Investigating the Structure of Semantic Memory

by

Kevin Durda

A Thesis
submitted to the Faculty of Graduate Studies and Research
through the Department of Mathematics and Statistics
in partial fulfillment of the requirements for
the degree of Master's of Science at the
University of Windsor

Windsor, Ontario, Canada

Investigating the Structure of Semantic Memory

by

Kevin Durda

APPROVED BY:

K. Taylor
Chemistry & Biochemistry

T. Traynor
Mathematics & Statistics

L. Buchanan, Co-Supervisor
Psychology

R. Caron, Co-Supervisor
Mathematics & Statistics

S. Ahmed, Chair of Defense
Mathematics & Statistics

January 25^{th}, 2006

Abstract

In this paper, a method to extract semantic associations between words from a large corpus of text is presented. These associations are then used to construct groups of associated words, called semantic neighbourhoods, for each word. In addition, a way to use these semantic neighbourhoods, together with a graph-theoretic clustering algorithm, to compute a measure of ambiguity for words with more than one meaning is described.

Acknowledgments

This research was possible by infrastructure support from the Canadian Foundation for Innovation (CFI) and the Ontario Innovation Trust (OIT), and through Natural Sciences and Engineering Research Council (NSERC) and Social Sciences and Humanities Research Council (SSHRC) operating funds, as well as a Shared Heirarchical Academic Research Computing Network (SHARCNET) graduate student award.

Table of Contents

List of Figures

List of Tables

1 Introduction

In a lexical decision (LD) experiment a string of characters is displayed on a computer screen and the subject is asked to determine as quickly as possible, and without sacrificing accuracy, whether or not the text is an English word. The amount of time which elapses between the appearance of the string and the subject's response is called the reaction time (RT) and is recorded, together with the accuracy of the response. Many characteristics of a word have been found to affect a subject's RT in a LD task, including the orthographic (i.e., visual; Coltheart, Davelaar, Jonasson, & Besner, 1977; Andrews, 1992; Grainger & Jacobs, 1996; Peereman & Content, 1995; Sears, Hino, & Lupker, 1995) and phonological (i.e., aural; Westbury, Buchanan, & Brown, 2002; McClelland & Elman, 1986; Marslen-Wilson, 1987; Luce, Pisoni, & Goldinger, 1990; Peereman & Content, 1997) properties of the word, the word's written frequency (Kucera & Francis, 1967), as well as attributes related to the meaning of the word (Meyer, Schvandeveldt, & Ruddy, 1975; Buchanan, Westbury, & Burgess, 2001; Azuma & Van Orden, 1997; Hino & Lupker, 1996; J. M. Rodd, 2004). Cognitive psychologists have developed several models that attempt to explain how we perform word recognition. (Coltheart, Rastle, Perry, & Zeigler, 2001; Ratcliff, Gomez, & McKoo, 2004; Seidenberg & McClelland, 1989; McClelland & Rumelhart, 1981; Rumelhart & McClelland, 1982).

A model of word recognition consists of a specification of how knowledge about words is stored in the mind and a description of the processes that are used to retrieve, manipulate, and make decisions based on this knowledge. The validity of these models can be evaluated by comparing the performance of a computer implementation of the model to human performance across a battery of cognitive tasks. If the model cannot perform with speed and accuracy similar to that seen in humans, we may conclude either that this theory, in its current form, is not a satisfactory explanation of how humans recognize words, or that the computational model has been poorly implemented.

Unfortunately, computer implementations of these models have been incomplete.

While each of these models describes a semantic processing unit, many implementations have failed to include this unit. One possible reason for this is the lack of a reliable computer-based representation of the semantic content of a word. Several measures have been developed to quantify the visual and aural qualities of a word by comparing orthographic and phonological similarity between words (see Buchanan & Westbury, 2000, for a comprehensive listing). These are finite domains that are consistent between individuals. One cannot argue against the facts that the word *cat* is spelled *C-A-T*, that the spelling of *bat* is different from *cat* by only a single letter in the first position, and that both words are three letters in length. The situation is much the same for phonology (although we have all heard that *tomato* has more than one pronunciation). Comparisons between these features of a word are concrete and objective.

Judgments of semantic similarity, on the other hand, are subjective and can vary greatly from person to person. A baseball player will most likely associate the word *bat* with a wooden stick used to hit a ball, while a spelunker will probably associate this word with a flying mammal that hangs upside down in caves. These two individuals have very different semantic memories, but we hope that these memories have a common underlying structure and were formed by the same means (Buchanan et al., 2001).

The preceding example uncovers another difficulty in analyzing semantic relationships: many English words are ambiguous, that is, they have more than one meaning. In an analysis of 4,930 words in the Wordsmyth dictionary (Wordsmyth, 1999) having a word-form frequency greater than ten per million words in the CELEX lexical database (Baayer, Piepenbrock, & Van Rijn, 1993), J. M. Rodd, Gaskell, and Marslen-Wilson (2004) reported that 7.4% of the words have more than one distinct meaning. For example, *bank* may refer to a financial institution or the land at the edge of a river. 84% of these words have more than one variation of the same meaning. *Paper*, for example, may refer to a material made from pressed wood pulp, or a single, standard-size sheet of this material. These subtle differences in meaning are called *senses*, and 37% of the words analyzed have more than five senses. Thus, a

psychologically relevant representation of semantic content must be able to account for multiple meanings and senses of a word.

In this thesis I present a new method for determining semantic associations between words, and with these associations, vector-based representations of the semantic content of each word, and their "semantic neighbourhoods" (SN; a group of words which are strongly associated or semantically similar) are created. Using a graph-theoretic clustering algorithm, the SNs are separated into several groups that are hypothesized to contain only words related to the target by a particular meaning. By examining the written frequency of the words in these groups, the probability of the target word appearing in each of its available contexts is estimated. These data are then used to calculate a measure of the amount of uncertainty inherent in each word's meaning.

The next chapter contains a brief overview of this new method. Chapter 3 gives a more in-depth discussion of lexical ambiguity. Chapter 4 describes the two previously developed methods of creating semantic representations that form the foundation of the current method, and Chapter 5 describes the graph clustering algorithm used as a tool to measure ambiguity. Chapter 6 presents a mathematical framework in which relationships between words can be determined, and provides a technical description of the new algorithm. Results are presented in Chapter 7 and possible future directions for this work are discussed in Chapter 8.

2 A Brief Overview

The following provides a quick outline of a new procedure for extracting semantic associations between words by analyzing lexical co-occurrence in a large corpus, how a representation of the semantic characteristics of a word can be created based on these associations, and how these associations are used, together with a graph clustering algorithm, to measure the ambiguity of a word. A mathematically rigorous description of this process is given in Chapter 6.

The first goal is to measure the strength of the relationship between every pair

of words. This strength is called *semantic association*. Words that are strongly associated (e.g., *umbrella* and *rain*), or semantically similar (e.g., *coffee* and *tea*), have a higher semantic association than words that are only weakly related (e.g., *coffee* and *umbrella*). These values are calculated by analyzing the number of times a pair of words occur together in written text, referred to as *lexical co-occurrence*.

The method begins calculating semantic associations by passing a small window over each word in the corpus. The word currently under inspection is called the *target word*, or the *target*. This window contains a user defined number of words preceding and following the target. For each word, the number of times that every other word appears in each window position, as well as the total number of times the word appears in the corpus, is counted. This last number is adjusted to give the written frequency per million words of text, called the *orthographic frequency* of the word. Once this step is completed, the lexical co-occurrence counts are adjusted to more accurately measure the importance of this co-occurrence. Some words (such as *the*) have a very high orthographic frequency and will occur near every word a disproportionate number of times. Because the semantic associations are calculated based on lexical co-occurrence, they are subject to influence from orthographic frequency. To counteract this effect, co-occurrence counts with high frequency words are reduced. The details of how this is done are given in Section 6.2. Next, a weight is assigned to each window position. Weights are assigned in a manner that allows the algorithm to optimally calculate word ambiguity. This task requires the background provided in Chapter 5 and is described in Section 6.5. In this brief introduction, it is assumed that these weights have already been found. The semantic association between two words is the weighted sum of the adjusted co-occurrence counts across all window positions. If two words never occur together in a window, the semantic association between those words is set to zero.

These associations are then used to create a vector representing the semantic characteristics of a word. For a given target word, the semantic association between the target and each unique word in the corpus (including the target itself) is calculated. These values are then sorted alphabetically according to the word they correspond

to, and this ordered list is used as the semantic representation of the target.

Finally, the semantic associations are used to measure the ambiguity of each word. A graph containing only the words most strongly related to a target is constructed and the algorithm described in Section 5.3 is applied to find groups of highly interconnected words within this graph. Each group, or cluster, should contain only words that are related to the target by a particular meaning. For example, the graph for *bank* should contain one cluster of words related to the financial institution meaning, one cluster containing words related to the river bank meaning, as well as other clusters corresponding to the other meanings of *bank*. Using the orthographic frequencies of the words in each cluster, the proportion of occurrences of the target that are associated with each meaning is estimated. These values are then used to find the information entropy (Shannon, 1948), given by

$$U = -\sum_{i=1}^{n} p_i \log p_i,$$

where n is the number of meanings of the word and p_i is the proportion of occurrences of the target word that are related to meaning i. The entropy of a word's meaning is an established measure of semantic ambiguity (Twilley, Dixon, Taylor, & Clark, 1994).

3 Homographs and Polysemous Words

As discussed in Chapter 1, many English words have more than one meaning. Some words (e.g., *bank*) have multiple distinct meanings, and are referred to as *ambiguous*. Another class of ambiguous words have many subtle variations centred around a core meaning (e.g., *paper*). These variations in meaning are called *senses*, and these words are called *polysemous*.

Several studies have shown that ambiguity and polysemy affect RT in LD and word naming[1]. Early studies suggested that a high level of ambiguity in a word's

[1] In a word naming task, a string of letters is presented on the a computer screen and the subject is asked to say the word aloud.

meaning offered an advantage in these tasks (Azuma & Van Orden, 1997; Hino & Lupker, 1996; Lichacz, Herdman, Lefevre, & Baird, 1999; Pexman & Lupker, 1999). J. Rodd, Gaskell, and Marslen-Wilson (2002) suggested that these studies may have confounded ambiguity and polysemy, and showed that words with multiple distinct meanings (e.g., *bank*) are recognized slower than non-ambiguous words, while words with multiple variations of a single meaning (e.g., *paper*), are recognized faster. A similar result was also found by Klepousniotou (2002).

Twilley et al. (1994) created relative meaning frequency norms for 566 homographs by providing lists of words to subjects and asking them to write down the first word that came to mind. The responses were then grouped into meaning categories and the proportion of responses corresponding to each category was calculated as a measure of relative meaning frequency. For each word, a measure of ambiguity was calculated by using the entropy formula from information theory (Shannon, 1948). For a word with n meanings, each with a corresponding proportion $p_i, i = 1, \ldots, n$, of the responses, the ambiguity measure can be found using

$$U = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{1}$$

U is a measure of the degree of randomness inherent in an event. Higher U values correspond to more ambiguous words and the maximum value of U increases with the number of meanings. For a word with n meanings, the range of U is $0 \leq U \leq log_2 n$. $U = 0$ when only one meaning occurs, and the maximum value occurs when all meanings are equally likely.

Additionally, a balance value, B, was calculated for each word. This used the same formula as U, but included only the two most frequent meanings, with their proportions adjusted to total to 1. B values range from 0 to 1. This is a measure of the degree of dominance of the first meaning over the second meaning. A B value of 0 indicates that the relative frequency of the second meaning is 0, and a B value of 1 indicates that the first and second meanings are perfectly balanced. Any word with a B value greater than 0.95 is considered a balanced homograph, and those with a B value between 0.1 and 0.3 are considered to be polarized.

As an example, consider the word *ball*. As measured by Twilley et al. (1994), the proportion of responses related to the most frequent meaning, playing sports, is 0.92, and the proportion of responses related to dancing, the second most frequent meaning, is 0.02. Adjusting these proportions to total to 1, the proportion related to sports is .979 and the proportion related to dancing is 0.021. Using these values in the equation

$$B = -p_1 \log_2 p_1 - p_2 \log_2 p_2,$$

which is simply Equation 1 with $n = 2$, the balance measure for this word is $B = 0.147$. This indicates that *ball* is a polarized homograph, or that one meaning (in this case the object used in sports) dominates the second most common meaning. To further illustrate this idea, consider the word *park*. The proportion of responses related to cars is 0.45, while the proportion of responses related to city parks is 0.42. These values give a balance measure of $B = 0.999$. Thus, *park* is a balanced homograph.

Shannon (1948) defines the entropy of a data source as a measure of the amount of choice, or uncertainty, involved in the selection of one of n possible outcomes for an event, with each outcome having a probability of $p_i, i = 1, \ldots, n$. Let $H(p_1, \ldots, p_n)$ be such a measure. H must meet the following three requirements:

1. H should be a continuous function of the p_i's.

2. If all of the outcomes are equally likely, then H should be an increasing function of n, since a higher number of equally likely outcomes produces more uncertainty.

3. The following identity should hold:

$$H(p_1, p_2, p_3) = H(p_1, p_2 + p_3) + (p_2 + p_3) H\left(\frac{p_2}{p_2 + p_3}, \frac{p_3}{p_2 + p_3}\right). \qquad (2)$$

   This condition states that if a choice is broken down into two successive choices, the value of H should not change. In Figure 1, the probability tree on the right shows an event with three possible outcomes. The tree on the left shows the same choice broken down into two steps. The probability of each of the three

outcomes is the same in both trees. Equation 2 states that the entropy should be the same in both of these situations.

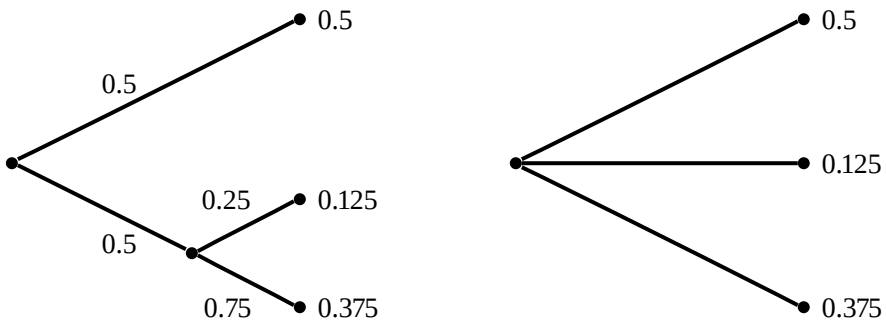


Figure 1: Two decompositions of a three choice decision.

In Appendix 2 of Shannon (1948), it is shown that

$$H(p_1, \ldots, p_n) = -K \sum_{i=1}^{n} p_i \log p_i, \tag{3}$$

where K is a positive constant, is the only H satisfy these three conditions. The value of K amounts to a choice of the base of the logarithm.

Alternatively, consider a word with n meanings, each of which is equally likely. Let A_i be the number of time we encounter meaning i of the word in a total of P occurrences of the word, where $P = \sum_{i=1}^{n} A_i$. The probability of obtaining the distribution $(A_1, A_2, \ldots, A_n)$ is

$$p = \frac{\Omega}{T}$$

where

$$\Omega = \frac{P!}{A_1! A_2! \ldots A_n!}$$

and $T = n^P$. The entropy of this distribution is obtained from $H = \log \Omega$ (see Wikipedia, 2005, for a derivation) and is given by Equation 3.

This function possesses several properties that make it particularly well suited as a measure of ambiguity:

1. $H = 0$ if and only if there exists i' such that $p_{i'} = 1$ and $p_i = 0$, for all $i = 1, \ldots, n, i \neq i'$. Put into words, this property becomes obvious: there is no uncertainty only when we are certain what the outcome will be.

2. H attains its maximal value of $\log n$ only when all outcomes are equally likely.

3. If the p_i are adjusted to be more nearly equal, the value of H increases.

Together, properties 2 and 3 listed above describe the following important property of H: the outcome of an event is less certain if all outcomes are nearly equiprobable than if a single outcome is much more likely than all others.

One goal of the present work is to extract a measure of word ambiguity. Using a graph-theoretic clustering technique, discussed in Section 5.3, the procedure used by Twilley et al. (1994) will be automated and used to obtain an ambiguity and balance measure for each unique word in a corpus.

4 Models of Semantic Memory

Semantic memory refers to our knowledge of words, their meanings, and their relationships to each other and to the physical world. It may be thought of as a dictionary, encyclopedia, and thesaurus, all rolled into one (Tulving, 1972). A model of semantic memory refers to a description of how the semantic features of a word are represented, how these representations can be combined into larger units of meaning (such as phrases and sentences), what deductions can be made about a word based on the context in which it appears, and how word meaning is related to the perceptual systems that provide access to the world (McNamara & Holbrook, 2003).

In this chapter, two broad classifications of semantic memory models are introduced and two techniques for constructing semantic representations based on word co-occurrence in a large corpus are discussed. Note that these co-occurrence techniques deal only with the first goal of a semantic memory model as described above, that is, specifying a representation of a word's meaning. Thus, these methods of constructing representations do not form complete models of semantic memory, but for the sake of clarity they will be referred to as models in this thesis.

In an object-based view of semantics, words are considered to be associated if the objects they refer to have shared properties. Closeness is a measure of the similarity

between objects. The property investigated may be common features or inclusion in a common category. Category membership can be treated as a strongly weighted feature, making these two ways of classifying words essentially identical. As a quick example, consider the words *cat* and *dog*. Both have teeth, claws, fur and a tail. Additionally, both can be included in the category *pets*. In an object-based view of semantics, the word *cat* would be closely semantically related to *dog*.

There are numerous models of object-based semantics (McRae, de Sa, & Seidenberg, 1997; Collins & Loftus, 1975; Collins & Quillian, 1969), but the focus of this thesis is on the second type of model.

4.1 Language-Based Semantics

Language-based models determine semantic associations between words by analyzing lexical co-occurrence in a large corpus of written text. Considerable evidence suggests that representations reflecting a language model are more consistent with the organization of our own semantic memory than those reflecting objects (Buchanan, Brown, Cabeza, & Maitson, 1999). In this section two language-based models of semantic memory are described, upon which this new method of constructing semantic representations is based. Note that this list is by no means exhaustive, and that many other language-based models exist (see Landauer & Dumais, 1997; Lemaire & Denhière, 2004; Nelson, McEvoy, & Schreiber, 1998, for example).

4.1.1 Hyperspace Analogue to Language (HAL)

HAL (Lund & Burgess, 1996; Burgess, 1998; Burgess & Livesay, 1998; Burgess, Livesay, & Lund, 1998; Burgess & Lund, 1997) is a computational model that uses vectors to represent entries in semantic memory. A large corpus of written text, consisting of approximately 160 million words collected from Usenet newsgroups, was analyzed by passing a small window over each word in this text. Each position of this window is assigned a weight. The window position closest to the target is assigned a weight equal to the size of the window and these weights decrease linearly as distance

from the target increases, with the farthest window position receiving a weight of one. For example, if the window extends five words in front of the target, the closest position receives a weight of 5, the next position receives a weight of 4, and so on, until the fifth position is assigned a weight of 1. Weighted word co-occurrences were recorded in a matrix containing one row and one column for each unique word in the corpus. A co-occurrence vector was constructed for each word by concatenating the transpose of the row corresponding to the target word to the word's corresponding column. These vectors were then normalized to a constant length and a measure of semantic similarity between words was calculated by using the Minkowski family of distance metrics,

$$d_r(\vec{w}_i, \vec{w}_j) = \left(\sum_{k=1}^{\rho} \left| w_i^k - w_j^k \right|^r \right)^{\frac{1}{r}},$$

where $\vec{w}_i = \langle w_i^1, w_i^2, \ldots, w_i^\rho \rangle$ and $\vec{w}_j = \langle w_j^1, w_j^2, \ldots, w_j^\rho \rangle$ are the vectors corresponding to the two words under consideration. This family of metrics includes both Euclidean distance, when $r = 2$, and rectilinear distance, when $r = 1$. If $d_r(\vec{w}_i, \vec{w}_j) < d_r(\vec{w}_i, \vec{w}_k)$, then w_i is said to be more strongly related (or closer) to w_j than to w_k.

In HAL, words with similar meaning (e.g., *street* and *road*), or words that are strongly related (e.g., *street* and *car*), exist closer together in semantic space than unrelated concepts. In addition, multidimensional scaling (Kruskal, 1978) reveals that these vectors contain some notion of categorical information. Exemplars from a common category (e.g., *apple* and *orange*) are closer together than exemplars from different categories (e.g., *apple* and *wrench*). The distance between words correlated with the priming advantage seen in a semantic priming task: words that are closer together based on HAL's distance measure produced a larger priming effect (e.g., *apple* primes *orange* more than it primes *umbrella*).

4.1.2 Computational Analysis of Text: Semantic Co-occurrence Association Norms (CATSCAN)

CATSCAN (Casey, 2005; Durda, Casey, Buchanan, & Caron, under review) is a language-based model developed to remedy two flaws inherent in HAL: CATSCAN's

vector representations contain less influence from the written frequency of words, and window position weights are assigned in a non-arbitrary fashion.

The most prominent difference between CATSCAN and HAL is a reduced sensitivity to orthographic frequency. Consider extremely high frequency words, such as *the*, *and*, and *a*. Since these words occur so frequently in written English, they will have a high number of co-occurrences with every word. In an attempt to overcome this problem, CATSCAN uses an adjustment factor to reduce the semantic associations between high frequency words. Let

$$\lambda = e^{-\frac{f_t + f_a}{c}},$$

where f_t and f_a are the orthographic frequency of the target word and potential associate, respectively. The value of c is chosen so that 99% of the words in the corpus have a frequency less than c. λ is multiplied by the semantic association between the target and the associate to reduce association between high frequency words.

CATSCAN also differs from HAL in the method used to assign a weight to each window position. In HAL, the window position closest to the target was assigned the highest weight, and weights decrease linearly as distance from the target increases. Assigning weights in this way implies that the words immediately preceding and following the target are the most relevant in determining semantic relationships, which may not necessarily be the case. Nouns, for example, are often preceded by one of the words *the*, *a*, or *an*. These words have little semantic value and should not be considered the most important in determining semantic associations.

In CATSCAN, window position weights are not determined until all co-occurrence data has been collected from the corpus. Instead of arbitrarily assigning linearly ramped weights, CATSCAN uses weights such that the magnitude of the correlation between the resulting semantic associations and RT in a LD experiment is maximized. It is interesting to note that, when weights were assigned in this manner, with the additional constraint of non-negativity, only the first and third closest window positions on either side of the target and the eleventh window position before the target were

assigned non-zero weights. Refer to Casey (2005) for a more complete description of the methods used to assign window weights in CATSCAN.

This model produces two types of SNs: A "local" SN, which measures the degree to which words are used together, and a "global" SN, which measures the degree to which two words are used in the same context. As a quick example, consider the word *bank*. The words *account* and *water* are possible local neighbours of *bank* corresponding to the financial institute and the river bank meanings, respectively. The word *shore* is a possible global neighbour. Stated more simply, a global neighbour will be a word that is synonymous with the target word, and is therefore closely related, but perhaps never appears with the target.

5 Graph Theory and Graph Clustering

In this section, some basic definitions from graph theory and the graph-theoretic clustering algorithm that is used in determining each word's ambiguity are presented. Section 5.1 introduces the concept of a graph and several related definitions. A method of constructing a graph based on semantic associations is discussed in Section 5.2, and the clustering algorithm is discussed in Section 5.3.

5.1 Graph Theory

This section introduces the terminology used to discuss graph theory. For more in-depth coverage of this material, refer to any of the several texts available on the subject (for example, Gibbons, 1985; Chartrand, 1985; Diestel, 2000)

Definition 5.1 Let S be a finite, non-empty set. The *unordered product of S with itself* is

$$S \otimes S = \big\{\{s_i, s_j\} \colon s_i, s_j \in S\big\}.$$

Definition 5.2 A *graph*, denoted $G = (V, E)$, is a finite, nonempty set V, together with a finite, possibly empty set $E \subseteq V \otimes V$. The elements of V are called *vertices*, and the elements of E are called *edges*.

As an example, consider the graph with $V = \{1, 2, 3, 4, 5\}$ and

$$E = \big\{\{1,2\}, \{1,5\}, \{2,3\}, \{2,5\}, \{3,3\}, \{3,4\}, \{3,5\}, \{4,4\}, \{4,5\}\big\}.$$

A graph can be represented in $\mathbb{R}^2$ by plotting the vertices as points and drawing a line between the endpoints of each edge. The example graph, depicted in this way, is shown in Figure 2.

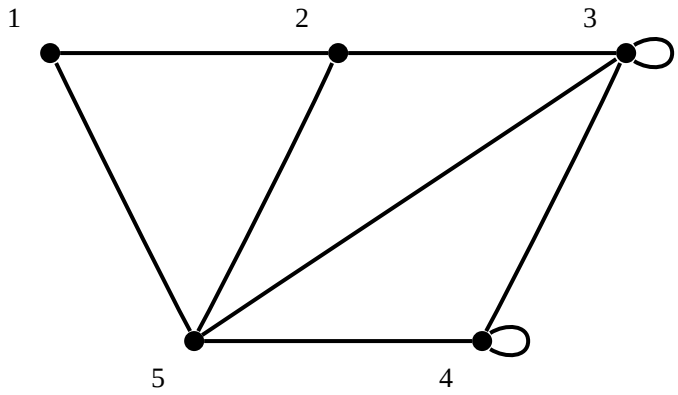


Figure 2: A representation of a graph in $\mathbb{R}^2$.

Definition 5.3 Let $G = (V, E)$ be a graph. The *order* of G is the number of vertices in V, denoted $|V|$. The *size* of G is the number of edges in E, denoted $|E|$.

The order of the example graph is $|V| = 5$, and its size is $|E| = 9$.

Definition 5.4 Let $e = \{v_i, v_j\}$ be an edge in a graph, $G = (V, E)$. Then e is *incident* with both v_i and v_j, and v_i and v_j are said to be *adjacent*, or *neighbours*.

Definition 5.5 Let $G = (V, E)$ be a graph and let $v \in V$. The *neighbourhood* of v, denoted $N(v)$, is the set of all vertices that are adjacent to v, given by

$$N(v) = \big\{w \in V : \{v, w\} \in E\big\}.$$

In the graph in Figure 2, vertices 1 and 2 are adjacent and the edge $\{1, 2\}$ is incident with both vertices 1 and 2. The neighbourhood of vertex 1 is $N(1) = \{2, 5\}$ and the neighbourhood of vertex 3 is $N(3) = \{2, 3, 4, 5\}$.

Definition 5.6 Let $e = \{v_i, v_j\}$ be an edge in a graph, $G = (V, E)$. If $v_i = v_j$, then e is called a *self-loop*, or simply a *loop*.

Definition 5.7 Let $G = (V, E)$ be a graph and let $v \in V$. The *degree of* v, denoted $\deg(v)$, is the number of edges in G that are incident with v, with loops counted twice.

In the example, $\deg(1) = 2, \deg(2) = 3$, and $\deg(3) = 5$.

Definition 5.8 The graph in which each pair of vertices are adjacent is called a *complete graph*. Alternatively, if $\deg(v) = |V| - 1$ for every $v \in V$, then G is a complete graph. The complete graph of order n is denoted by K_n.

The complete graphs of orders one through four are shown in Figure 3.

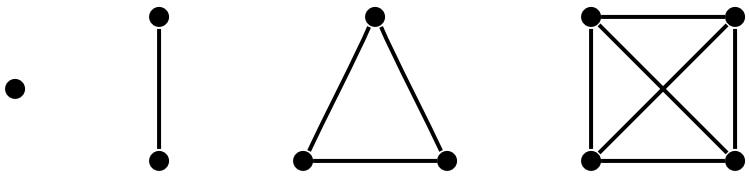


Figure 3: The complete graphs of orders 1 through 4.

Definition 5.9 Let $G = (V, E)$ be a graph. Let $V' \subseteq V$ and

$$E(V') = \big\{\{v_i, v_j\} \in E \colon v_i, v_j \in V'\big\}$$

be the set of all edges in G with both endpoints in V'. If $E' \subseteq E$, then the graph $G' = (V', E')$ is a *subgraph* of G, denoted $G' \sqsubseteq G$. If $E' = E(V')$, then G' is called the *subgraph of* G *induced by* V', denoted $G[V']$.

Let $V' = \{1, 3, 4, 5\}$. Then

$$E(V') = \big\{\{1, 5\}, \{3, 3\}, \{3, 4\}, \{3, 5\}, \{4, 4\}, \{4, 5\}\big\}.$$

Figure 4 shows a subgraph of our example with

$$E' = \big\{\{1, 5\}, \{3, 3\}, \{3, 4\}, \{4, 5\}\big\},$$

and $G[V']$ is shown in Figure 5.

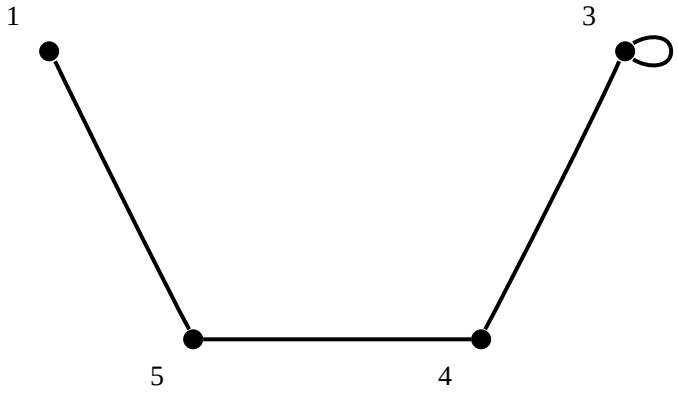


Figure 4: A subgraph.

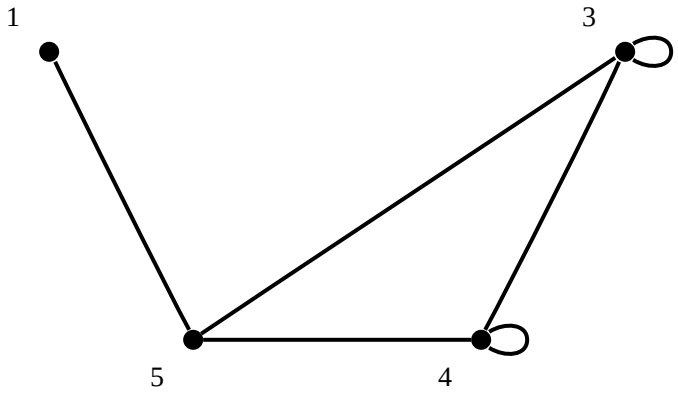


Figure 5: The subgraph induced by $V' = \{1, 3, 4, 5\}$

Definition 5.10 Let $G = (V, E)$ be a graph and let $P \subseteq V$ with the property that $P \otimes P \subseteq E$. Then $G[P]$ is a complete graph and is called a *clique* of G. If P has the further property that adding any $v \in V \setminus P$ breaks the condition that $P \otimes P \subseteq E$, then $G[P]$ is called a *maximal clique* of G. Let $w \in V$. If $G[P]$ is the largest clique containing w, then $G[P]$ is called a *major clique* of G.

Note that a maximal clique is not necessarily a major clique, but every major clique is a maximal clique. Also, every vertex in a graph is contained in at least one clique, since the subgraph induced by a single vertex is complete. To simplify notation, the set of vertices, P, is used to refer to the clique $G[P]$.

In the graph shown in Figure 2, the set of vertices $P = \{1, 2, 5\}$ forms a clique. The subgraph induced by P is shown in Figure 6. Note that this is the complete graph of order 3.

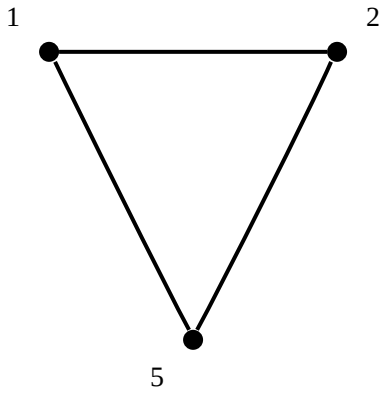


Figure 6: A clique of a graph.

As another example, consider the graph Figure 7. This graph contains a clique consisting of the five vertices displayed in white. Note that the subgraph induced by these vertices, shown in Figure 8, contains the complete graph of order five, that is, every pair of two distinct vertices are joined by an edge.

5.2 Creating the Graph

This section describes how to construct the graph used as input to the clustering algorithm presented in Section 5.3. In this graph, words are represented as vertices

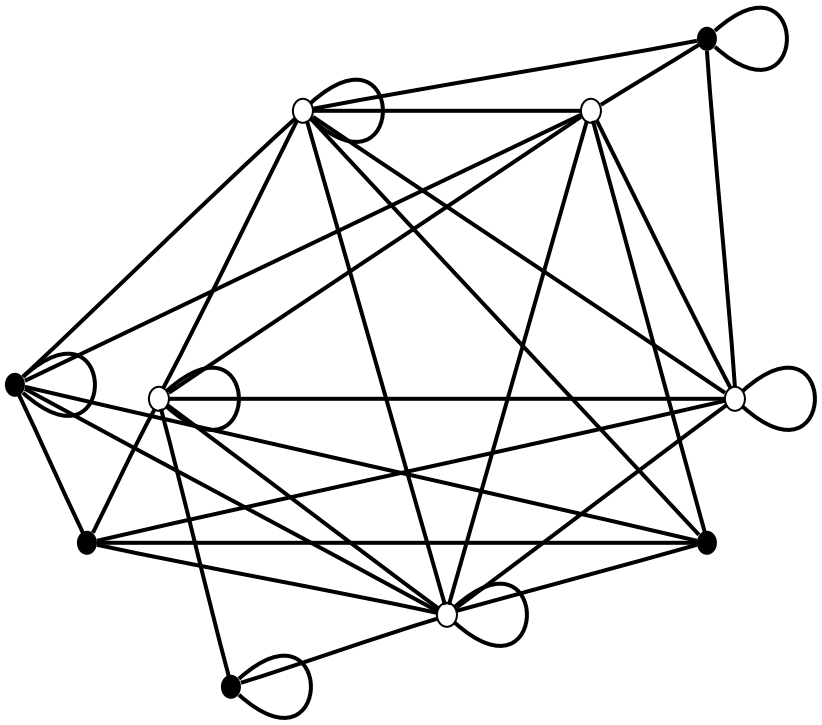


Figure 7: A graph containing a clique of five vertices.

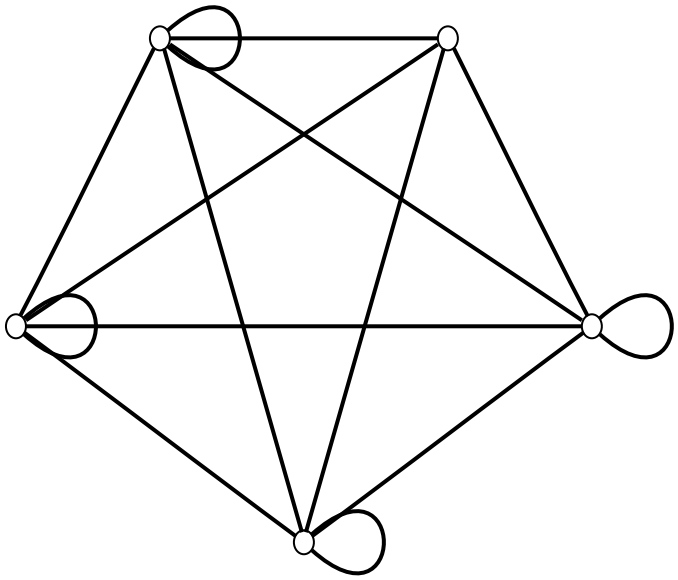


Figure 8: The subgraph induced by the clique in Figure 7.

and edges are placed between strongly associated words. The method described below was used by Aksoy and Haralick (1999a, 1999b) to construct the graph used in their image retrieval algorithm.

Suppose that the semantic association between each pair of words in the corpus have been calculated. A target word, w_0, is selected and a graph of this word's semantic neighbourhood is constructed. The N words with the strongest association to w_0 are found and put into a result set, $S_0 = \{w_{01}, w_{02}, \ldots, w_{0N}\}$. If w_0 did not occur with at least N distinct words, then the full list of semantic associates of w_0 is taken as the result set. For each word, $w_{0i} \in S_0, i = 1, \ldots, N$, the closest N neighbours are retrieved and put in a set, $S_i = \{w_{i1}, w_{i2}, \ldots, w_{iN}\}$, giving $N + 1$ sets containing up to N words each. Next, the set,

$$V = \{w_0\} \cup \left(\bigcup_{i=0}^{N} S_i \right),$$

containing all words that were used as query words, as well as all words that appeared in any result set, is formed. Note that $|V| \leq N^2 + N + 1$. V is the vertex set of the graph. The graph contains an edge from each word that was used as a query word to each word appearing in the result set of that query word. Keeping this in mind, the edge set, E, is described by

$$E = \big\{ \{w_i, w_{ij}\} \in V \otimes V : w_{ij} \in S_i, i = 0, \ldots, N, j = 1, \ldots, N \big\}.$$

By constructing the graph in this manner, *higher-order co-occurrences* are included in the SN of each word. Suppose that word w_i is a close semantic associate of w_j, and w_j is strongly related to w_k, but that w_i is not related to w_k (i.e., the semantic association between w_i and w_j is 0). By considering higher-order co-occurrences during the construction of w_i's SN, w_k may appear in the SN of w_i, even though the two words never occurred together in the corpus. For example, the word *street* may often occur with *car* and *car* may often occur with *road*, but *road* and *street* may occur together very infrequently because they are synonymous. However, this method of graph construction allows *road* to appear in the SN of *street*.

Table 1 contains the query and result set data used to create the graph shown in Figure 9. This graph will be used as an example to illustrate the clustering algorithm

Query	Result Set
1	1, 2, 3, 4, 8, 9
2	2, 1, 4, 5, 3, 6
3	3, 2, 1, 4, 6, 7
4	4, 1, 7, 3, 5, 6
8	8, 9, 1, 10, 11, 12
9	9, 1, 8, 10, 11, 12

Table 1: Queries and result sets for the graph in Figure 9

presented in the next section. Figure 10 shows a graph constructed for the word *bark*, using $N = 43$.

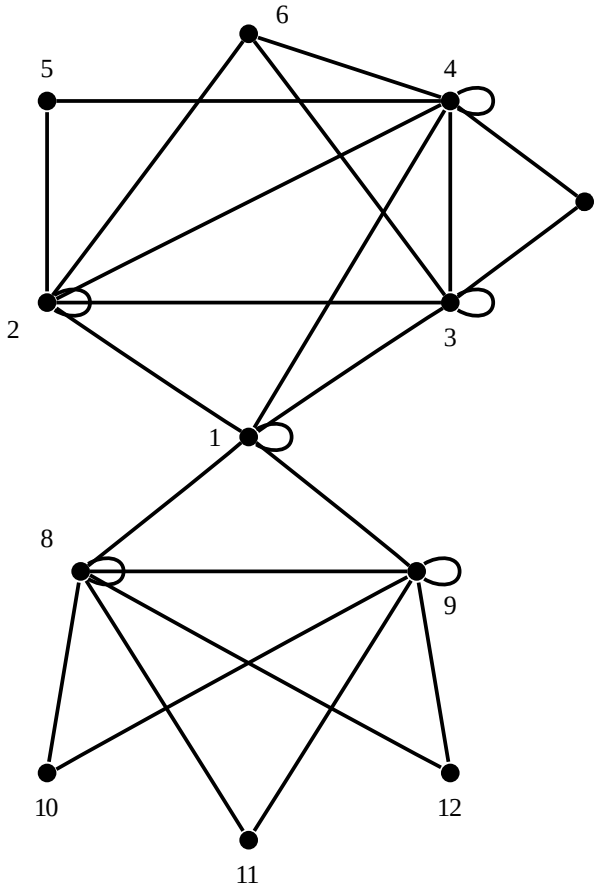


Figure 9: An example of a graph constructed using the method of Section 5.2.

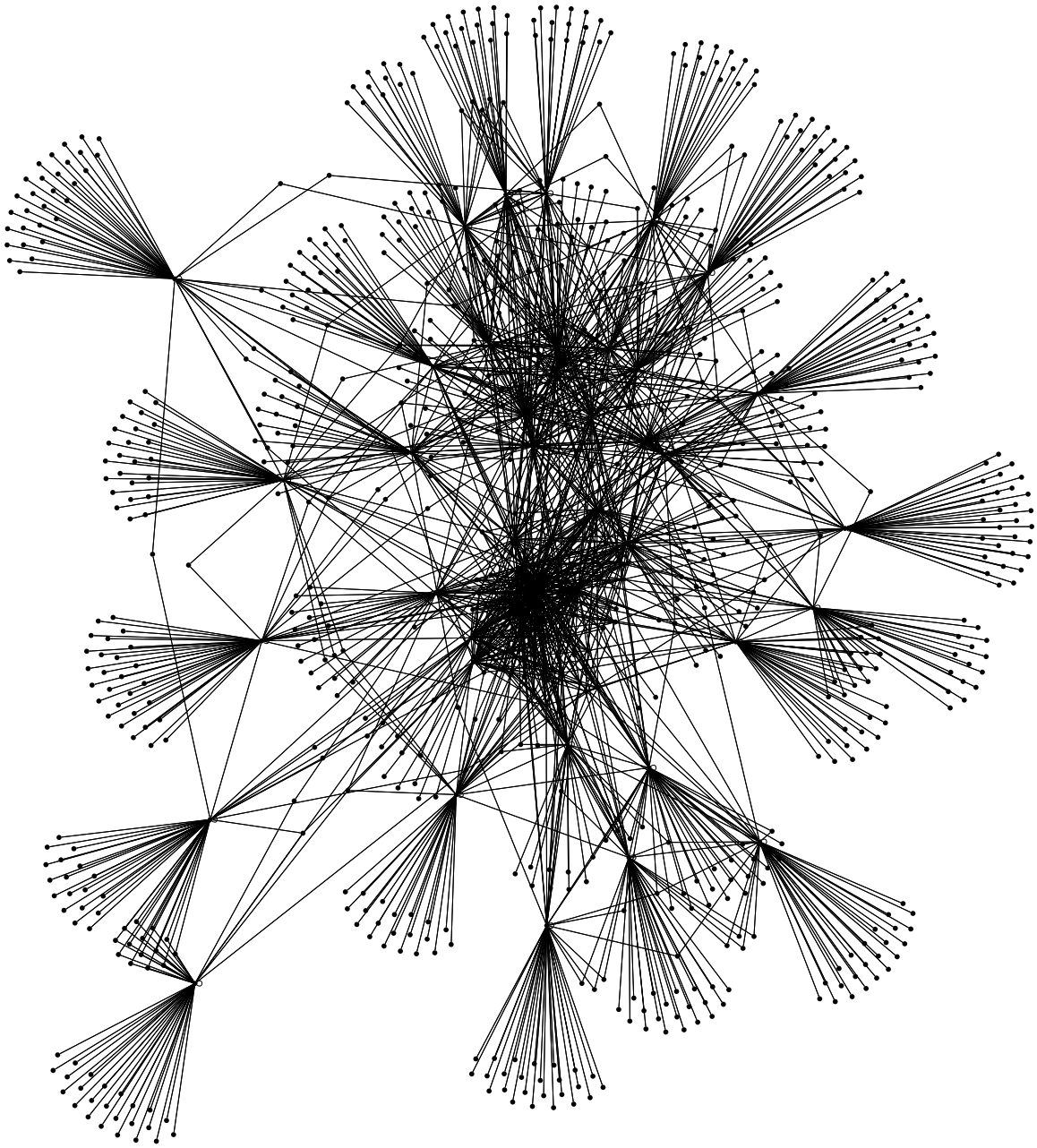


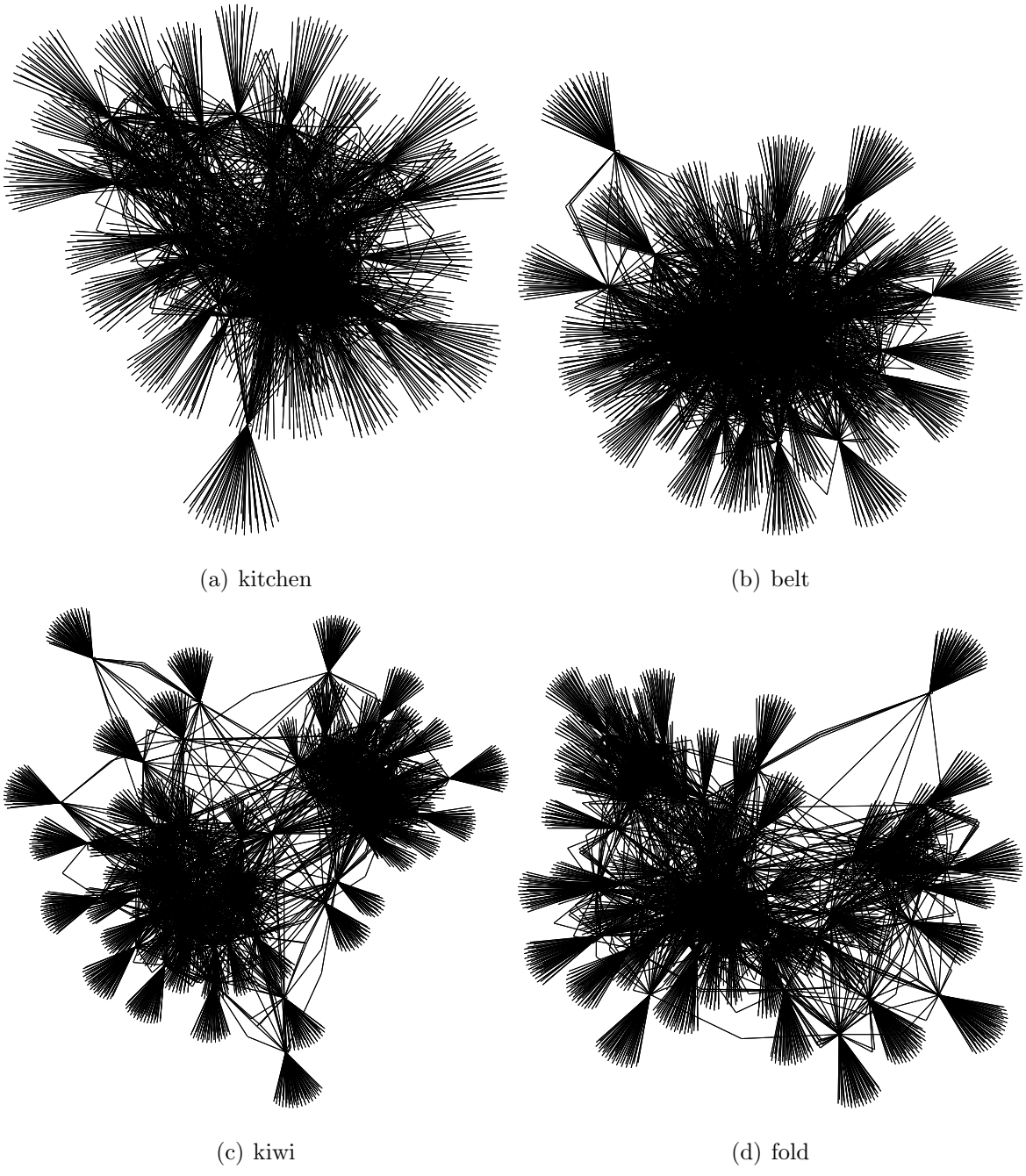
Figure 10: A semantic graph for the word *bark*, constructed using $N = 43$.

It is informative to compare the semantic graphs constructed for different types of words. Figure 11 shows the graphs for four words. The two graphs in the upper half of the figure are for words that are unambiguous, that is, they have only a single meaning, while the graphs in the lower half correspond to words that have two or more meanings. The graphs on the left half represent words that have few senses, and those on the right represent words with many senses. Note the different structures present in these graphs. Comparing the graphs of *kitchen* and *belt*, two words that have only a single meaning, reveals that both graphs have a similar structure. Each contains only a single group of highly interconnected vertices, corresponding the the unique meanings of these words. However, this group is much larger and spread out in the graph for *belt*, a word with many senses, compared to that in the graph for *kitchen*, which has only a few senses. A similar result is found by comparing the graphs of *kiwi* and *fold*. Both of these graphs have multiple groups of highly interconnected vertices, but in the graph for *kiwi*, these groups are distinct, with relatively few edges connecting the two groups. Because *fold* has multiple meanings with numerous senses, this distinction between groups, while still present, is not as prominent.

5.3 Graph Clustering

This section presents the graph clustering algorithm. This algorithm was developed by Shapiro and Haralick (1979) to analyze and deconstruct the edges of two dimensional shapes into their component parts in a shape recognition algorithm, and has since been applied to image grouping and image retrieval from a large database (Aksoy & Haralick, 1999a, 1999b). An attractive feature of this algorithm is that the items returned in each cluster are not only strongly related to the target item, but also to each other. In addition, this algorithm allows for overlap between clusters, which does not occur in other clustering algorithms, such as K-means. This allows a word such as *flow*, which is related the both the financial (i.e. cash flow) and river bank (i.e., flowing water) meaning of the word *bank*, to appear in both of these meanings' clusters.

A straightforward method of finding the clusters of a graph is to first determine



(a) kitchen

(b) belt

(c) kiwi

(d) fold

Figure 11: Semantic graphs for four words.

the complete set of major cliques in G and then iteratively merge any two cliques that have enough overlap, as determined by some user specified threshold (Kumar, 1968; Augustson & Minker, 1970). Unfortunately, finding the cliques of a graph is an $\mathcal{NP}$-complete[2] problem (Karp, 1972). Instead of finding cliques, the algorithm of Shapiro and Haralick (1979) is used to find near-cliques, referred to as dense regions, each of which contains a major clique of the graph. This algorithm runs in $O(n^2)$ time for a graph of order n. (Shapiro & Haralick, 1979). Once the set of dense regions has been found, they may be iteratively merged to form clusters.

Throughout this section, $G = (V, E)$ is a graph constructed by the method described in the previous section.

Definition 5.11 Let $x, y \in V$. The *conditional density* of vertex y given vertex x, denoted $D(y \mid x)$, is the number of nodes in the neighbourhood of x that also have y as a neighbour. More precisely,

$$\begin{aligned} D(y \mid x) &= \left|\left\{n \in V : \{x, n\} \in E \wedge \{n, y\} \in E\right\}\right| \\ &= \left|\left\{n \in N(x) : \{n, y\} \in E\right\}\right| \\ &= \left|\left\{N(x) \cap N(y)\right\}\right|. \end{aligned}$$

Note that $D(y \mid x) = D(x \mid y)$. This measure is used to find sets of vertices which are "dense enough", according to some user supplied parameters, but not necessarily as dense as the cliques of the graph. This will lift the restriction that the subgraph induced by the set of vertices is complete, removing the heavy computational burden of finding cliques. Table 2 contains the values of $D(x \mid y)$ for each pair of vertices in the graph from Figure 9.

Let k be a positive integer and $x \in V$ and consider the set of vertices

$$Z(x, k) = \{y \in V : D(y \mid x) \geq k\}.$$

The integer k determines how many neighbours a vertex y must share with x for it to be included in $Z(x, k)$. As k is increased, the vertices of $Z(x, k)$ must share

[2] A discussion of $\mathcal{NP}$-complete problems is given in Appendix A. For now, it suffices to say that an $\mathcal{NP}$-complete problem is computationally intractable, and no efficient algorithm to solve such problem is known to exist.

	1	2	3	4	5	6	7	8	9	10	11	12
1	6	4	4	4	2	3	2	3	3	2	2	2
2	4	6	5	6	2	3	2	1	1	0	0	0
3	4	5	6	6	2	3	2	1	1	0	0	0
4	4	6	6	7	2	3	2	1	1	0	0	0
5	2	2	2	2	2	2	1	0	0	0	0	0
6	3	3	3	3	2	3	2	0	0	0	0	0
7	2	2	2	2	1	2	2	0	0	0	0	0
8	3	1	1	1	0	0	0	6	6	2	2	2
9	3	1	1	1	0	0	0	6	6	2	2	2
10	2	0	0	0	0	0	0	2	2	2	2	2
11	2	0	0	0	0	0	0	2	2	2	2	2
11	2	0	0	0	0	0	0	2	2	2	2	2

Table 2: Conditional densities.

more neighbours and $G[Z(x,k)]$ becomes smaller and more tightly interconnected. Note that these regions are nested, with $Z(x,1) \supseteq Z(x,2) \supseteq Z(x,3) \supseteq \ldots$. Now, let C be a major clique of size M containing vertex x. If $y \in C$, then it must be that $D(y \mid x) \geq M$, which means $C \subseteq Z(x,M)$. Thus, for any $k \leq M$, we have $C \subseteq Z(x,k)$, so $k \leq M \leq |Z(x,k)|$. Hence, any k that does not satisfy the inequality $|Z(x,k)| \geq k$ cannot be the size of a major clique containing vertex x. Only values of k that satisfy this last inequality will be considered.

Definition 5.12 Let $Z(x,k) = \{y \in V : D(y \mid x) \geq k\}$ and let $j = \max\{k : |Z(x,k)| \geq k\}$. The set of vertices, $Z(x) = Z(x,j)$, is called a *dense region candidate*, or *DRC*.

$Z(x)$ contains a major clique of size j containing x, but may also include some additional nodes which are not part of the clique, and so is not necessarily a clique itself.

A few additional restrictions are placed on the DRC before it can be called a dense region. First, regions should not be too small, as determined by some user

provided threshold. This can easily be enforced by rejecting any DRC containing fewer than a user specified number of vertices. Secondly, each vertex in the DRC should be adjacent to a high proportion of other vertices in the DRC. The following two definitions are used in meeting this second requirement.

Definition 5.13 Let $B \subseteq V$. The *association* of a vertex $x \in V$ to the set B, denoted $A(x \mid B)$, is the proportion of vertices in B which are also in $N(x)$, given by

$$A(x \mid B) = \frac{|N(x) \cap B|}{|B|},$$

where $0 \leq A(x \mid B) \leq 1$.

Definition 5.14 The *compactness* of a set $B \subseteq V$, denoted $C(B)$, is the average association of the vertices in B to B itself, given by

$$C(B) = \frac{1}{|B|} \sum_{x \in B} A(x \mid B),$$

where $0 \leq C(B) \leq 1$.

A dense region of a graph can now be defined:

Definition 5.15 Let $B \subseteq V$. Given MINSIZE, a positive integer, MINASSOC, a real number in the interval $[0, 1]$, and MINCOMP, a real number in the interval $[\mathsf{MINASSOC}, 1]$, B is called a *dense region* if all of the following conditions are met:

1. $B = \{y \in Z(x) : A\,(y \mid Z(x)) \geq \mathsf{MINASSOC}\}$ for some $x \in V$,
2. $C(B) \geq \mathsf{MINCOMP}$, and
3. $|B| \geq \mathsf{MINSIZE}$.

The first condition states that the vertices in B are the vertices from a DRC that have the highest association with the DRC. The second condition ensures that the average association of the vertices in B are high enough, as determined by the threshold MINCOMP. These two conditions ensure that each vertex in B is adjacent to a high number of other vertices of B. The last condition ensures that the region

x	$N(x)$	j	$Z(x) = Z(x, y)$	$C(Z(x))$
1	1, 2, 3, 4, 8, 9	4,	1, 2, 3, 4	1
2	1, 2, 3, 4, 5, 6	4	1, 2, 3, 4	1
3	1, 2, 3, 4, 6, 7	4	1, 2, 3, 4	1
4	1, 2, 3, 4, 5, 6, 7	4	1, 2, 3, 4	1
5	2, 4	2	1, 2, 3, 4, 6	0.88
6	2, 3, 4	3	1, 2, 3, 4, 6	0.88
7	3, 4	2	1, 2, 3, 4, 6	0.88
8	1, 8, 9, 10, 11, 12	3	1, 8, 9	1
9	1, 8, 9, 10, 11, 12	3	1, 8, 9	1
10	8, 9	2	1, 8, 9	1
11	8, 9	2	1, 8, 9	1
12	8, 9	2	1, 8, 9	1

Table 3: The neighbourhood, maximal value of j, dense region candidate, and compactness of each vertex in G

is large enough to be of interest. Note that if $\mathsf{MINASSOC} = 1$ and $\mathsf{MINCOMP} = 1$, then the dense regions found will be the major cliques of G.

The maximal values for j, the dense region found around each node, and the compactness of the dense regions in our example are shown in Table 3, together with the neighbourhood of each vertex. These regions were found using the parameters $\mathsf{MINSIZE} = 3, \mathsf{MINASSOC} = 0.5$, and $\mathsf{MINCOMP} = 0.75$.

To determine the clusters of G, a dense region is found around each vertex in G and these regions are then merged.

Definition 5.16 The *overlap* between two sets, B_1 and B_2 is

$$\max\left\{\frac{|B_1 \cap B_2|}{|B_1|}, \frac{|B_1 \cap B_2|}{|B_2|}\right\}.$$

If the overlap between two dense regions exceeds a user supplied threshold between 0 and 1, called MINOVERLAP, then the two regions are merged, provided that each

of the nodes in the resulting set have a high enough association to the new set. This merging is iteratively performed until no two regions can be merged. The result is a set of clusters of the graph G.

After merging the dense regions given in Table 3 (using MINOVERLAP $= 0.75$), the resulting clusters are $\{1, 2, 3, 4, 6\}$ and $\{1, 8, 9\}$. The subgraphs induced by these clusters are shown in Figures 12 and 13, respectively. Note that vertex 6 was included in the first cluster, even though it was not in the result set when vertex 1 was used as the query item (see Table 1). This is because vertex 6 is related to a high enough number of other vertices in the cluster, as determined by MINASSOC and MINCOMP, to warrant its inclusion.

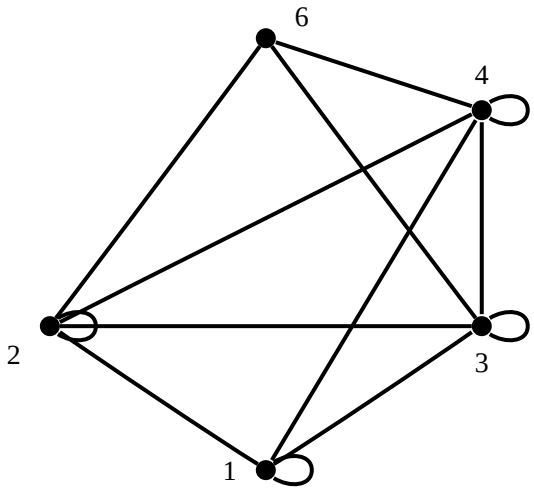


Figure 12: Subgraph induced by $\{1, 2, 3, 4, 6\}$.

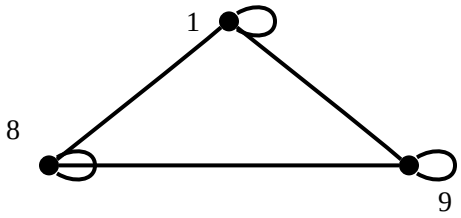


Figure 13: Subgraph induced by $\{1, 8, 9\}$.

6 Method

Now that the necessary background has been provided, a detailed description of the algorithm discussed in Chapter 2 is presented. Section 6.1 introduces the notation required to present the algorithm more rigorously, and the algorithm itself is discussed in Sections 6.2 through 6.5.

6.1 Analyzing the Corpus

Let $\mathcal{W} = \{w_1,\ w_2,\ \ldots,\ w_\rho\}$ be a set of ρ unique words. This set is called the *dictionary* and contains the full set of strings that are considered valid English words. Let $\mathcal{T} = (t_1, t_2, \ldots, t_\nu)$ be an ordered list of words such that for every $t_i, i = 1, \ldots, \nu$, there is a word $w_x \in \mathcal{W}$ such that $t_i = w_x$. $\mathcal{T}$ is called a *corpus*, and the semantic associations created in this thesis are based on lexical co-occurrence between the words in this these. Note that $\mathcal{T}$ may consist of many separate written works, but that these works are abstracted into a single entity. Also note that the words in $\mathcal{T}$ are not necessarily unique, and as such, may occur multiple times in $\mathcal{T}$. In fact, if each word in $\mathcal{T}$ occurred only a single time, the type of analysis performed in this thesis would be ineffective and would fail to produce any usable results. Throughout this discussion, w_j is used to denote the target word and w_i is used to denote a potential associate of w_j.

A small window is constructed and passed over each word in the corpus, recording which words occur together in a window. The parameter η determines how far this window extends on either side of the target word. Let $I(t_i)$ be the window centered around t_i, the i^{th} word in $\mathcal{T}$. This window contains the η words preceding t_i and the η words following t_i, but does not contain t_i itself. $I(t_i)$ can be written as

$$I(t_i) = (t_{i-\eta}, t_{i-\eta+1}, \ldots, t_{i-1}, t_{i+1}, \ldots, t_{i+\eta-1}, t_{i+\eta})\,.$$

The functions in the next three definitions provide information about how many times words occur together in the corpus.

Definition 6.1 Let $w_i, w_j \in \mathcal{W}$ and

$$f_\ell^n(w_i \mid w_j) = |\{k : t_k = w_j \wedge t_{k+n} = w_i\}|.$$

$f_\ell^n(w_i \mid w_j)$ is called the *local co-occurrence function*.

$f_\ell^n(w_i \mid w_j)$ counts how many times w_i occurs in the n^{th} position in an interval around w_j. If $n < 0$, then w_i appeared before w_j, and w_i is said to have occurred in the pre-context of w_j. If $n > 0$, then w_i appeared after w_j, and is said to have appeared in the post-context.

Definition 6.2 Let $w_i, w_j \in \mathcal{W}$ and let

$$f_\ell(w_i \mid w_j) = \sum_{\substack{n=-\eta \\ n\neq 0}}^{\eta} f_\ell^n(w_i \mid w_j).$$

$f_\ell(w_i \mid w_j)$ is called the *local frequency function*.

$f_\ell(w_i \mid w_j)$ counts how many times a word w_j occurred in an interval, in any position, around w_i.

Definition 6.3 Let $w_i \in \mathcal{W}$ and let

$$\overline{f_g}(w_i) = |\{k : t_k = w_i\}|.$$

$f_g(w_i)$ is called the *absolute global frequency function*.

$f_g(w_i)$ counts how many times w_i appeared in the entire corpus. A more interesting measure global frequency per million words of written text, rather than a number that depends on the size of the corpus scanned. An additional function that provides exactly this value is defined next.

Definition 6.4 Let $w_i \in \mathcal{W}$. The *global frequency function*, denoted $f_g(w_i)$, is given by

$$f_g(w_i) = \frac{1000000\overline{f_g}(w_i)}{\sum_{w_k \in \mathcal{W}} \overline{f_g}(w_k)}.$$

Once this window has been passed over the entire corpus, recording the information specified above, how words are semantically related can be determined based on these data.

6.2 Removing Frequency Effects

At this point, an enormous amount of data about the corpus has been collected. The number of times each word appears, given by $f_g(w_i)$, and how many times each pair of words appears together, given by $f_\ell(w_i \mid w_j)$ have been calculated. In addition, the number of times a word appears in each relative window position around a target word, given by, $f_\ell^n(w_i \mid w_j)$, has been collected from the corpus. This last number must be converted to a measure of how important the co-occurrence is. Suppose $f_\ell^n(w_i \mid w_j)$ is large. If w_i and w_j are both very high frequency words, this high incidence of co-occurrence may not mean that w_i and w_j are semantically related. It may merely be a side effect of both words having high frequency. Keeping this in mind, any measure that determines the strength of the relationship between two words must be independent of the global frequency of the two words.

Definition 6.5 Let $w_i, w_j \in \mathcal{W}$. The *local co-occurrence strength* of w_i given w_j is

$$s^n(w_i \mid w_j) = \frac{f_\ell^n(w_i \mid w_j)}{\sqrt{f_g(w_i)}} \left(\frac{1}{1 + e^{-f_g(w_i)}} \right) \left(\frac{1000000 - f_g(w_i)}{1000000} \right)^{32}$$

Higher values of $s^n(w_i \mid w_j)$ correspond to a stronger relationship between w_i and w_j. It is beneficial to inspect this function in more detail. Let

$$\lambda(w_i) = \frac{1}{\sqrt{f_g(w_i)}} \left(\frac{1000000 - f_g(w_i)}{1000000} \right)^{32}$$

and

$$\gamma(w_i) = \frac{1}{1 + e^{-f_g(w_i)}}.$$

A graph of $\lambda(w_i)$ for words with a global frequency between 0 and 65,000[3] is shown in Figure 14. Figure 15 shows a graph of this function for low frequency words (less than ten occurrences per million words).

$\lambda(w_i)$ is multiplied by the raw co-occurrence counts to reduce these values for high frequency associates. Note that $\lambda(w_i) \approx 1$ when $f_g(w_i) = 1$. For words with $f_g(w_i) < 1, \lambda(w_i) > 1$, and co-occurrence values are actually increased. To counterbalance this effect, the co-occurrence counts are multiplied by $\gamma(w_i)$, shown in Figure 16.

[3] In our corpus, *the* was the most frequency occurring word, with a frequency of 64,355.56 per million words.

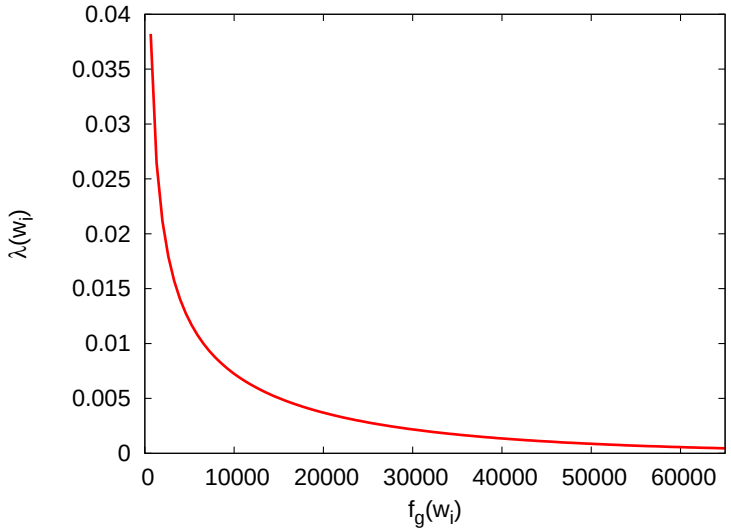


Figure 14: Graph of function $\lambda(w_i)$

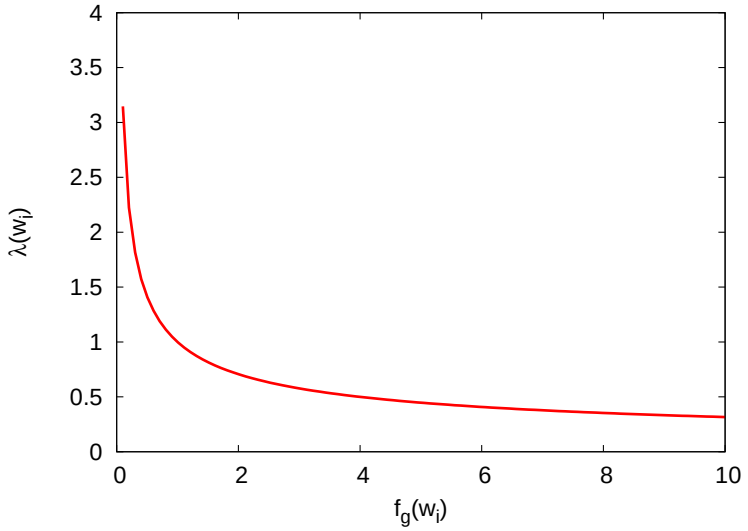


Figure 15: Graph of $\lambda(w_i)$ for low frequency words.

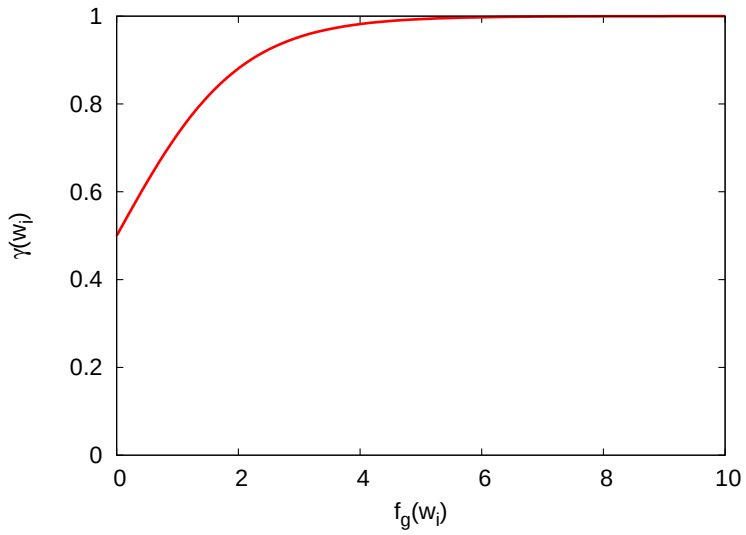


Figure 16: Graph of function $\gamma(w_i)$

In the analysis performed in this thesis, any word occurring less than 0.3 times per million words was excluded. If $w \in \mathcal{W}$ with $f_g(w) = 0.3$, then $\lambda(w) = 1.825$. Thus, co-occurrence for these ultra low frequency words is nearly doubled. However, for the same word, $\gamma(w) = 0.5744$, and the product of the two is $\lambda(w)\gamma(w) = \lambda\gamma(w) = 1.049$. This function leaves co-occurrence counts for low frequency words relatively unchanged, but counts for high frequency associates are greatly reduced. This function,

$$\lambda\gamma(w) = \frac{1}{\sqrt{f_g(w)}} \left(\frac{1000000 - f_g(w)}{1000000} \right)^{32} \left(\frac{1}{1 + e^{-f_g(w)}} \right),$$

shown in Figure 17, is used as the co-occurrence adjustment factor. $\lambda\gamma(w_i)$ is multiplied by $f_\ell^n(w_i \mid w_j)$ to obtain the final co-occurrence value, $s^n(w_i \mid w_j)$.

6.3 Creating Semantic Representations

Next, a weight α_i is assigned to the i^{th} window position, $i = -\eta, \ldots, \eta$. To simplify formulas, we will set $\alpha_0 = 0$. These variables are assigned values that allow optimal performance of this method in its task of measuring word ambiguity. As the algorithm has yet to described in its entirety, a description of how these weights are determined is postponed until Section 6.5, after all steps of the method have been presented. For

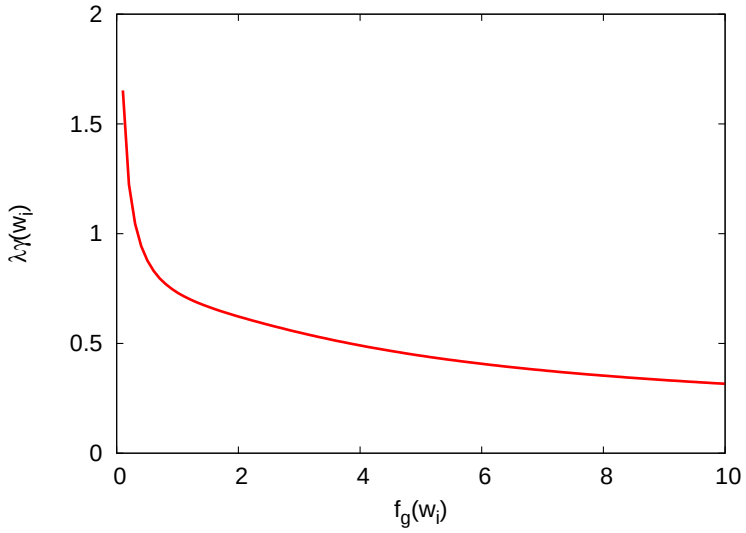


Figure 17: Graph of function $\lambda\gamma(w_i)$

now, assume that an optimal value has been assigned to each α_i, $i = -\eta, \ldots, \eta$, using the technique presented in Section 6.5.

For each $w_i \in \mathcal{W}$, a vector representing w_i in semantic space is created.

Definition 6.6 Let $\vec{w}_j = \langle w_j^1, w_j^2, \ldots, w_j^\rho \rangle$, with the i^{th} component given by

$$w_j^i = \sum_{\ell=-\eta}^{\eta} \alpha_\ell s^\ell(w_i \mid w_j).$$

$\vec{w}_j$ is the *semantic representation* of w_j and w_j^i is the *semantic association* of w_i to w_j.

Once each component is calculated, $\vec{w}_j$ is normalized to unit length using the ℓ^2 norm, and use this as our vector-based representation of the semantic characteristics of w_j. The contents of these vectors, as opposed to some form of distance metric, are used to determine semantic relationships between words. Note that, in general, $w_j^i \neq w_i^j$, which is consistent with the literature (Chwilla, Hagoort, & Brown, 1998; Koriat, 1981; Thompson-Schill, Kurtz, & Gabrieli, 1998).

6.4 Measuring Ambiguity

This section describes how the semantic representations can be used to measure the degree of ambiguity inherent in a word, w_i.

Using the method of Section 5.2, a "semantic graph" for a target word w_i, containing only the words most strongly related to w_i, is constructed. In the current setting, the N closest neighbours of w_i are found by examining $\vec{w}_i$ and finding the N highest values of w_i^j. Let these values be $w_i^{j_1}, w_i^{j_2}, \ldots, w_i^{j_N}$. For each w_{j_n}, $n = 1, \ldots, N$, the components of $\vec{w}_{j_n}$ are inspected to find the highest N values of $w_{j_n}^k$. Let these values be $w_{j_n}^{k_m}$, $n = 1, \ldots, N, m = 1, \ldots, N$, and let $w_{k_{n,m}}$ be the semantic associate of w_{j_n} corresponding to $w_{j_n}^{k_m}$.

These values are used to construct the semantic graph, G, for word w_i. Let

$$V = \{w_i\} \cup \{w_{j_n} \colon n = 1, \ldots, N\} \cup \{w_{k_{n,m}} \colon n = 1, \ldots, N, m = 1, \ldots, N\}$$

be the vertex set of G and

$$E = \big\{\{w_i, w_{j_n}\} \colon n = 1, \ldots, N\big\} \cup \big\{\{w_{j_n}, w_{k_{n,m}}\} \colon n = 1, \ldots, N, m = 1, \ldots, N\big\}.$$

be the edge set of G. Then $G = (V, E)$ is the semantic graph for w_i.

Next, the graph clustering algorithm described in Section 5.3 is applied to G. Let $C_1, C_2, \ldots, C_q$ be the clusters of G. Each cluster is interpreted as containing only words related to a single meaning of w_i. The union of all clusters,

$$C = C_1 \cup C_2 \cup \cdots \cup C_q,$$

is the SN of w_i.

If only a single cluster is found by the algorithm, then w_i is a non-ambiguous word. Otherwise, we will use Shannon's entropy formula to measure w_i's ambiguity. Let $C_\ell = \{c_{\ell,1}, c_{\ell,2}, \ldots, c_{\ell,\sigma_\ell}\}$, $\ell = 1, \ldots, q$, where σ_ℓ is the number of words in C_ℓ, and $c_{\ell,j} = w$ for some $w \in \mathcal{W}$, $j = 1, \ldots, \sigma_\ell$. Let

$$f(C_\ell) = \sum_{j=1}^{\sigma_\ell} f_g(c_{\ell,j})$$

be the summed orthographic frequency of the words in C_ℓ. The probability that a word in the SN of w_i is in C_ℓ is given by

$$p_\ell = p(C_\ell) = \frac{f(C_\ell)}{\sum_{j=1}^{q} f(C_j)}.$$

These probabilities are calculated for $\ell = 1, \ldots, q$, and are then used to calculate the ambiguity and balance measures from Twilley et al. (1994), using Equation 1 given in Chapter 3.

The parameters of the graph clustering algorithm (N, MINSIZE, MINASSOC, MINCOMP, and MINOVERLAP) are determined using the same method that is used to determine the window weights, which is described in the next section.

6.5 Determining Window Weights and Graph Clustering Parameters

A genetic algorithm (GA; Holland, 1992) is used to determine the weight assigned to each window position, as well as the parameters used in the graph clustering algorithm. A GA is an iterative global search method that uses Darwin's principles of natural selection (Darwin, 1859) to evolve a near optimal solution to a problem. In this situation, the solution space is large and complex. One cannot perform any mathematical analysis or calculate derivatives to guide a search. In fact, this problem cannot even be represented as a single function (or group of functions) to be optimized. GA is used because this problem does not meet any of the requirements (i.e., differentiability, or even continuity) needed by a traditional search algorithm, such Newton's method or the conjugate-gradient method.

The GA begins by creating several random solutions to the given problem. Each solution is called an individual, and a set of solutions is referred to as a population. Individuals in the population are evaluated by a fitness function, and the strongest individuals are combined to create a new population, called a new generation. The construction of new individuals by combining the features of two individuals from the previous generation is referred to as crossover. After several generations have been computed in this way, the individuals will converge to a near optimal solution.

In this situation, an individual specifies a complete set of window weights and clustering parameters. That is, each individual consists of a weight for each window position, a value for N, the number of items returned in the result sets used to construct the semantic graph, and values of the clustering algorithm parameters, MINSIZE, MINASSOC, MINCOMP, and MINOVERLAP.

An individual is evaluated by selecting a subset of the 566 words for which ambiguity norms are available, applying the algorithm described in Sections 6.1 to 6.4 to determine the ambiguity of each of these words, then calculating the correlation between these data and the corresponding data from Twilley et al. (1994). Individuals that produce a stronger correlation are deemed to better perform the task at hand, and are more likely to be selected by the GA for use in crossover. In addition, individuals are penalized in the event that they are unable to find any clusters in the semantic graph of a given word. Let p be the proportion of words from the evaluation set for which the algorithm was unable to find any clusters. To avoid assigning high fitness values to individuals for which this situation is likely to occur, p is subtracted from the correlation to obtain our final fitness value.

In this work, a custom GA was written in C++ using the LAM implementation (Squyres & Lumsdaine, 2003; Burns, Daoud, & Vaigl, 1994) of the Message Passing Interface (MPI) library. This program was run on a cluster of 11 Macintosh G4's, each with two CPUs. The population size was set to 300 and the algorithm was allowed to run for 3045 generations. The crossover rate was set to 0.95 and the mutation rate was set to 0.05. Individuals were selected for crossover using tournament selection with a tournament size of three, and 375 randomly selected words from the Twilley et al. (1994) norms were used to evaluate individuals. During crossover, new individuals were created as a linear combination of the individuals selected for crossover. Let v_1 be a parameter from individual I_1, and let v_2 be the corresponding value from another individual, I_2. Let w_1 and w_2 be the values to be calculated for the new individuals being created. A random value $x \in [0, 1]$ is selected, and the parameters for the new individuals are:

$$\begin{aligned} w_1 &= xv_1 + (1-x)v_2 \\ w_2 &= xv_2 + (1-x)v_1. \end{aligned}$$

Values for each parameter were calculated in this way, with both N and MINSIZE rounded to the nearest integer, to construct the new individuals. During mutation, the value of N and MINSIZE were randomly incremented or decremented, each weight was changed by a random value selected from a normal distribution with mean 0 and standard deviation 3, and all of the other parameters were altered by a random value from a normal distribution with mean 0 and standard deviation 0.1.

It is important to note that, until execution of the GA has been completed, the final weights used by the method are unknown. The algorithm attempts to use several different sets of weights and evaluates the performance of the algorithm under each set of weights across a number of words (in this work, 375 words were used). In addition, the graph clustering parameters are evaluated in the same manner. This means that, for each of the 375 words selected from the Twilley et al. (1994) data set, the GA must apply the weights to the local co-occurrence strengths calculated earlier, construct a semantic graph, apply the graph clustering algorithm, then measure the ambiguity of the word. As such, in Section 6.3, the phrase "assume that an optimal value has been assigned to each α_i" grossly curtails the amount of work required to assign values to each α_i.

7 Results

In these experiments, a corpus of approximately 267 million words was analyzed using a window that extended 15 words on either side of the target. Any word with a frequency of less than three per ten million words was excluded from the analysis. These words appeared less than 80 times in the corpus, and the co-occurrence data was insufficient for determining the semantic associations between these words. From our corpus a dictionary of 64,391 distinct words was constructed, of which 37,269 occurred with sufficient frequency to be included in our analysis. After co-occurrence data was collected and adjusted to reduce frequency effects, a GA was used to determine the optimal window weights and graph clustering parameters. The optimal weights are shown in Table 4. These weights were allowed to vary between 0 and 1000. The

Distance from Target	Pre-context	Post-context
1	416.89	486.73
2	491.53	412.73
3	524.48	461.62
4	480.24	492.30
5	513.03	458.10
6	573.83	497.03
7	482.00	381.82
8	521.66	574.40
9	416.86	515.92
10	564.91	619.43
11	419.51	475.66
12	637.58	446.94
13	455.13	484.82
14	566.26	532.40
15	529.15	539.81

Table 4: Optimal window weights

minimum and maximum allowable values for each of the clustering parameters, as well as the optimal values as determined by the GA, are given in Table 5.

7.1 Semantic Representations

The practical purpose of this exercise was to develop a database the would be useful to psychologists interested in semantic process. Such a database would require the the measure of semantics be free from orthographic frequency, be sensible in terms of their word lists, and provide information that could be used in different types of experiments. This section provides an overview of how well this objective has been met.

Parameter	Minimum	Maximum	Optimal
N	10	60	43
MINSIZE	1	60	1
MINASSOC	0	1	0.188
MINCOMP	MINASSOC	1	0.860
MINOVERLAP	0	1	0.531

Table 5: Optimal graph clustering parameters

7.1.1 Independence of Frequency

To determine the amount of influence from frequency in our semantic representations, the correlation between orthographic frequency and semantic association within each vector was calculated. The average correlation was 0.077, with a standard deviation of 0.124. The correlation between the absolute values of these correlations and the orthographic frequency of the target word is 0.129, revealing that there is a stronger influence of orthographic frequency in high frequency words.

In addition, the correlation between semantic association across all targets and the orthographic frequency of the associate was calculated. The average correlation was -0.043, with a standard deviation of 0.016. Thus, the vectors computed by this method contain only minimal interference from orthographic frequency, and these effects are most prominent within the representations of higher frequency words.

7.1.2 Semantic Neighbourhoods

Table 6 contains the ten closest semantic associates for the ten words given as the headings in this table. Note that these words are both semantically (*volcano-mountain*) and associatively (*volcano-lava*) related to the target.

7.1.3 Category Exemplars

To investigate whether or not this method was able to extract categorical information from the corpus, the method's ability to find category exemplars (i.e., an *apple* is a

COFFEE	**FEAST**	**HAMMER**	**FLEET**	**BOAT**
CUP	WEDDING	ANVIL	SHIP	OARS
TEA	BANQUET	SLEDGE	ADMIRAL	ROW
POT	PASSOVER	CHISEL	VESSELS	SAIL
SUGAR	GUESTS	STEAM	SAILED	SHORE
BREAKFAST	CELEBRATE	NAIL	NAVAL	ASHORE
SIPPED	INVITED	TONGS	SEA	WATER
MUG	WINE	FORGE	BOATS	CREW
DRANK	HARVEST	MACK	COAST	RIVER
COCOA	FESTIVAL	CLAW	HARBOUR	FERRY
TABLE	MERRY	DRILL	BRITISH	STERN

BARK	**PLANT**	**CIGARETTE**	**MOUNTAIN**	**VOLCANO**
BIRCH	SOIL	LIT	RANGES	ERUPTION
TREE	NUCLEAR	SMOKING	PEAKS	CRATER
DOG	ANIMAL	ASH	VALLEY	LAVA
SAP	SEED	ASHTRAY	SLOPES	CONE
TWIGS	POLLEN	LIGHTER	ROCKY	EARTHQUAKE
LEAVES	LEAVES	TOBACCO	SUMMIT	ISLAND
PINE	SPECIES	STUBBED	BIKE	ASH
BITE	ROOTS	PUFF	TOP	ACTIVE
TRUNK	NUTRIENTS	SMOKER	SIDE	DORMANT
BRANCHES	GROWTH	MATCH	CLIMBING	MOUNTAIN

Table 6: Most strongly related semantic neighbours

member of the category *fruit*) was tested. Ten categories were selected and, by inspecting the vector representing the name of the category, ten exemplars with the strongest association to each category were found. The results are shown in Table 7, with the category names given as headings.

Typical exemplars appear to have stronger associations to the category than those that are less common. For example, when the semantic associates of *fruits* are sorted by association strength, the word *peaches* appears 26^{th} from the top of the list. The word *plantain*, a much less common type of fruit, did not appear until position 727.

7.1.4 Multidimensional Scaling

As further evidence of the existence of categorical information within these vector representations, multidimensional scaling was applied to several of the vectors created by this method. Using the same words used to evaluate HAL, the dimension of the vectors was reduced to only two dimensions and each of these words were plotted in the plane. The results, shown in Figure 18, are similar to those found using vectors from HAL. However, the word *tooth*, which was incorrectly grouped with the animal names by HAL, was correctly classified as a body part. Also note that country names were very distinctly grouped from the body parts and animal names.

To further investigate this property of our vectors, 31 words from the categories *fruits*, *vegetables*, *tools*, and *furniture* were selected multidimensional scaling was applied to their vectors. The results are shown in Figure 19. Again, words from different categories were grouped together. There were four notable exceptions. First, *saw* was classified as a piece of furniture. However, this word was very close to *table*, which is a common type of saw. Next, *desk* was grouped in the tools category. A desk is a common setting in which work is performed, and we suspect that *desk*'s similarity to *work* was more strongly represented than its similarity to other pieces of furniture. Analysis of the vectors representing *desk*, *furniture*, and *work* revealed that the distance between *desk* and *work* (1.02 using Euclidean distance, and 0.0154 using rectilinear distance) is slightly smaller than the distance from *desk* to *furniture* (1.06 using the Euclidean metric, and 0.0174 using rectilinear distance). The correlation

ANIMALS	FOOD	VEGETABLES	SPORTS	FRUITS
DOG	VEGETABLES	POTATOES	FOOTBALL	ORANGES
HORSE	PASTA	CARROTS	TENNIS	BERRIES
CAT	CEREAL	BEANS	BASKETBALL	BANANAS
ELEPHANT	MEAT	TOMATOES	GOLF	GRAPES
PIG	MILK	PEAS	SOCCER	APPLES
BIRD	FRUIT	CABBAGE	RUGBY	APRICOTS
GOAT	CHEESE	ONIONS	BOXING	PEARS
COW	FISH	MUSHROOMS	CRICKET	PLUMS
MONKEY	CHOCOLATE	LETTUCE	SQUASH	LEMONS
KANGAROO	RICE	TURNIPS	SWIMMING	PEACHES

FISH	BIRDS	TREES	COUNTRIES	EMOTIONS
TROUT	HUMMING	FIR	GERMANY	ANGER
SALMON	PIGEON	PINE	JAPAN	PASSION
COD	PARROT	PALM	BRITAIN	FEAR
CARP	SPARROW	APPLE	AFRICA	EXCITEMENT
GOLDFISH	THRUSHES	PEAR	FRANCE	RAGE
HERRING	STARLING	BEECH	AMERICA	PAIN
MACKEREL	ROBINS	OAK	SWEDEN	JOY
TUNA	CROWS	BIRCH	ITALY	EMPATHY
SHARK	BLACKBIRD	ELM	BRAZIL	PITY

Table 7: Top category exemplars

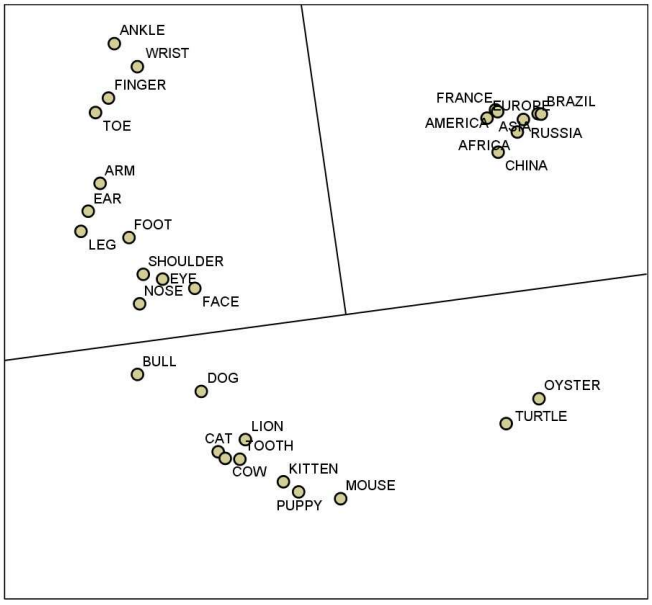


Figure 18: Multi-dimensional scaling of *animals*, *countries*, and *body parts*.

and cosine between *desk* and *work* (0.46 and 0.51, respectively) and between *desk* and *furniture* (0.43 and 0.47, respectively) were calculated, and it was found that there was a slightly stronger similarity between *desk* and *work*.

Finally, both *lemon* and *kiwi* were classified as vegetables rather than fruits. While *lemon* was classified as a vegetable, it was also placed very close to the other fruits. However, *kiwi* was grouped separately from all other fruits and vegetables. One reason for this may be the ambiguous nature of this word. A *kiwi* may refer to a fruit, a small bird, or a resident of New Zealand. In addition, this word has a very low orthographic frequency (less than one occurrence per million words). The low amount of co-occurrence data collected for this semantically rich word may have prevented the method from constructing a representation consistent with those created for other fruits.

As a final analysis of the category information contained within these vectors, multidimensional scaling was performed on the vectors corresponding to 72 words. The results are shown in Figure 20. Words did not fall into categories as distinctly

Figure 19: Multi-dimensional scaling of *fruits*, *vegetables*, *tools*, and *furniture*.

as in Figures 18 and 19, but there remains a strong tendency for similar concepts to group together. The fruits and vegetables were grouped together, and most animals were grouped near the body parts. There is also a strong separation between non-living and living objects. While this experiment does not provide much additional insight into the categorical information contained within our vectors, it does begin to give an overview of the general structure of the semantic memory formed by this method.

7.2 Ambiguity Measurements

The ambiguity measurements created by this method are now examined. The correlation between U (as calculated by this method) and orthographic frequency is 0.298, and the correlation between B (as calculated by this method) and frequency is 0.189. While this may suggest that there is some relationship between written frequency and ambiguity, this may not be the case. Figure 21 shows scatter plots of U and B versus orthographic frequency. These data suggest that there is no linear relationship

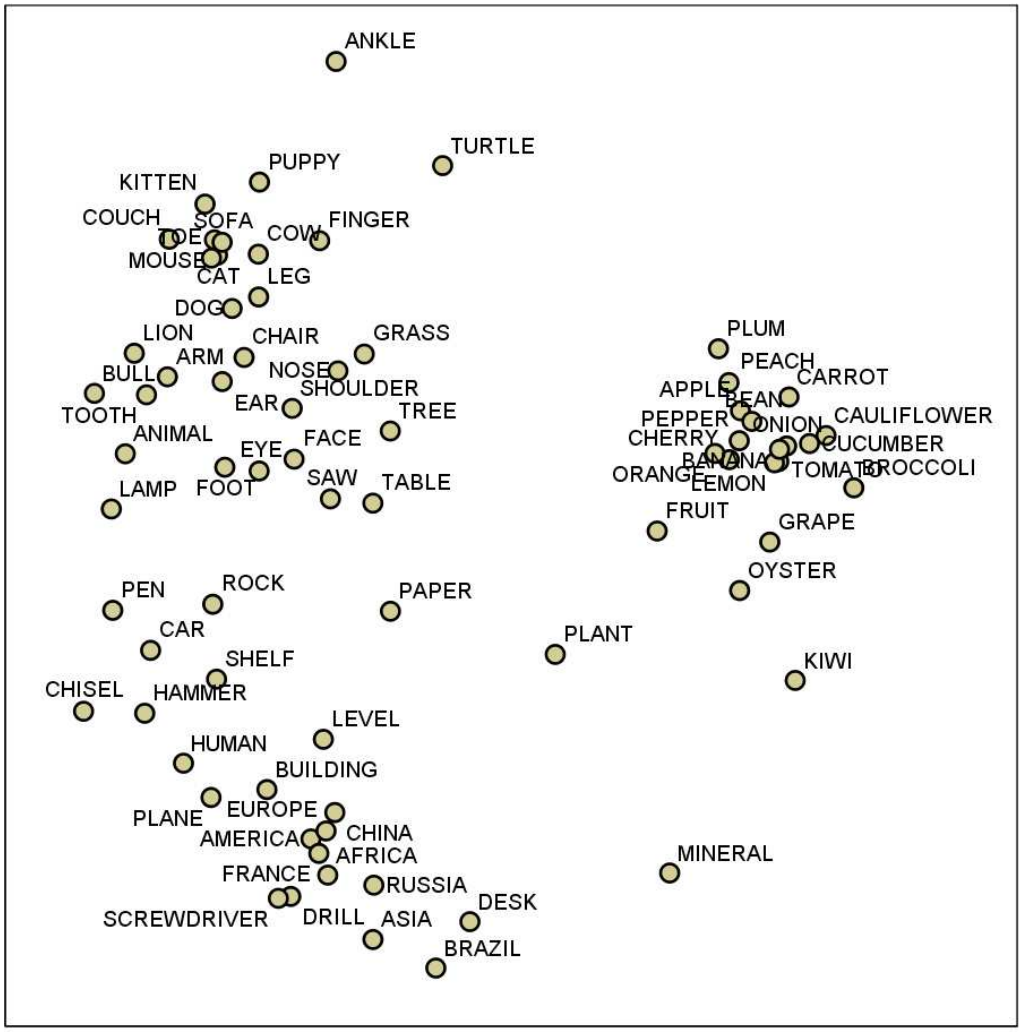
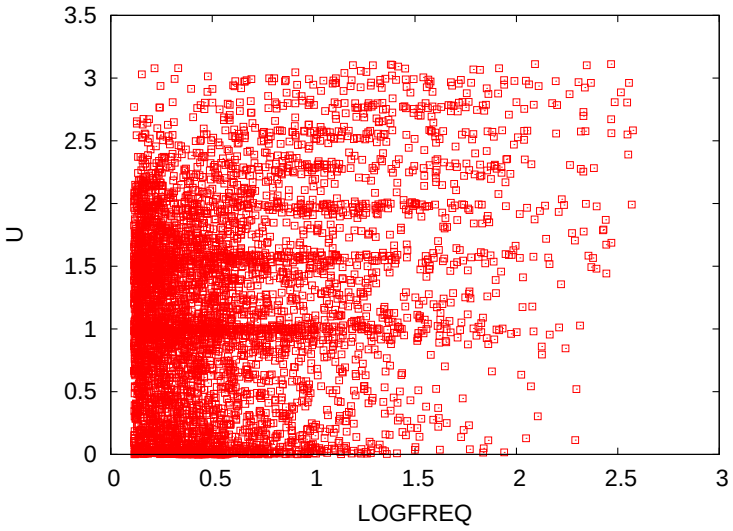
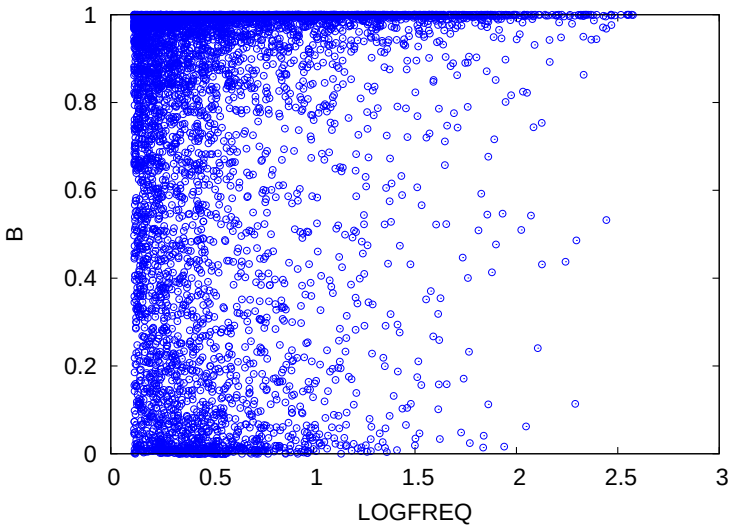


Figure 20: Multi-dimensional scaling of 72 words

between either U or B and frequency.



(a) U vs. Log Frequency



(b) B vs. Log Frequency

Figure 21: Scatter plots of ambiguity versus frequency.

Comparison with the data from Twilley et al. (1994) revealed only a weak correlation of 0.112. This may be explained by examining the type of words used in this study. Since the norms collected were for homographs, many of the words under consideration have two distinct meanings. One possibility is that this method better differentiates between senses of a word than between distinct meanings.

To further investigate this possibility, the ambiguity measurements were compared to RT in LD (taken from Balota, Cortese, & Pilotti, 1999). The correlation between U and RT is -0.265, and the correlation between B and RT is -0.190. A graph of U plotted against RT, with the line of best fit, is shown in Figure 22. Note that these data suggest a linear relationship between U and RT. As discussed in Chapter 3, homographs should be recognized more slowly than non-ambiguous words, and polysemous words should be measured faster. Since the data produced by this method predicts that high "ambiguity" words are recognized faster, it is possible that this algorithm is actually measuring the degree of polysemy in a word's meaning.

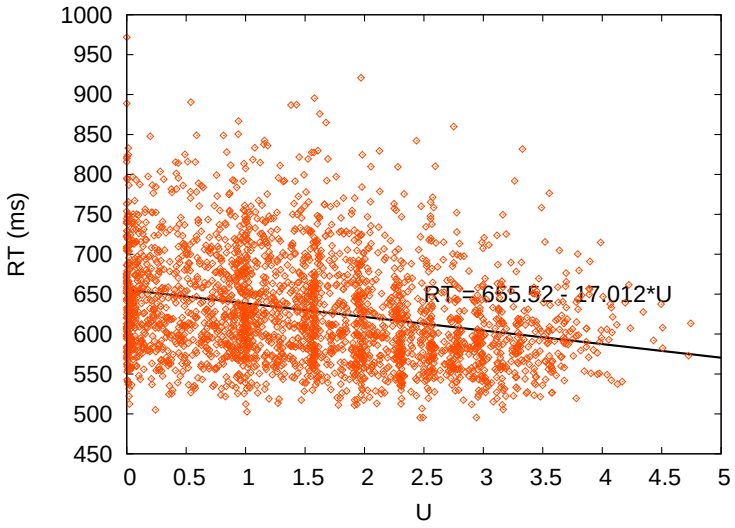


Figure 22: U vs. RT with line of best fit.

Inspecting the clusters found in the semantic graphs of several words, it appears that this method does, in fact, seem prone to finding the senses of a polysemous word rather than distinct meanings. For example, for the word *annual*, three clusters were found. All three of them related to the yearly occurrence meaning, and two focused on the financial aspects of this word, but none of the clusters contained words related to the flower meaning of *annual*. For the word *grate*, both clusters referred to the metal frame work of bars used to cover an opening, such as a storm grate or a fireplace grate. Neither of the two clusters contained words concerned with the *to irritate* meaning

of *grate*.

This may be caused by the limited number of words used to construct the semantic graph. Since only the 43 closest semantic associates were used in creating the graph, common words associated with a particular meaning may not have had a strong enough association with the target to be included in this graph. Increasing the number of associates used to construct the graph causes the number of vertices to become very large. For example, with $N = 43$, the graph may contain up to 1893 distinct words. If $N = 100$, the maximum order of the graph is $10,101$. As more computational power becomes available, this possibility can be further investigated.

8 Future Directions

Future work includes continuing to improve the quality of the semantic representations created using this method. A genetic programming package, such as NUANCE (Hollis & Westbury, 2003), will be used to further develop an understanding of the relationships between written frequency, lexical co-occurrence, and semantic association. Using this information, the influence of orthographic frequency in the representations created by this method will be further reduced, particularly in high frequency words. In addition, a much larger corpus, on the order of billions of words, will be analyzed. This will provide a wealth of co-occurrence data and increase the number of words for which sufficient data is available to construct representations.

This method will also be modified to allow it to better differentiate between separate meanings of a word and the relatively subtle differences between senses of a word. As an example, consider the word *apple*. The algorithm found three clusters in the SN of this word: one corresponding to the fruit meaning of the word, and two corresponding to the corporation. The two business oriented clusters focus on different aspects of the corporation's activity. One is concerned with the products they produce, and the other focuses on the financial workings of the company. These clusters overlap by 50%, but are not similar enough to be merged by the algorithm. One possibility is to use two thresholds to determine when clusters are merged. The

first value, $\mathsf{MINOVERLAP_a}$ can be set to a lower value to allow for the more aggressive merging strategy required to separate words into meanings. Next, a second threshold, $\mathsf{MINOVERLAP_p}$, can be set to a higher value, allowing for more clusters to differentiate between different senses of the same meaning.

Other methods of constructing and clustering the semantic graph for each word will be investigated. As more resources become available, much larger graphs can be constructed. This can be done by increasing the number of items returned in the query sets or increasing the depth of the search. The latter method will cause the upper bound on the number of vertices in the graph to grow to $N^3 + N^2 + N + 1$. If $N = 43$, the graph can contain up to $81,400$ vertices. Since clustering must be performed thousands of times in the GA used to select parameters, using graphs of this order is infeasible. With more computational power, the effects of graph size on the quality of the ambiguity measurements can be investigated.

Using the data produced by our method, a semantic processing unit will be implemented. This unit will most likely be based on a connectionist framework (Rumelhart, McClelland, & The PDP Research Group, 1986; McClelland, Rumelhart, & The PDP Research Group, 1986), and I hope to be able to train this unit to perform well in many semantic tasks, such as living-nonliving judgments, property verification, and category inclusion judgments. This unit can then be integrated into a full model of word recognition.

9 Summary

In this thesis, I have presented a new method for evaluating semantic association by analyzing lexical co-occurrence in a large corpus. Using these associations, vector-based representations of the semantic characteristics of each distinct word in the corpus were constructed. It was demonstrated that these representations contain only minimal influence from orthographic frequency, and that this influence is strongest in high frequency words. The vectors created by this method contain categorical information, and the semantic associations between words are intuitive.

By using graph theoretic clustering techniques, the SN for each word was divided into several groups of words related to the target through a common meaning, and the orthographic frequency of the items in these groups was used to estimate the frequency of each meaning of an ambiguous word. Based on these data, an ambiguity measure was calculated for each distinct word in the corpus. It was shown that these measurements are independent of frequency and are able to predict RT in a LD task. Analysis of the ambiguity measurement revealed that this method better distinguished between the senses of a polysemous word than between the distinct meanings of a homograph.

As a final note, it is important to point out a subtle but crucial difference between our results and those found in Casey (2005). In Casey's work, the window weights were optimized to best predict RT, that is, to minimize the correlation between the data produced by his method and experimental RT data. Thus, the data was constructed in such a manner that, by its very nature, required it to predict RT in LD. In the current method, the data were optimized to best match established ambiguity norms, a variable that has previously been shown to affect RT in LD. Although it was never attempted to alter the data to best predict RT, the data produced did so in a way that is consistent with recent literature investigating the effects of polysemy in word recognition. This method was able to extract values for a psychologically relevant variable that was consistent with previous norms and predicted RT in LD without any prior knowledge of experimental RT values.

This use of polysemy measures for optimization is a significant improvement over Casey et al's use of lexical decision RT for two reasons. First, polysemy is a well known semantic measure and the goal is to create a measure that is semantic in nature. Second, by using RT from lexical decision as the optimization metric, the final product is rendered ineffective as a tool in examining semantic effects in lexical decision: It would not be a surprise to find that the measure correlates with lexical decision RT, it was developed to do just that. The neighbourhoods from Casey et al. are interesting and useful in many ways but this potential source of circularity makes them less than ideal for psycholinguistic research. This method provides a

more widely useful measure that is mathematicaly linked to a known semantic value.

10 References

Aksoy, S., & Haralick, R. M. (1999a). A graph-theoretic approach to image database retrieval. In *Visual information and information systems* (p. 341-348).

Aksoy, S., & Haralick, R. M. (1999b). Graph-theoretic clustering for image grouping and retrieval. In *IEEE conference on computer vision and pattern recognition* (Vol. 1, p. 63-68).

Andrews, S. (1992). Frequency and neighborhood effects on lexical access: Lexical similarity or orthographic redundancy? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *18*(2), 234-254.

Augustson, J. G., & Minker, J. (1970). An analysis of some graph theoretical cluster techniques. *J. ACM*, *17*(4), 571–588.

Azuma, T., & Van Orden, G. C. (1997). Why SAFE is better than FAST: The relatedness of a word's meaning affect lexical decision times. *Journal of Memory and Language*, *36*, 484-504.

Baayer, R. H., Piepenbrock, R., & Van Rijn, H. (1993). *The CELEX lexical database.*

Balota, D. A., Cortese, M. J., & Pilotti, M. (1999). Item-level analysis of lexical decision performance: Results from a mega-study. In *Abstracts of the 40th annual meeting of the Psychonomics Society* (p. 44). Los Angeles, CA: Psyconomic Society.

Buchanan, L., Brown, N. R., Cabeza, R., & Maitson, C. (1999). False memories and semantic lexicon arrangement. *Brain and Language*, *68*, 172-177.

Buchanan, L., & Westbury, C. (2000). *Wordmine database: Probabilistic values for all four to seven letter words in the English language.* `http://www.wordmine.org/`.

Buchanan, L., Westbury, C., & Burgess, C. (2001). Characterizing semantic space: Neighborhood effects in word recognition. *Psychonometric Bulletin & Review*, *8*(3), 531-544.

Burgess, C. (1998). From simple associations to the building blocks of language: Modeling meaning in memory with the HAL model. *Behavior Research Medhos, Instruments, & Computers*, *30*, 188-198.

Burgess, C., & Livesay, K. (1998). The effect of corpus size in predicting reaction time in a basic word recognitin task: Moving on from Kucera and Francis. *Behavior Research Methods, Instruments, & Computers*, *30*, 272-277.

Burgess, C., Livesay, K., & Lund, K. (1998). Explorations in context space: Words, sentences, discourse. *Discourse Processes*, *25*, 211-257.

Burgess, C., & Lund, K. (1997). Modelling parsing constraints with high-dimensional context space. *Language & Cognitive Processes*, *12*, 177-210.

Burns, G., Daoud, R., & Vaigl, J. (1994). LAM: An Open Cluster Environment for MPI. In *Proceedings of supercomputing symposium* (pp. 379–386).

Casey, J. (2005). *The mathematical structure of semantic distances in language analysis*. Unpublished master's thesis, University Of Windsor.

Chartrand, G. (1985). *Introductory graph theory*. New York: Dover.

Chwilla, D. J., Hagoort, P., & Brown, C. M. (1998). The mechanism underlying pack-ward priming in a lexical decision task: Spreading activation versus semantic matching. *Quarterly Journal Of Experimental Psychology*, *51A*, 531-560.

Collins, A. M., & Loftus, E. F. (1975). A spreading-activation theory of semantic processing. *Psychological Review*, *82*, 407-428.

Collins, A. M., & Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, *8*, 240-247.

Coltheart, M., Davelaar, E., Jonasson, J., & Besner, D. (1977). Access to the internal lexicon. In S. Dornic (Ed.), *Attention and performance VI* (p. 535-555). Hillsdale, NJ: Lawrence Erblaum Associates.

Coltheart, M., Rastle, K., Perry, C., & Zeigler, J. (2001). DRC dual route cascaded model of visual word recognition and reading aloud. *Psychological Review*, *108*(1), 204-256.

Darwin, C. (1859). *On the origin of species by means of natural selection*. London: John Murray.

Diestel, R. (2000). *Graph theory* (2 ed.). New York: Springer-Verlag.

Durda, K., Casey, J., Buchanan, L., & Caron, R. (under review). CATSCAN. *Mental Lexicon*.

Gibbons, A. (1985). *Algorithmic graph theory*. Cambridge University Press.

Grainger, J., & Jacobs, A. M. (1996). Orthographic processing in visual word recognition: A multiple read-out model. *Psychological Review*, *103*(3), 518-565.

Hino, Y., & Lupker, S. J. (1996). Effects of polysemy in lexical decision and naming: An alternative to lexical access accounts. *Journal Of Experimental Psychology: Human Perception and Performance*, *22*(6), 1331-1356.

Holland, J. H. (1992). *Adaptation in natural and artificial systems*. MIT Press.

Hollis, G., & Westbury, C. (2003). *NUANCE: Naturalistic University of Alberta Non-linear Correlation Explorer*. `http://www.ualberta.ca/~hollis/nuance.html`.

Karp, R. M. (1972). Reducibility among combinatorial problems. In R. E. Miller & J. W. Thatcher (Eds.), *Complexity of computer computations* (p. 85-103). Plenum Press.

Klepousniotou, E. (2002). The processing of lexical ambiguity: Homonymy and polysemy in the mental lexicon. *Brain and Language*, *81*, 205-223.

Koriat, A. (1981). Semantic facilitation in lexical decision as a function of prime-target association. *Memory & Cognition*, *9*, 587-598.

Kruskal, J. B. (1978). *Multidimensional scaling*. Beveryl Hills, California: Sage Publications.

Kucera, H., & Francis, W. N. (1967). *Computational analysis of present-day American English*. Brown University Press.

Kumar, S. (1968). Semantic clustering of index terms. *J. ACM*, *15*(4), 493–513.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, *104*(2), 211-240.

Lemaire, B., & Denhière, G. (2004). Incremental construction of an associative network from a corpus. In K. Forbus, D. Gentner, & T. Regier (Eds.), *Proceedings*

26th annual meeting of the congnitive science society (p. 825-830). Chicago.

Lichacz, F. M., Herdman, C. M., Lefevre, J., & Baird, B. (1999, June). Polysemy effects in word naming. *Canadian Journal Of Experimental Psychology*, *53*(2), 189-193.

Luce, P. A., Pisoni, D. B., & Goldinger, S. D. (1990). Similarity neighborhoods of spoken words. In G. T. M. Altmann (Ed.), *Cognitive models of speech processing: Psycholinguistic and computational perspectives* (p. 122-147). Cambridge, MA: MIT Press.

Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behaviour Research Methods, Instruments, & Computers*, *28*(2), 203-208.

Marslen-Wilson, W. D. (1987). Functional parallelism in spoken word recognition. *Cognition*, *25*, 1987.

McClelland, J. L., & Elman, J. L. (1986). The TRACE model of speech perception. *Cognitive Psychology*, *18*(1), 1-86.

McClelland, J. L., & Rumelhart, D. E. (1981, September). An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review*, *88*(5), 375-407.

McClelland, J. L., Rumelhart, D. E., & The PDP Research Group. (1986). *Parallel distributed processing: explorations in the microstructure of congnition* (Vol. 2). Cambridge, Mass.: MIT Press.

McNamara, T. P., & Holbrook, J. B. (2003). Semantic memory and priming. In A. F. Healey & R. W. Proctor (Eds.), *Handbook of psychology vol. 4: Experimental psychology* (p. 447-474). John Wiley & Sons, Inc.

McRae, K., de Sa, V. R., & Seidenberg, M. S. (1997). On the nature and scope of featural representations of word meaning. *Journal Of Experimental Psychology*, *126*(2), 99-130.

Meyer, D. E., Schvandeveldt, R. W., & Ruddy, M. G. (1975). Loci of contextual effects on visual word-recognition. In P. M. A. Rabbitt & S. Dornic (Eds.), *Attention and performance V* (p. 98-118). London: Academic Press.

Nelson, D. L., McEvoy, C. L., & Schreiber, T. A. (1998). *The University of South Florida word association, rhyme, and word fragment norms.* `http://w3.usf.edu/FreeAssociation`.

Peereman, R., & Content, A. (1995). Neighborhood size effect in naming: Lexical activation or sublexical correspondences? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *21*(2), 409-421.

Peereman, R., & Content, A. (1997). Orthographic and phonological neighborhoods in naming: Not all neighbors are equally influential in orthographic space. *Journal of Memory and Language*, *37*, 382-410.

Pexman, P. M., & Lupker, S. J. (1999, December). Ambiguity and visual word recognition: Can feedback explain both homophone and polysemy effects. *Canadian Journal of Experimental Psychology*, *53*(4), 323-334.

Ratcliff, R., Gomez, P., & McKoo, G. (2004). A diffusion model account of the lexical decision task. *Psychological Review*, *111*(1), 159-182.

Rodd, J., Gaskell, G., & Marslen-Wilson, W. (2002). Making sense of semantic ambiguity: Semantic competition in lexical access. *Journal of Memory and Language*, *46*, 245-266.

Rodd, J. M. (2004). When do leotards get their spots? semantic activation of lexical neighbors in visual word recognition. *Psychonometric Bulletin & Review*, *11*(3), 434-439.

Rodd, J. M., Gaskell, M. G., & Marslen-Wilson, W. D. (2004). Modelling the effects of semantic ambiguity in word recognition. *Cognitive Science*, *28*, 89-104.

Rumelhart, D. E., & McClelland, J. L. (1982). An interactive activation model of context effects in letter perception: Part 2. The contextual enhancement effect and some tests and extentions of the model. *Psychological Review*, *89*(1), 60-94.

Rumelhart, D. E., McClelland, J. L., & The PDP Research Group. (1986). *Parallel distributed processing: explorations in the microstructure of congnition* (Vol. 1). Cambridge, Mass.: MIT Press.

Sears, C. R., Hino, Y., & Lupker, S. J. (1995). Neighborhood size and neighborhood frequency effects in word recognition. *Journal of Experimental Psychology: Hu-*

man Perception and Performance, *21*(4), 876-900.

Seidenberg, M. S., & McClelland, J. L. (1989). A distributed, developmental model of word recognition and naming. *Psychological Review*, *96*(4), 523-568.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*, 379-423, 623-656.

Shapiro, L. G., & Haralick, R. M. (1979). Decomposition of two-dimensional shapes by graph-theoretic clustering. In *IEEE transactions on pattern analysis and machine intelligence* (Vol. 1, p. 10-20).

Squyres, J. M., & Lumsdaine, A. (2003, September / October). A Component Architecture for LAM/MPI. In *Proceedings, 10th european pvm/mpi users' group meeting* (p. 379-387). Venice, Italy: Springer-Verlag.

Thompson-Schill, S. L., Kurtz, K. J., & Gabrieli, J. D. E. (1998). Effects of semantic and associative relatedness on automatic priming. *Journal of Memory and Language*, *38*, 440-458.

Tulving, E. (1972). Episodic and semantic memory. In E. Tulving & W. Donadlson (Eds.), *Organization of memory* (pp. 381–403). New York: Academic Press.

Twilley, L. C., Dixon, P., Taylor, D., & Clark, K. (1994). University of Alberta norms of relative meaning frequency for 566 homographs. *Memory & Cognition*, *22*(1), 111-126.

Westbury, C., Buchanan, L., & Brown, N. R. (2002). Sounds of the neighborhood: False memories and the structure of the phonological lexicon. *Journal of Memory and Language*, *46*, 622-651.

Wikipedia. (2005). *Information entropy — Wikipedia, the free encyclopedia.* ([Online; accessed December 12, 2005])

Wordsmyth. (1999). *Wordsmyth: on-line dictionary and thesaurus.* `http://www.wordsmyth.net/`. ([Online; accessed February 1, 1999])

A Complexity and $\mathcal{NP}$-Complete Problems

The *time complexity* of an algorithm is the number of steps required by the algorithm to solve the problem as a function of the size of the data provided as input to the algorithm. This is typically measured by counting the number of operations the algorithm requires to process the input. Here, the term *operations* is vague and should be defined to include only those operations which are most relevant to the performance of the algorithm being analyzed. When analyzing an algorithm that sorts a list of items, for example, the number of comparisons between items may be counted. For an algorithm that multiplies two matrices, the number of addition and multiplication operations may most strongly affect the running time.

Definition A.1 Let $g\colon \mathbb{N} \to \mathbb{R}^+$, where $\mathbb{R}^+$ denotes the positive real numbers. Then $O(g)$ is the set of all functions $f\colon \mathbb{N} \to \mathbb{R}^+$ for which there exists some $c \in \mathbb{R}, c > 0$ and some $N \in \mathbb{N}$ such that $f(n) \leq cg(n)$ for all $n \geq N$.

The set $O(g)$ is called *big oh of gee* of just *oh of gee*, and contains all functions which are bounded above by g. If the number of operations required by an algorithm is $f(n)$, then we say that the algorithm has complexity $O(f(n))$. When analyzing the complexity of an algorithm, it is common practice to include only the highest-order term and ignore any constants. Thus, instead of describing the complexity of an algorithm as $O(3n^2 + 2n + 1)$, it is simply written as $O(n^2)$. This allows for the classification of algorithms into broad categories based on their *asymptotic growth rate* or *order*.

Figure 23 shows the relative growth rates for several common orders of functions. As is clear from this figure, some functions grow much faster than others as the size of the input increases. The function 2^n grows very quickly. If an algorithm is order $O(2^n)$ a very large number of operations are required to solve the problem for even moderately sized inputs. As the size of the input grows the time requirements of the algorithm become extremely high and for large inputs, the algorithm may take days, months, or even years to complete. Problems for which all known algorithms require immense amounts of computational time are considered *intractable*. As a general rule

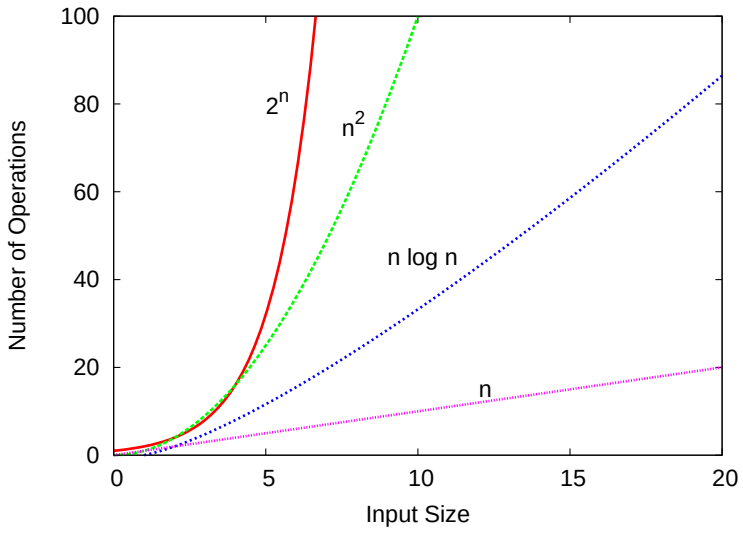


Figure 23: Relative growth rates of 2^n, n^2, $n \lg n$ and n.

of thumb, any problem that can be solved in polynomial time (i.e., there exists an algorithm with complexity $O(n^k)$ for some constant $k > 0$) or faster is considered tractable.

Definition A.2 An algorithm is said to be *polynomially bounded* if its complexity is $O(n^k)$ for some fixed $k > 0$. A problem is said to be polynomially bounded if there exists a polynomially bounded algorithm to solve the problem.

Many optimization problems can be formulated as *decision problems*. A decision problem consists of a problem description and a specific input to the problem. The only possible answers to a decision problem are *yes* or *no*. As an example, consider the traveling salesperson problem (TSP):

> Given a set of n cities and the costs of traveling between each pair of cities, find the minimum cost of traveling to each of the n cities exactly once and returning to the starting city.

This problem is formulated as an optimization problem. A decision problem version of the TSP is:

Given a set of n cities, the costs of traveling between each pair of cities, and a positive real number k, is there a way to travel to each of the n cities exactly once, returning to the starting city, with a total cost of at most k?

$\mathcal{P}$ and $\mathcal{NP}$ are two classes of decision problems. $\mathcal{P}$ is the class of all decisions problems that are polynomially bounded. Inclusion in $\mathcal{P}$ does not guarantee that a problem has a reasonable efficient solution, but all problems with efficient solutions are contained in $\mathcal{P}$. If a problem is *not* in $\mathcal{P}$, then the problem is extremely difficult to solve, and most likely will be impossible to solve in practice.

The class $\mathcal{NP}$ is more difficult to describe. Consider the task of verifying a potential solution to a problem. A potential solution is referred to as a *certificate*. In the decision problem version of the TSP, a certificate would consist of a permutation of the cities to be visited. A certificate can be verified by the following steps:

1. Check that each city is visited exactly once.
2. Check that the ending city is the same as the starting city.
3. Check that the total cost of travel is less than k.

Clearly, and algorithm for verifying a solution to the TSP that uses these three steps is polynomially bounded. If the solution meets all requirements of the problem, the algorithm returns a *yes* answer. Otherwise, the algorithm may either return *no* or enter an infinite loop and provide no output.

Definition A.3 A *nondeterministic algorithm* is an algorithm with two phases:

1. Create a random solution to the problem.
2. Determine if the random solution satisfies the problem. If it does, output *yes*. Otherwise, there is no output.

When a deterministic algorithm is run multiple times on the same input, it produces the same output. A nondeterministic algorithm, however, may produce different (or no) output on each execution. The number of operations required by a

nondeterministic algorithm is the sum of the number of operations required to produce a random solution, plus the number of operations required to verify the random solution. A nondeterministic algorithm is polynomially bounded if there exists a polynomial p such that for every input of size n for which a correct solution exists, there is an execution of the algorithm that completes in fewer than $p(n)$ operations.

Definition A.4 $\mathcal{NP}$ is the class of all decision problems for which a polynomially bounded nondeterministic algorithm exists.

Clearly, $\mathcal{P} \subseteq \mathcal{NP}$, since the verification stage for any polynomially bounded decision problem may simply produce a correct solution in polynomial time, then output *yes*. An open problem in theoretical computer science is whether or not $\mathcal{P} = \mathcal{NP}$. Unfortunately, no one has yet shown that any single problem in $\mathcal{NP}$ is not also in $\mathcal{P}$. That is, while there are no known polynomially bounded solutions for any problem in $\mathcal{NP}$, none of these problems have been shown to have a lower bound on their time complexity that is larger than polynomial.

Next consider the task of converting the input to one problem to a valid input to another problem.

Definition A.5 Let **P** and **Q** be two decision problems, and let T be a map from the input set of **P** to the input set of **Q**. T is called a *polynomial reduction* from **P** to **Q** if the following three conditions are satisfied:

1. T is polynomially bounded.

2. For any input x to **P**, if x produces a *yes* output for **P**, then $T(x)$ produces a *yes* output for **Q**.

3. For any input x to **P**, if x produces a *no* output for **P**, then $T(x)$ produces a *no* output for **Q**.

If there exists a polynomial reduction from **P** to **Q**, then **P** is said to be *polynomially reducible* to **Q**, denoted $\mathbf{P} \leq_P \mathbf{Q}$. Note that if $\mathbf{P} \leq_P \mathbf{Q}$ and $\mathbf{Q} \in \mathcal{P}$, then $\mathbf{P} \in \mathcal{P}$.

Definition A.6 A decision problem $\mathbf{Q}$ is said to be $\mathcal{NP}$-*hard* if $\mathbf{P} \leq_P \mathbf{Q}$ for every $\mathbf{P} \in \mathcal{NP}$.

If $\mathbf{Q}$ is an $\mathcal{NP}$-hard problem, then it must be at least as difficult as any other problem in $\mathcal{NP}$. This provides a lower bound on the complexity of $\mathbf{Q}$. Note that a problem may be $\mathcal{NP}$-hard and not be in $\mathcal{NP}$ since it must only be as hard as any other problem in $\mathcal{NP}$, but there are no stipulations on the complexity or existence of an algorithm that solves the problem. Inclusion in $\mathcal{NP}$ provides an upper bound on a problem, since a nondeterministic polynomially bounded algorithm must exist to solve the problem.

Definition A.7 If $\mathbf{Q} \in \mathcal{NP}$ and $\mathbf{Q}$ is $\mathcal{NP}$-hard, then $\mathbf{Q}$ is called $\mathcal{NP}$-*complete*.

Karp (1972) showed that the decision version of many optimization problems, including finding maximal cliques in a graph, are $\mathcal{NP}$-complete. From this definition, and the fact that the class $\mathcal{P}$ is closed under the operation of polynomial reduction, follows an important result:

$$\text{If } \mathbf{Q} \text{ is } \mathcal{NP}\text{-complete and } \mathbf{Q} \in \mathcal{P}, \text{ then } \mathcal{NP} = \mathcal{P}.$$

This result displays the value of finding an $\mathcal{NP}$-complete problem that is polynomially bounded. Unfortunately, such solutions have been sought for several problems in $\mathcal{NP}$ without success. Most researchers believe that polynomially bounded solutions to $\mathcal{NP}$-complete problems do not exist. Unfortunately, at this point, it is still unknown whether or not $\mathcal{NP} = \mathcal{P}$, and it is considered computationally difficult to find exact solutions to any $\mathcal{NP}$-complete, as no known polynomially bounded algorithm to solve any of these problems has been found.

Vita Auctoris

Name:	Kevin Durda
Place of Birth:	Windsor, Ontario
Date Of Birth:	December 6, 1979
Education:	University Of Windsor Windsor, Ontario 1998 - 2002 B.C.S.
	University Of Windsor Windsor, Ontario 2002 - 2003 B.M.H.
	University of Windsor Windsor, Ontario 2003 - 2006 M.Sc