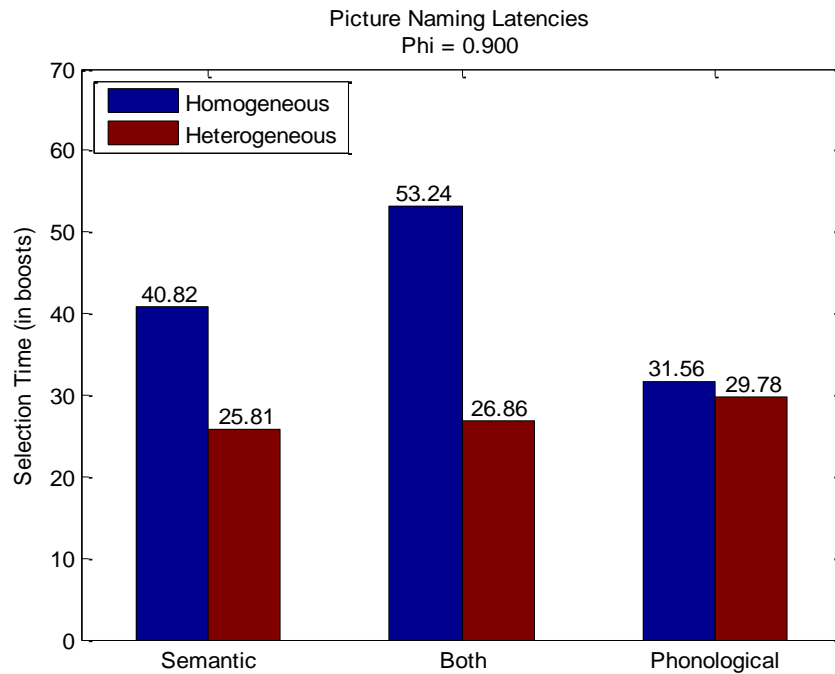
effect of phonological facilitation to all words which share a common category.

As in simulation 1, four test phases are conducted again.

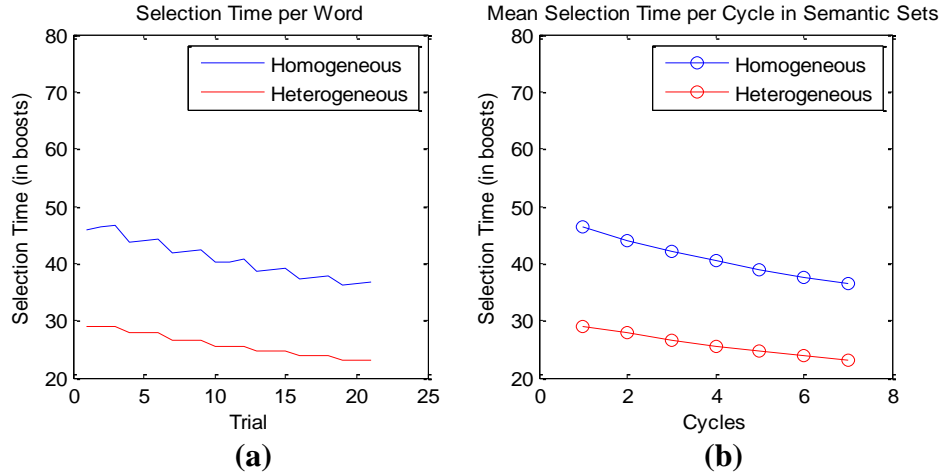
### 5.2.1 Simulation 2 Method

The exact same methodology is used here as in simulation 1 and the same parameters are used from Table 2.

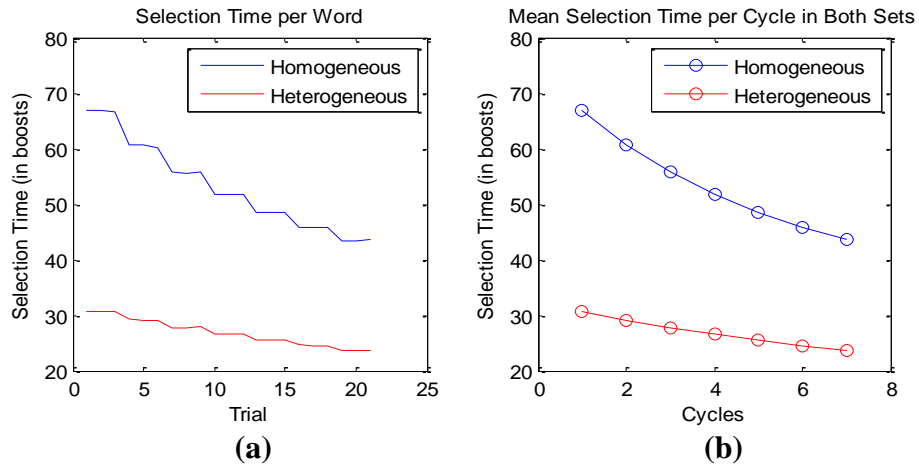
### 5.2.2 Simulation 2 Results



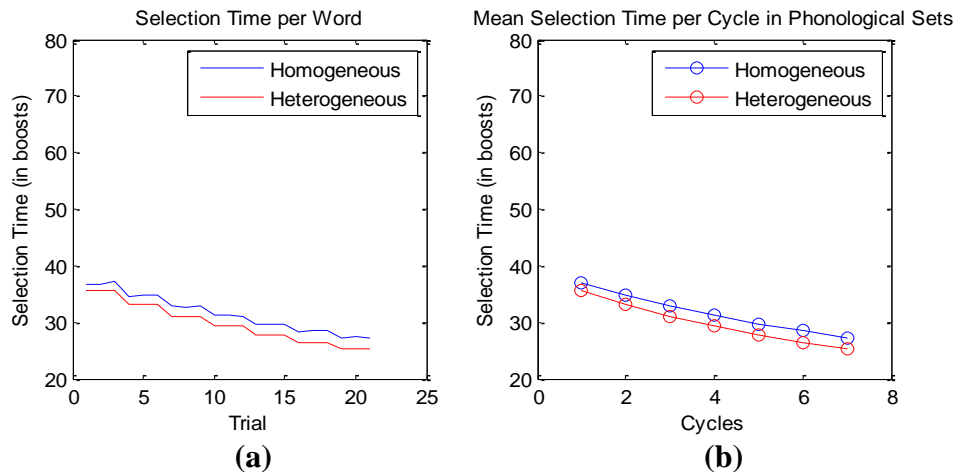
**Figure 19. Total mean selection times for each three conditions.**



**Figure 20. Semantic condition simulation test 1. (a) selection time per word, (b) mean selection time per cycle.**



**Figure 21. Both condition simulation test 1. (a) selection time per word, (b) mean selection time per cycle.**



**Figure 22. Phonological condition simulation test 1. (a) selection time per word, (b) mean selection time per cycle.**

Figures 19 to 22 show results of test 2.1. Tests 2.2 to 2.4 are shown in the appendix. Extending phonological facilitation to more competitors provides some interesting results. Again, as with using only 2 competitors, cumulative semantic interference is not present. In fact the opposite occurs in test 2.1. In Figure 21 for the *both* condition, we see an increased priming effect for future retrieval for the homogeneous condition without any interference effect.

### 5.2.3 Simulation 2 Discussion

With every cycle, interference effects appear to actually decrease in the both condition. They are stable though in the semantic condition in Figure 20. In test 2.3, where we don't use overlapping mixed word sets, we also see decreased interference effects with each passing cycle in the selection time plots. A large selection time gap is also present in the beginning trails of all four tests similar to trails in simulation 1. No interference effects are present in the *phonological* condition.

Unlike simulation 1, altering the order of testing each condition appears to have a substantial impact on the outcome. In tests 2.2 in the appendix, when applying the both condition first, carried over priming effects appear to decrease interference when later testing the semantic condition. This also appears to be the case when there no overlapping words between homogeneous and mixed sets in test 2.4, indicating that carried over priming effects occurred in the homogeneous sets sharing a common semantic feature between the both and semantic conditions. By increasing activation levels to more competitors when a phonological feature is present, priming effects are magnified.

### 5.3 Simulation 3 – Adjusted Learning

In the Oppenheim model, weight updates apply only to activated features. Only the features which are connected to the activation node of the target word will get their links updated. So in effect there are no changes to non-shared features. Considering the *semantic* and *both* conditions, we compute a model with learning adjustments to all features. We apply adjustments to the links of all features for each target and competitor. The way to do this would be simply to trace back the links from each target and competitor to its features and apply equation 6 but excluding the last element  $a_j$  which traces back to the input semantic node. This becomes:

(Eq. 9)

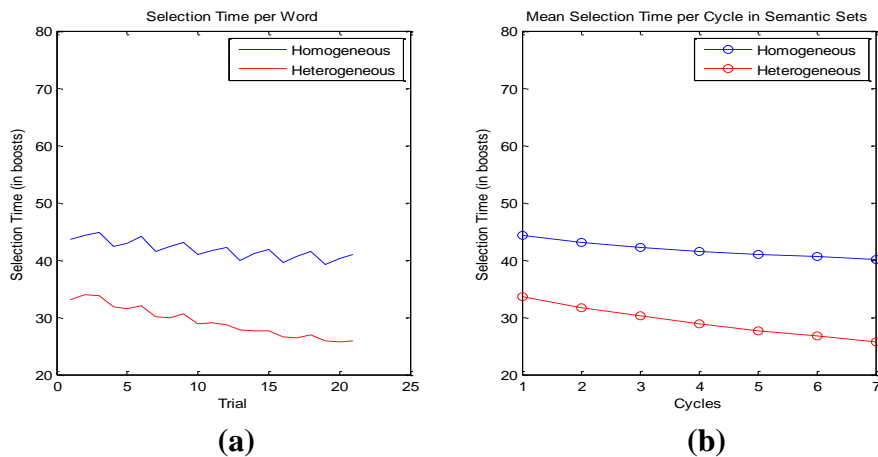
$$\Delta w_{ij} = \eta(a_i(1 - a_i)(1 - d_i))$$

By providing weight coefficient adjustments to all features, we should not only see increased interference for the *both* condition but should also see cumulative semantic interference effects now, unlike simulations 1 and 2.

### 5.3.1 Simulation 3 Method

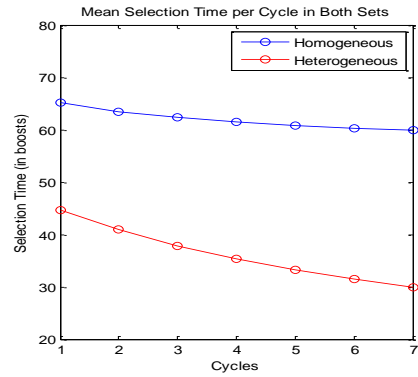
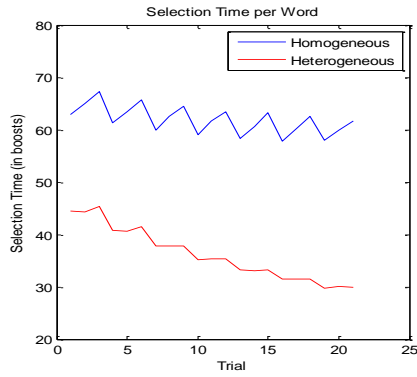
Here for simplicity we conduct only one test, under the blocked cyclic picture naming paradigm again using 7 cycles, for adjusted learning using mixed groups with no overlapping words and testing the *semantic* condition first, followed by the *both* and last the *phonological* condition. We use the two strongest competitors as in simulation 1. Also according to simulation 1, ordering of conditions did not affect outcomes, particularly when heterogeneous sets did not overlap with the homogenous sets, so we replicate here test 1.3 using equation 9. The same parameters are used from Table 2 for this test with 100 training epochs conducted before the test.

### 5.3.2 Simulation 3 Results



**Figure 23. Semantic condition simulation test 1. (a) selection time per word, (b) mean selection time per cycle.**

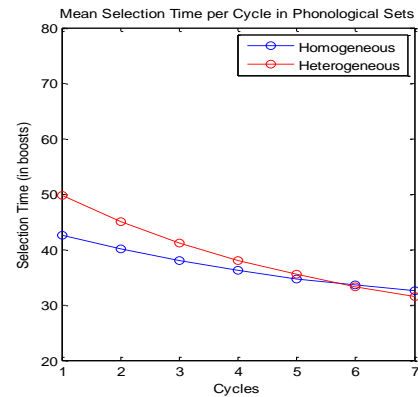
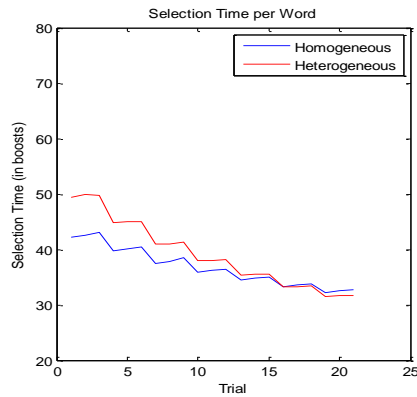




(a)

(b)

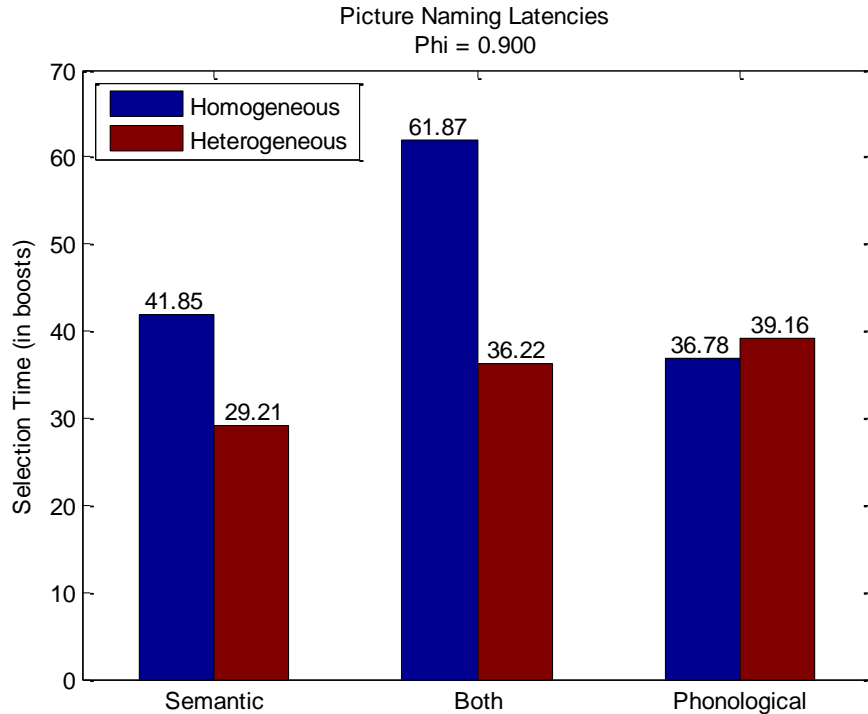
**Figure 24. Both condition simulation test 1. (a) selection time per word, (b) mean selection time per cycle.**



(a)

(b)

**Figure 25. Phonological condition simulation test 1. (a) selection time per word, (b) mean selection time per cycle.**



**Figure 26. Total mean selection times for each three conditions.**

### 5.3.3 Simulation 3 Discussion

With adjusted learning we observe higher interference effects than in simulations 1 or 2. We see again interference at the beginning of the cycle trails for the semantic and both conditions, but this time we observe cumulative interference effects in the semantic and both conditions as seen in the diverging homogenous and heterogeneous curves in Figures 23 and 24. Semantic interference is also considerably greater in the both condition, maxing at over twice the interference present in the semantic condition in figure 26. In Figure 25, we see that selection time for homogenous sets decreases below selection time of mixed sets. These results indicate that using adjusted learning yields a better phonological model.

## Chapter 6 – Conclusion

As a person hears or sees a picture of an object they would look up the corresponding word from memory. Retrieving a word from memory multiple times enables a person's lexical retrieval process to prime the word so that it can be picked faster with every time. It has been argued (Howard et al., 2006) that retrieving a word induces an incremental learning process which can have interfering effects known as cumulative semantic interference. Oppenheim et al. (2010) presented a computer model that could simulate such behaviors. It has been hypothesized (Preusse et al., 2013) that when a person hears a series of words which not only have a common trait with each other, such as belonging to a similar semantic category, but also sound similar by having shared sound components (phonemes) would result in increased semantic interference. This thesis explores the formation of a neural network computer model to study the effects of shared phonological onsets on lexical retrieval.

Chapter 1 introduced the field of computational cognitive neuroscience and the importance of computer models in simulating various phenomena. It also introduced the phenomenon of cumulative semantic interference as well as the three cognitive properties necessary to manifest it: lexical selection, competition, and priming. In providing a context in which to produce semantic interference a picture naming paradigm known as the blocked cyclic paradigm was described which is used in word production studies. Lastly, fundamental concepts of feed

forward neural networks were discussed that provide the backbone in creating a model of cumulative semantic interference.

The Oppenheim model is presented in detail in chapter 2 which is based on an error driven incremental learning algorithm. It represents lexical retrieval as node activations in a neural network and includes a “winner takes all” method in modeling lexical competition. Input data is modeled as an array of binary values where each value represents a word and each 1 represent a semantic feature of that word. As a word enters the network lexical output nodes are activated. These activations are fed into a boosting process which simulates competitive selection by increasing each words activation level until a single word emerges with the highest level that passes a differential threshold. The model then updates weight coefficients through the error driven learning algorithm between lexical output nodes to semantic input nodes which represent semantic to lexical mappings.

In chapter 3 we present the phonological model which extends the Oppenheim model to include the effect of phonological similarities in word inputs. We describe the phonological interference hypothesis presented by Preusse et al. (2013) as well as their human experimental trails. These trails and their outcomes provide constraints and a benchmark for the creation of the phonological computer model. Trails are conducted under the blocked cyclic picture naming paradigm in examining three conditions: semantic, both, and phonological. Each condition uses a test word set which shares semantic features, phonological and semantic features, or just phonological features. Each condition

also divides word sets into homogeneous and heterogeneous groups where in a homogeneous set, words have a semantic and/or phonological feature in common and heterogeneous sets have no features in common. In human trails, heterogeneous word sets were created by a reordering of homogeneous word sets. Utilizing the constraints of the human trails, the framework for the phonological model is described. This uses a whole new word test set and extends several mechanisms to accommodate the use of phonological features in lexical selection and competitive selection. Chapter 4 then discusses the structure of the test sets as well as the structure of the Matlab code that the model was implemented in.

In chapter 5 we explore the use of the phonological model to assess its behavior under certain conditions. Three main simulations are conducted. The first two simulations assess the model's capability in dealing with different ordering of test sets and test conditions and analyzing carry over priming effects. These simulations also tested a second separate heterogeneous group which did not use words already seen in homogeneous sets, unlike some human trails, in order to examine the effect of none shared words. In simulations 1 and 2, interference was higher in the both condition but cumulative effects over cycles were either minute or absent. Simulation 3 introduced a change in the learning algorithm. Through this learning adjustment, we saw significant results in cumulative semantic interference effects over cycles as well as higher levels in interference in the *both* condition.

## 6.1 Future Work

Based on the simulation conducted, more work is suggested to extend the phonological model described in chapter 3 by the learning adjustments seen in simulation 3; applying weight modifications to competitor feature links along with adjustments of semantic to lexical links of the target word result in increased interference in the both condition and cumulative interference over time through incremental learning. More simulations could be conducted to evaluate the influence of this adjustment to learning under different word test sets.

Additionally, the changes to learning could further be extended to create a new condition where adjustments are made to competitors who only have a shared context with the target word. This could be called the context condition. We want to implement semantic interference among items that share no intrinsic semantic features but are activated by a shared context. This could be equated to the phonology only condition. The only difference is that there are no shared features but items linked by a context are nonetheless co-activated and therefore trigger incremental learning adjustments.

Motivation for a context based model is to simulate Remote Associate's Task (RAT) problems (Bowers et al., 1990). Learning how knowledge is reorganized in our minds is fundamental because of the large amounts of new information we learn each day. RAT looks at how we reorganize information based on new contextual information and how we can generate insightful solutions to problems. Bowers et al (1990) found that individuals can form

contexts toward a solution before solving a problem which proposes that insight problems may be solved gradually and unconsciously at first. An example of the remote associates test is when you have three pictures given to you such as BLUE, KNIFE, and COTTAGE, and the job of the subject or model is to solve how these words are associated. In this case these words are linked through the word CHEESE. The phonological model could be extended to create a context based model applying boosted lexical activations and weight updates to words whose features have a common context.

## **6.2 Final Remarks**

This work provides a framework for accommodating phonological features along with semantic features in producing semantic interference in picture naming tests utilizing a feed-forward neural network. Changing the number of competitors that are evaluated during the phonological facilitation and boosting processes of the model showed that it is possible to increase semantic interference in the presence of common phonological features versus having common semantic features alone. Further improvements in performance and in the creation of cumulative semantic interference came in the form of an adjustment in the learning algorithm for updating weight values of the neural network. Overall, our computer model offers an account of phonological influence on word production and has potential to be used in the study of other issues in memory and learning.

## References

- A.K. Hartmann, Practical Guide to Computer Simulations, Singapore: World Scientific, 2009
- Belke, E., Meyer, A. S., & Damian, M. F. "Refractory effects in picture naming as assessed in a semantic blocking paradigm." *Quarterly Journal of Experimental Psychology Section A – Human Experimental Psychology*, 58(4), (2005): 667–692.
- Bishop, Christopher M., and Nasser M. Nasrabadi. Pattern recognition and machine learning. Vol. 1. New York: springer, 2006.
- Blumer, Anselm. "Occam's razor." *Information processing letters* 24.6 (1987): 377-380.
- Bowers, Kenneth S., et al. "Intuition in the context of discovery." *Cognitive psychology* 22.1 (1990): 72-110.
- Burton, Richard M., and Børge Obel. "The validity of computational models in organization science: From model realism to purpose of the model." *Computational & Mathematical Organization Theory* 1.1 (1995): 57-71.
- Colin Prentice, I., Martin T. Sykes, and Wolfgang Cramer. "A simulation model for the transient effects of climate change on forest landscapes." *Ecological modelling* 65.1 (1993): 51-70.
- Damian, M. F., & Als, L. C. "Long-lasting semantic context effects in the spoken production of object names." *Journal of Experimental Psychology: Learning, Memory and Cognition*, 31(6), (2005): 1372–1384.
- Damian, M. F., Vigliocco, G., & Levelt, W. J. "Effects of semantic context in the naming of pictures and words." *Cognition*, 81(3), (2001): B77–86.
- Domingos, Pedro. "The role of Occam's razor in knowledge discovery." *Data mining and knowledge discovery* 3.4 (1999): 409-425.
- Duda, Richard O., Peter E. Hart, and David G. Stork. Pattern classification. John Wiley & Sons, 2012.
- Howard, D., Nickels, L., Coltheart, M., & Cole-Virtue, J. "Cumulative semantic inhibition in picture naming: Experimental and computational studies." *Cognition*, 100(3), (2006): 464–482.



- Masters, J. (2011). "Hurricane Forecast Computer Models." Weather Underground. Retrieved June 23, 2013, from <http://www.wunderground.com/hurricane/models.asp>
- McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. "Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory." *Psychological Review*, 102(3), (1995): 419–457.
- Mukherjee, Shubendu S., et al. "Wisconsin Wind Tunnel II: a fast, portable parallel architecture simulator." *Concurrency, IEEE* 8.4 (2000): 12-20.
- Myung, In Jae, and Mark A. Pitt. "Applying Occam's razor in modeling cognition: A Bayesian approach." *Psychonomic Bulletin & Review* 4.1 (1997): 79-95.
- Oppenheim, G., Dell, G., & Schwartz, M. "The dark side of incremental learning: A model of cumulative semantic interference during lexical access in speech production." *Cognition* (2010), 114, 227-252.
- Oppenheim, G. M., Dell, G. S., & Schwartz, M. F. "Cumulative semantic interference as learning." *Brain and Language*, (2007): 103, 175–176.
- Preusse K., Lewis J., Prieto C., Frazer A, & O'Séaghdha P.. "Exploring Phonological Preparation and Semantic Interference in Word Production: Phonological Similarity Boosts Semantic Interference." Poster presented at the Eastern Psychological Association Conference. (2013).
- Rosenblatt, Frank. "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review* 65.6 (1958): 386.
- Schnur, T. T., Schwartz, M. F., Kimberg, D. Y., Hirshorn, E., Coslett, H. B., & Thompson-Schill, S. L. "Localizing interference during naming: Convergent neuroimaging and neuropsychological evidence for the function of broca's area." *Proceedings of the National Academy of Sciences*, 106(1), (2009): 322–327.
- Welch, William, et al. "Computer experiments for quality control by parameter design." *Journal of Quality Technology* 22.1 (1990): 15-22.
- Widrow, B., & Hoff, M. E. (1960). Adaptive switching circuits. IRE WESCON convention record. New York: IRE (Reprinted in Anderson, J. A., & Rosenfeld, E. (Eds.) (1988). *Neurocomputing: Foundations of research* (pp. 123–134). Cambridge, MA: MIT Press.).

Witten, Ian H., and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

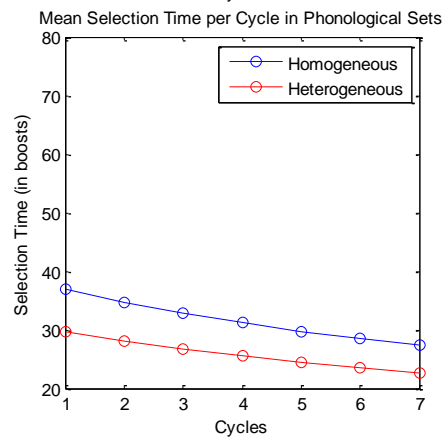
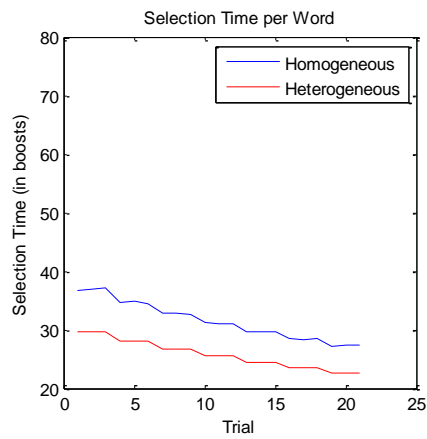
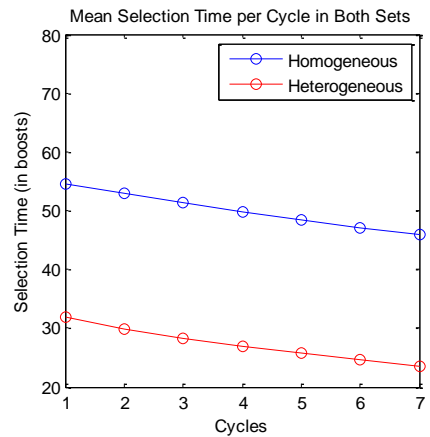
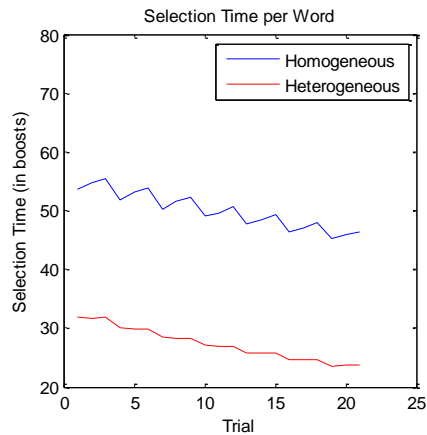
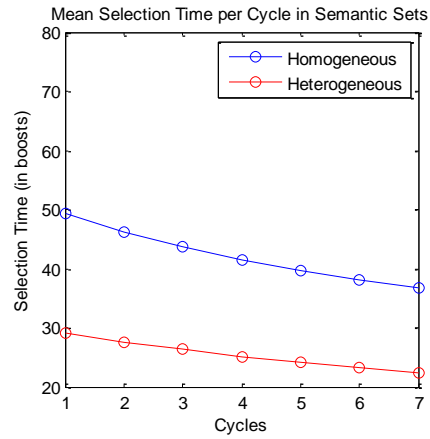
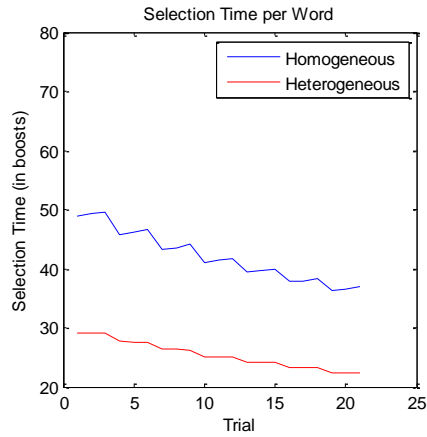
Young, Peter, Stuart Parkinson, and Matthew Lees. "Simplicity out of complexity in environmental modelling: Occam's razor revisited." *Journal of applied statistics* 23.2-3 (1996): 165-210.

Zhang, Guoqiang Peter. "Neural networks for classification: a survey." *Systems, Man, and Cybernetics, Part C: Applications and Reviews*, IEEE Transactions on 30.4 (2000): 451-462.

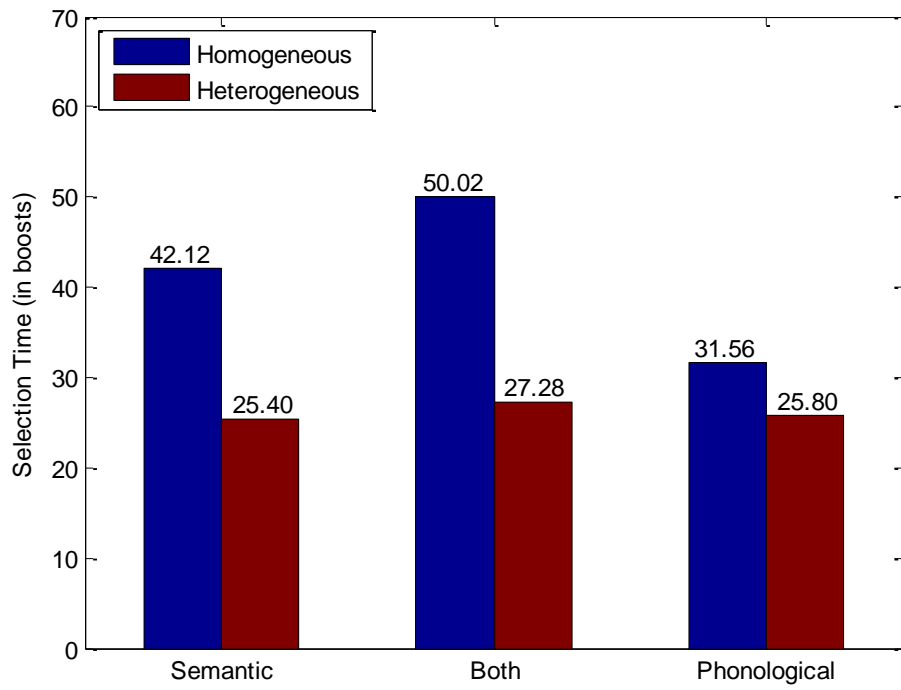
# Appendix

## Simulation 1 Test Plots

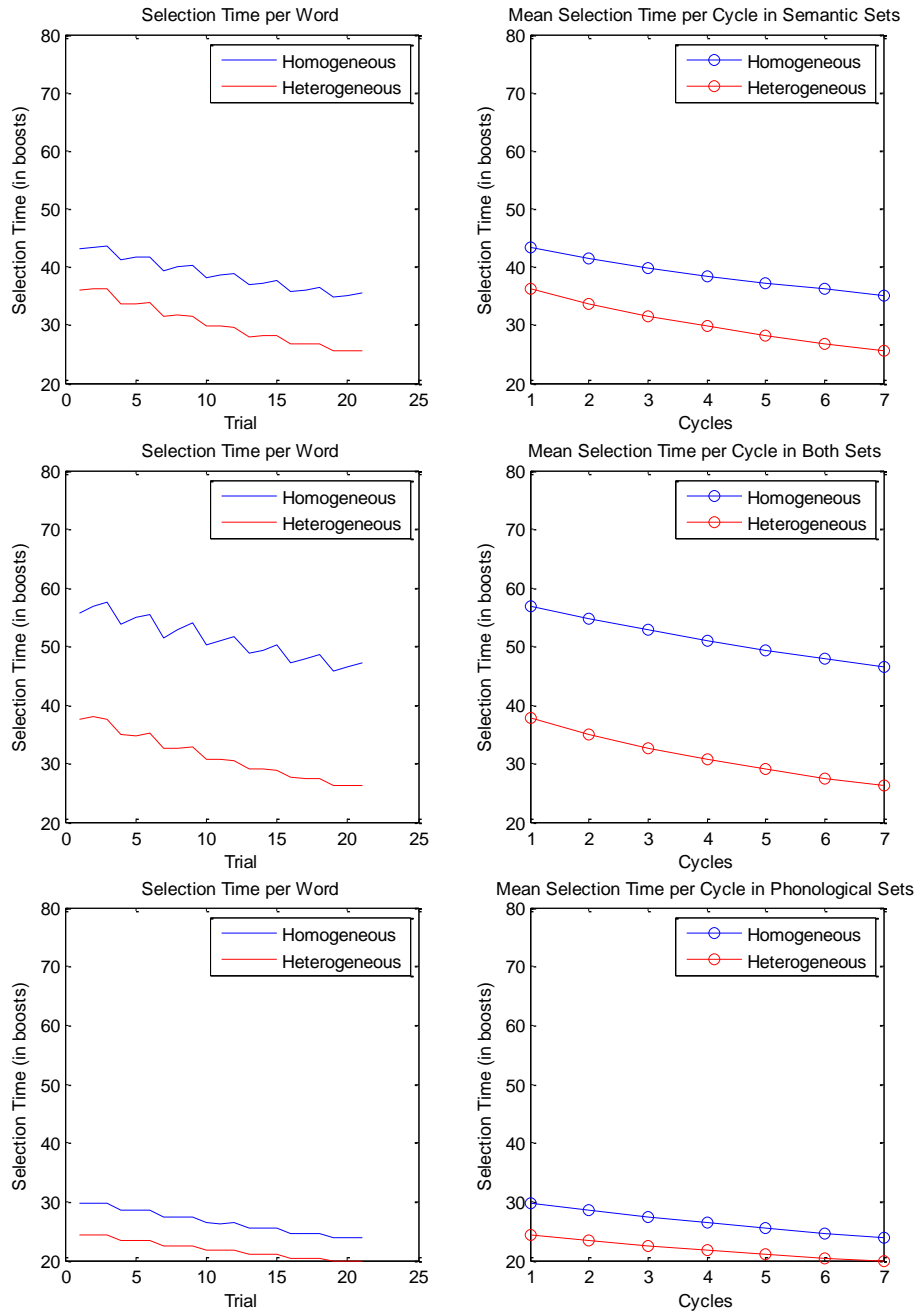
**Test 1.2** – *both* condition first followed by semantic followed by phonological condition. Heterogeneous sets overlap words with homogeneous sets.



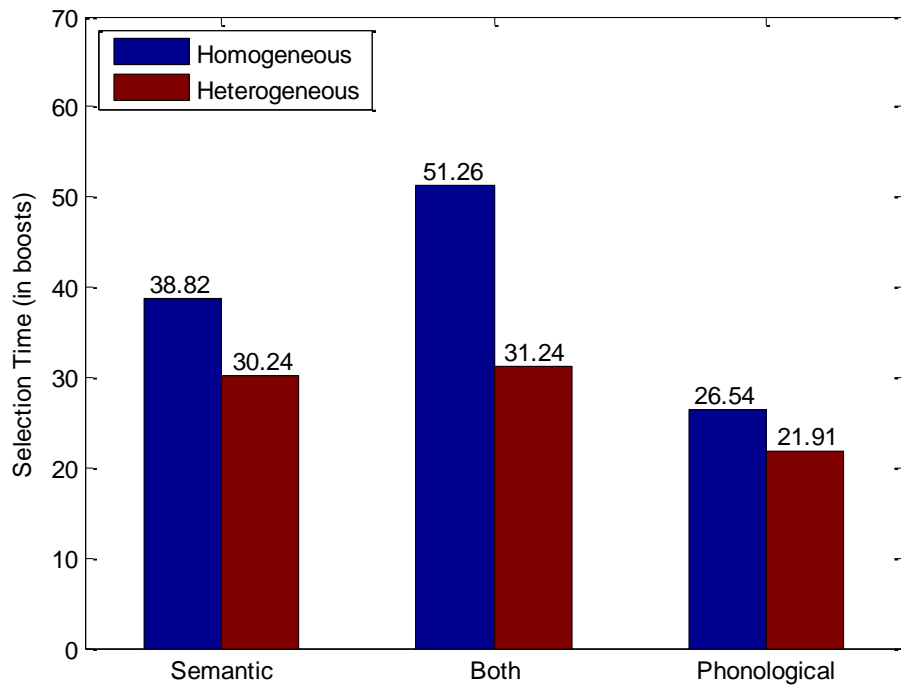
Picture Naming Latencies  
Phi = 0.900



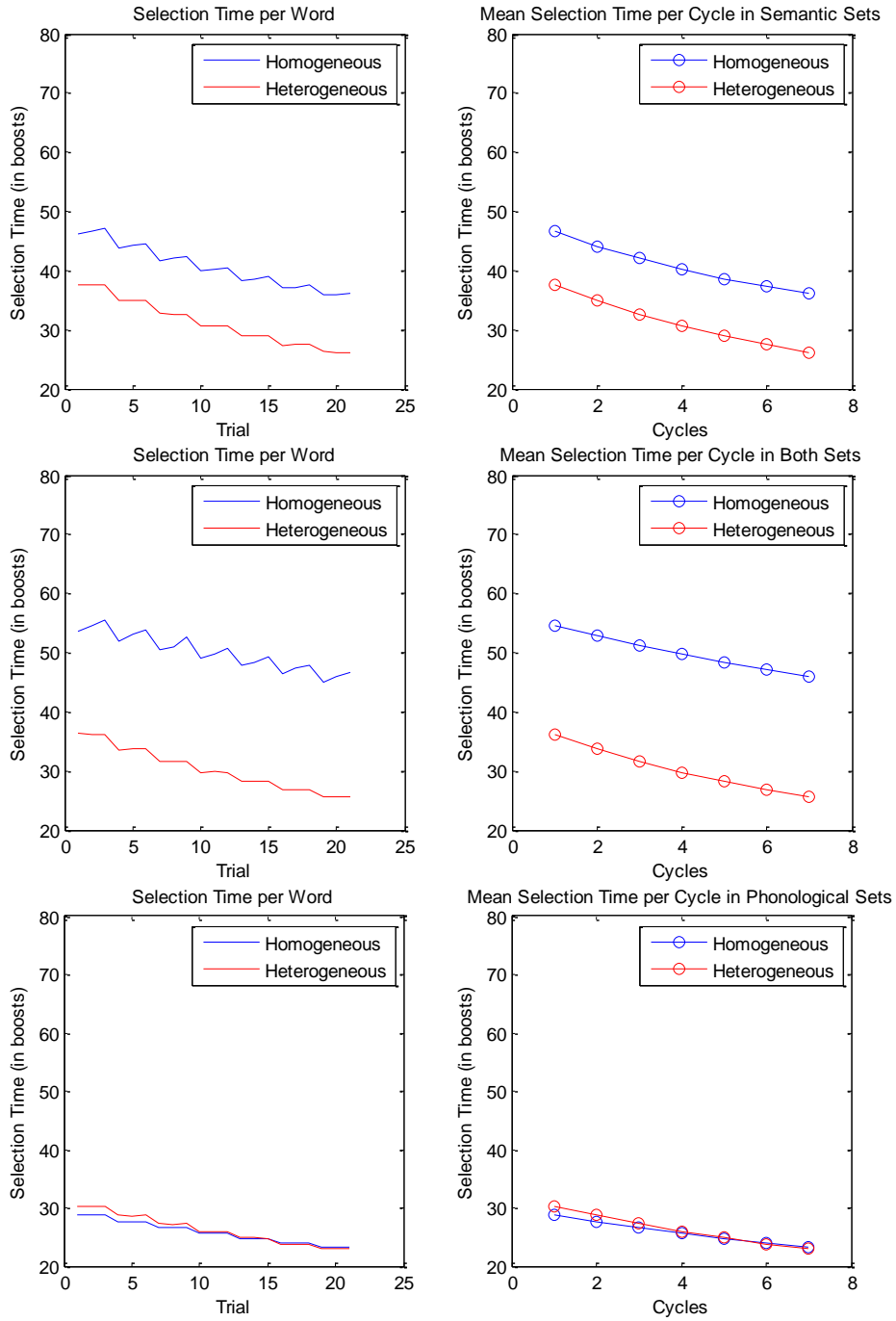
**Test 1.3** – *semantic* condition first followed by both followed by phonological condition. Heterogeneous sets DO NOT overlap words with homogeneous sets.



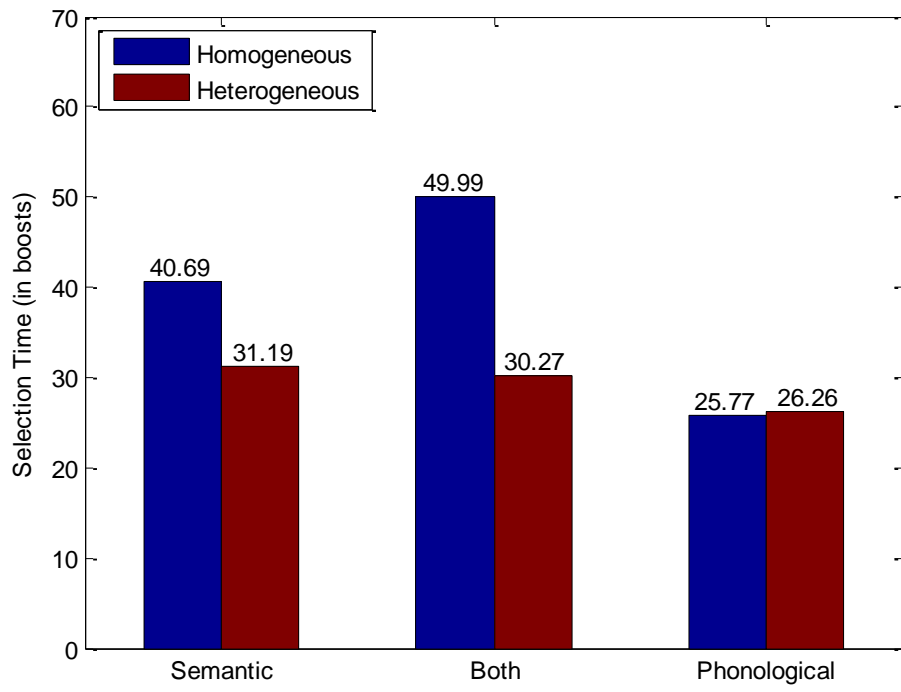
Picture Naming Latencies  
Phi = 0.900



**Test 1.4** – both condition first followed by semantic followed by phonological condition. Heterogeneous sets DO NOT overlap words with homogeneous sets.



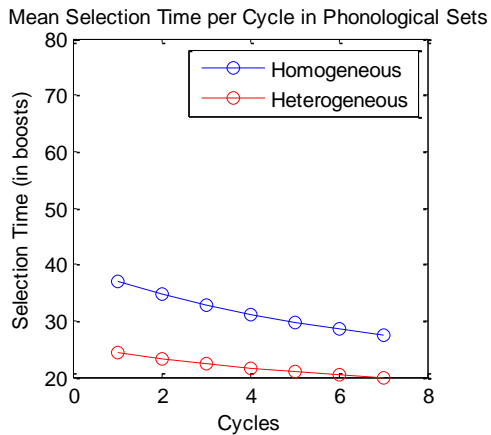
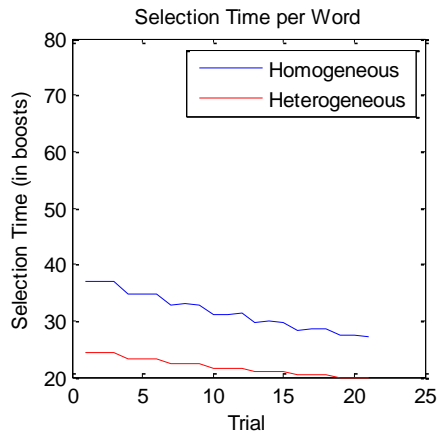
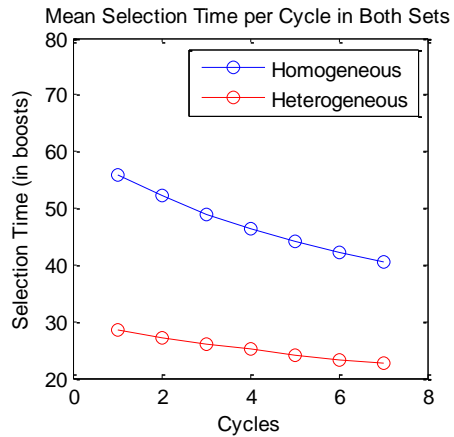
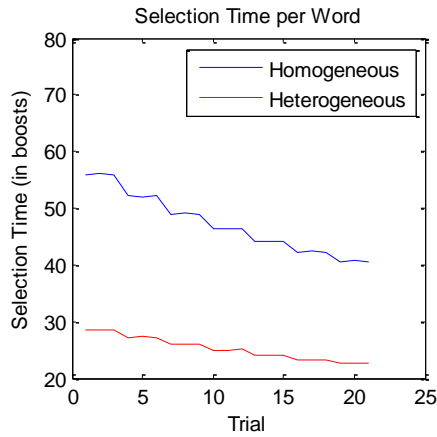
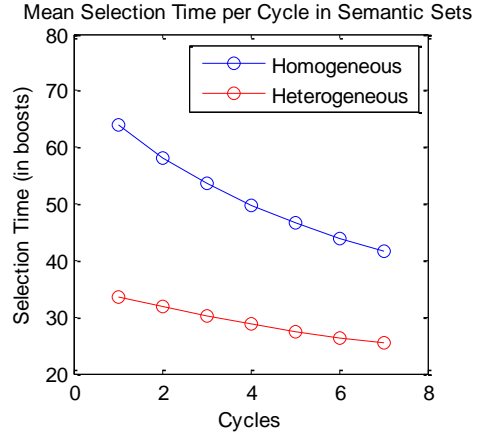
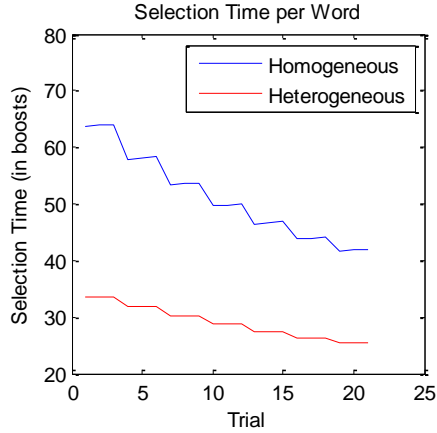
Picture Naming Latencies  
Phi = 0.900



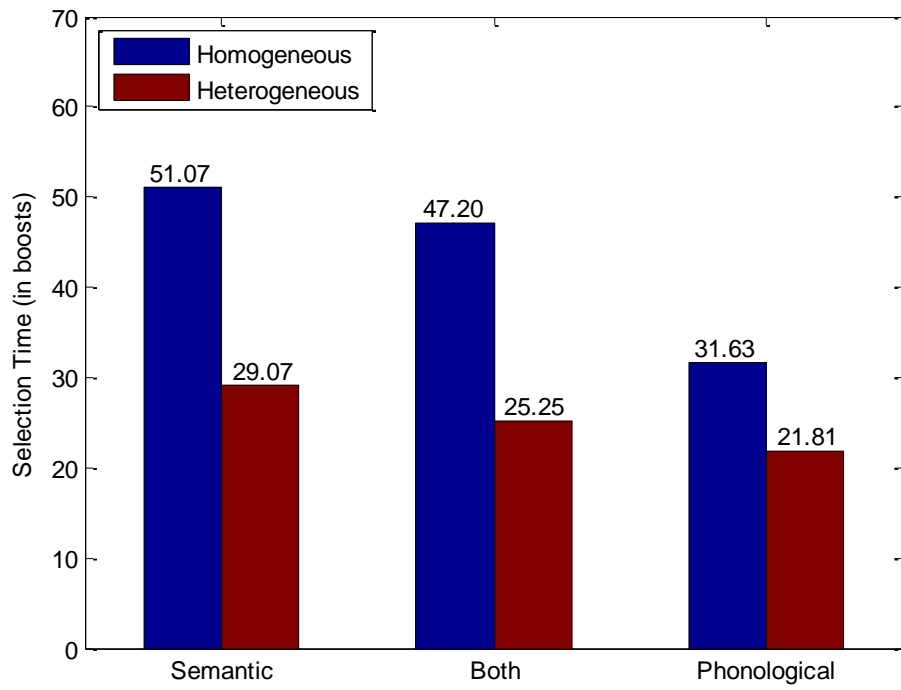


## Simulation 2 Test Plots

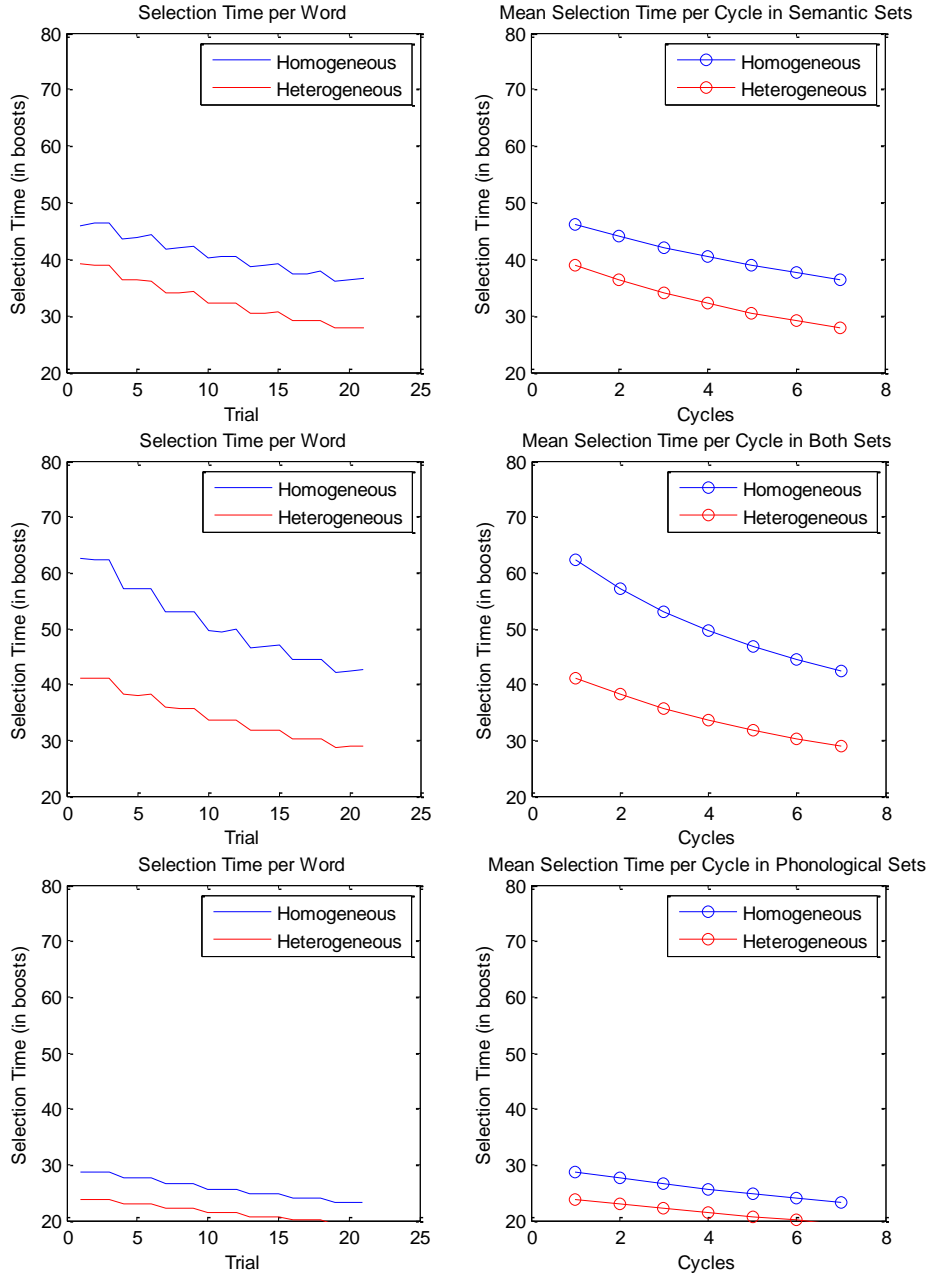
**Test 2.2** – *both* condition first followed by semantic followed by phonological condition. Heterogeneous sets overlap words with homogeneous sets.



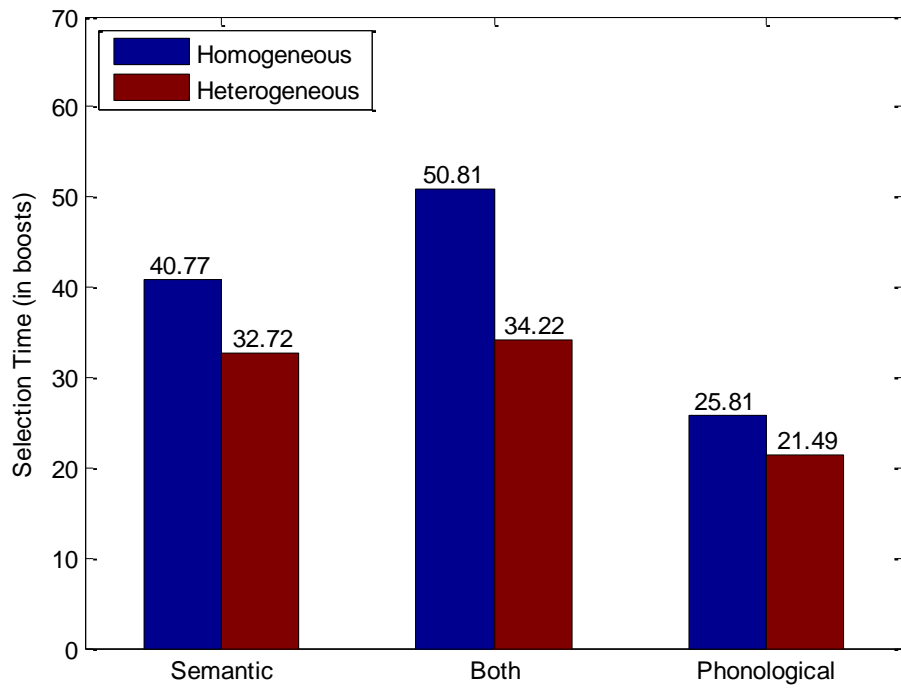
Picture Naming Latencies  
Phi = 0.900



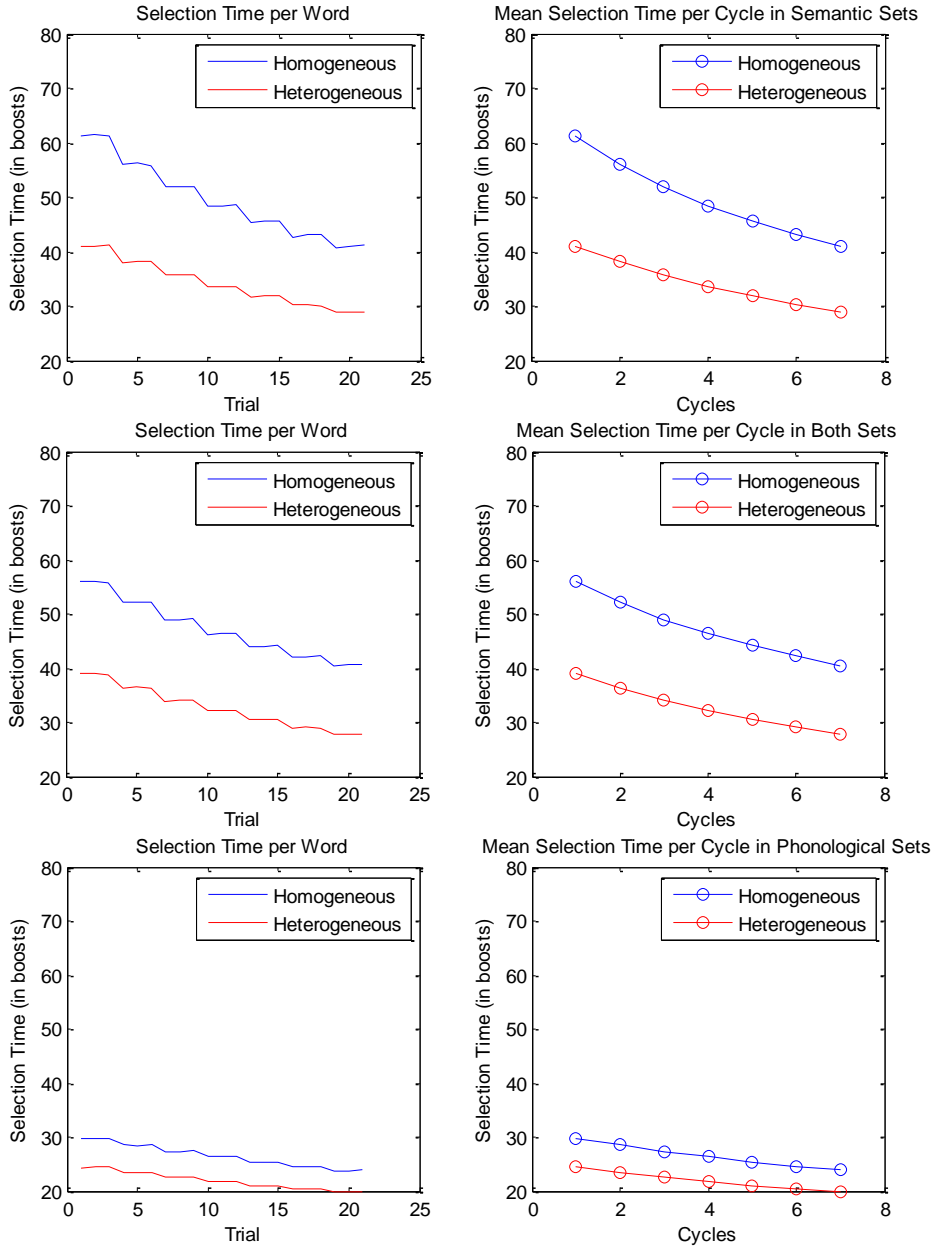
**Test 2.3** – *semantic* condition first followed by both followed by phonological condition. Heterogeneous sets DO NOT overlap words with homogeneous sets.



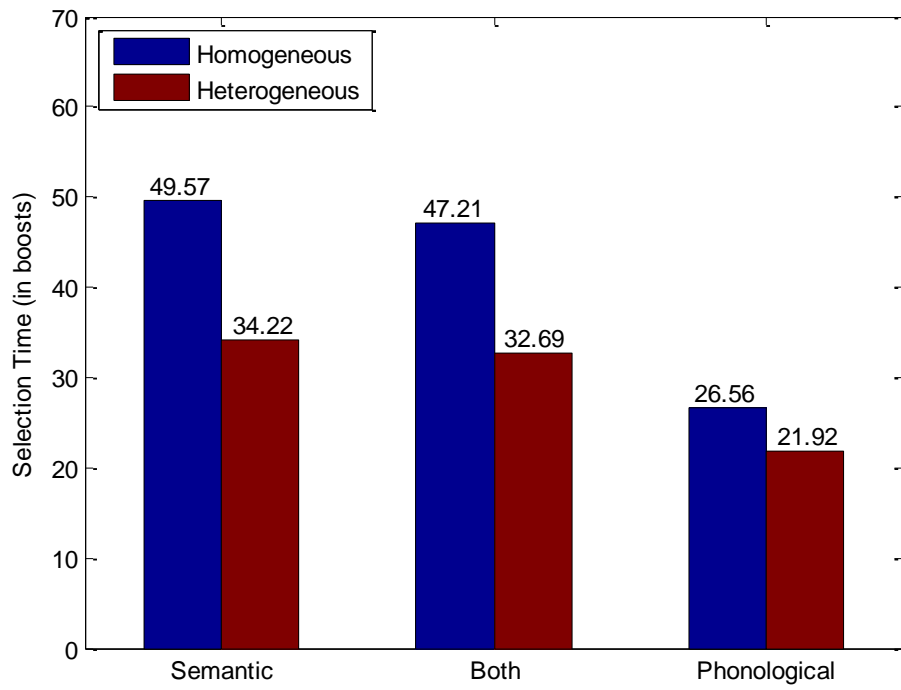
Picture Naming Latencies  
Phi = 0.900



**Test 4.4** – *both* condition first followed by semantic followed by phonological condition. Heterogeneous sets DO NOT overlap words with homogeneous sets.



Picture Naming Latencies  
Phi = 0.900



# Vita

## PERSONAL INFORMATION

Date of Birth: December 6, 1988  
Place of Birth: Bethlehem, PA, USA  
Father: Miltiadis Hatalis  
Mother: Stella Hatalis

## EDUCATION

M.S. Computer Science, Expected Fall 2013  
Lehigh University, Bethlehem, PA  
Advisor: Prof. Hector Munoz-Avila

B.S. Computer Science and Business, May 2012  
B.S. Computer Engineering, May 2011  
Lehigh University, Bethlehem, PA  
Advisors: Prof. Edwin Kay, Prof Sharon Kalafut

## WORK AND RESEARCH EXPERIENCE

Research Assistant, Intelligent Systems and Technologies Lab, Lehigh University

- Computational Neuroscience (September 2012 – Spring 2013): Researched the use of neural networks and other machine learning methods to model and analyze the cognitive theories of lexical priming and semantic interference in the brain. Part of this work resulted into simulations used for my Master's Thesis.
- Structural Health Monitoring (May 2012 – July 2012): Hierarchical reinforcement learning and statistical methods were applied to simulated sensor data of metal beams to improve damage detection and identify faulty sensors. This project was co-advised with Prof. Shamim Pakzad of the CEE dept.
- Machine Learning and Finance (January 2012 – Spring 2013): This was a side project that leveraged my background in business to test scalable machine learning algorithms. Reinforcement learning, ANN's, case based reasoning, and other methods are experimented with to optimize portfolio management and model derivatives markets.

Intern, IT Dept., Air Products and Chemicals

- April 2011 – December 2011: I worked on a team project to analyze hardware and develop software to create a kit of collaborative tools to enhance productivity and communication of on and off campus workers. Skills emphasized in this project were project management, team leadership, object oriented programming, and internal control assessment.

Senior Project, VADER Lab, Lehigh University

- September 2010 – April 2011: As a requirement for my computer engineering degree, I did a yearlong project on designing a wired glove which is an apparatus to detect and measure movement of the hand and fingers in three dimensional space. My advisor was Prof. John Spletzer, and co-advisor was Prof. William Haller.

## **HONORS AND AWARDS**

- Awarded Sherman Fairchild Fellowship for Summer 2012
- Included on the Dean's list for Spring 2012
- Won Mobile Robotics Competition Spring 2010
- Awarded Honors IB Diploma Spring 2007
- Member of NSHSS since Spring 2005