Behavior Research Methods
2010, 42 (2), 393-413
doi:10.3758/BRM.42.2.393

Exploring lexical co-occurrence space using HiDEx

Cyrus Shaoul and Chris Westbury
University of Alberta, Edmonton, Alberta, Canada

Hyperspace analog to language (HAL) is a high-dimensional model of semantic space that uses the global co-occurrence frequency of words in a large corpus of text as the basis for a representation of semantic memory. In the original HAL model, many parameters were set without any a priori rationale. We have created and publicly released a computer application, the High Dimensional Explorer (HiDEx), that makes it possible to systematically alter the values of these parameters to examine their effect on the co-occurrence matrix that instantiates the model. We took an empirical approach to understanding the influence of the parameters on the measures produced by the models, looking at how well matrices derived with different parameters could predict human reaction times in lexical decision and semantic decision tasks. New parameter sets give us measures of semantic density that improve the model's ability to predict behavioral measures. Implications for such models are discussed.

This work investigates a class of models of lexical semantics derived from the hyperspace analog to language (HAL; Burgess, 1998; Burgess & Lund, 2000), a computational model of word meaning that derives semantic relationships from lexical co-occurrence. Although the original HAL model was well specified, it contains several parameters whose values were set without formal or empirical justification. We have created a novel and freely available implementation of the HAL model—called High Dimensional Explorer (HiDEx)—that allows users to systematically vary those parameters, creating a class of models that are algorithmically identical but parameterized differently. Given the absence of any a priori formal justifications for parameter values in HAL, we have elected to assess different parameter settings by how well they perform at predicting human behavioral data on two lexical access tasks: lexical decision and semantic decision. In this article we explain how HiDEx works and how we were able to use it to explore HAL's parameter space.

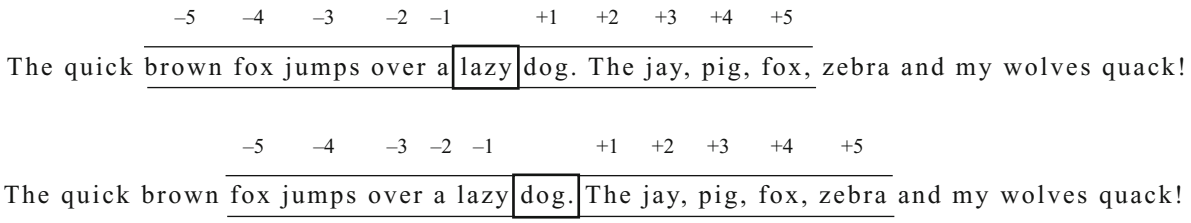
We begin with a brief overview of the HAL class of models and some related models. HAL uses word co-occurrence to build a vector space that contains contextual information for every word in a specified dictionary. A vector space is a geometric representation of data that has an ordered set of N numbers associated with each point in an N-dimensional space. Each such set of numbers defines the point's location in the space and is called its *vector*. HAL space is made up of vectors with one dimension for each word in the language. In the original HAL work, these word vectors had more than 100,000 dimensions.

Each point in a word's vector is a weighted count of the number of times another word co-occurs with that word in a corpus of text. Words can co-occur when they are adjacent or when they are separated by a small number of intervening words. The maximum distance between words considered to co-occur is called the *window size*. Window size is one of the free parameters in the HAL model. In the original model, words were considered to have co-occurred if they occurred within 10 words of each other in either direction.

Words in another word's co-occurrence window are weighted according to their proximity to that word, using a weighting function. The original HAL model used a linear weighting function called a *linear ramp* as a multiplier to give more weight to the words that co-occurred closer to the center of the window. Words that occurred on either side of the center word of the window were assigned 10 co-occurrence points. Each word's outside neighbors were assigned 9 co-occurrence points, and so on, down to a single point for a word that occurred 10 words away from the center word. This weighting function is another free parameter in HAL that has no a priori justification and can be changed in HiDEx.

Lexical memories in the HAL model are built by making the model read words in text one window at a time and then sliding the window forward one word. This process of counting local co-occurrences is illustrated in Figure 1. After reading a whole corpus and counting the local co-occurrences, the data are stored in a raw co-occurrence matrix containing the frequencies of co-occurrence for all possible combinations of words in all possible positions in the window. This matrix can become a very large set of numbers. For example, with a 100,000-word lexicon and HAL's 10-word window in each direction, the number of data points in the matrix would

C. Shaoul, cyrus.shaoul@ualberta.ca



AHEAD	brown	fox	jumps	over	a	lazy	dog	the	jay	pig	zebra
lazy	0	1	0	0	0	0	5	4	3	2	0
dog	0	2	0	0	0	0	0	5	4	3	1

BEHIND	brown	fox	jumps	over	a	lazy	dog	the	jay	pig	zebra
lazy	1	2	3	4	5	0	0	0	0	0	0
dog	0	1	2	3	4	5	0	0	0	0	0

Figure 1. A visualization of a sliding 5A5B (five words ahead, five words behind) window as it moves over a sentence. The tables show what the co-occurrence matrix would contain after HAL-style weighting of the counts from the sliding window (but before normalizing the rows).

be 100,000 target words × 100,000 co-ocurrence words × 20 positions = 200 billion entries.

The raw co-occurrence matrix is large but very sparse (containing mostly zeros) because most words never co-occur with each other. To do any meaningful work with the data, they must be condensed into a more compact form in the consolidation phase of the HAL model. One step in this consolidation, or *aggregation*, is done by simply summing the word frequencies in the window (see Figure 1). With a 10-word window size, this aggregation reduces the data set to one tenth of its former size, since a word's co-occurrence represented by 20 cells (10 forward and 10 backward) is represented by just two cells (1 forward and 1 backward) after summing.

These vectors are not yet usable, due to the influence of orthographic frequency. A small number of words have much higher orthographic frequencies than do the majority of words (Baayen, 2001; Zipf, 1935, 1949), and, consequently, those words will have very high co-occurrence frequencies. Due to this bias, high-frequency words will have vectors that are very dense with large values; therefore, they will be much closer in context space to all words than low-frequency words will be. The original HAL model dealt with this frequency issue by normalizing each vector—that is, by dividing each element in the vector by the vector's length. Normalizing vectors in this particular way leads to a systematic frequency bias (Durda & Buchanan, 2008; Shaoul & Westbury, 2006b) and is not used in our implementation of the model. We instead divide the vectors by the frequency of each word, normalizing the vectors and removing the effect of frequency.

The final stage in preparing the vectors for distance calculations is the elimination of the sparser, less informative parts of the matrix. In the original HAL model, this was done by retaining vectors just for the words with the greatest column variances (i.e., eliminating words that co-occur very often or very rarely with the target word). If only the columns with the top 10,000 most variant words are used, the forward and backward aggregates create columns of 20,000 elements instead of 200,000 elements. This matrix is smaller and denser than previous matrices and is small enough to fit into the memory of modern computers, making the calculations tractable.

The HAL model uses the Euclidean distance metric to calculate the distance between any two words in the space. For every element j in the vectors for words a and b, the Euclidean distance is

$$\sqrt{\sum_j \left(a_j - b_j\right)^2}.$$

This distance expresses how similar the contexts of usage of the two words are. If the words have similar values in the same dimensions, they will be closer together in the space.

In order to find the neighbors of a word in context space, we need to calculate the distance between the word and all the other words in the language. The closest words selected according to some criterion are considered neighbors in HAL space. Neighborhood density is a measure of how tightly packed the words in the neighborhood are. The density measure in the original HAL work was calculated by averaging the distances between the word and its 10 closest neighbors (Buchanan, Westbury, & Burgess, 2001). This produced a density value for each word in a co-occurrence space. The use of a fixed number of neighbors for calculating density is not ideal for two related reasons. One reason is that, if the density distribution of neighbors around a given word is not uniform (and there are many reasons to believe it will not be), then averaging

a uniform number of neighbors is not equivalent in different words. A word with many neighbors will probably have more neighbors that are close to it than a word with few neighbors, but the word with many neighbors may nevertheless have a larger average neighborhood size. By analogy, a person with many friends is more likely to have a few extremely short friends than a person with fewer friends, since extremely short people are uncommon. However, the person with many friends may nevertheless have a larger average friend height than the person with fewer friends. A second problem with using a radius with 10 words is a familiar one by now: This cutoff point is arbitrary. No one knows whether the average distance of a word's closest 10 neighbors is a better measure of co-occurrence density than the average distance of its closest 12 or 18 or 56 neighbors. It is possible that the density measure is sensitive to the number of closest neighbors that are averaged together. We will discuss solutions to these problems below.

Empirical Studies of HAL

HAL was originally put forth as a model of semantic memory. It was soon subject to scrutiny to see how it performed as a model of human semantic memory. Buchanan, Burgess, and Lund (1996) used HAL to model deep dyslexia. They found that words with denser neighborhoods produced more errors in deep dyslexics than did words with sparser neighborhoods. Buchanan et al. (2001) looked at HAL neighborhood effects on lexical decisions. They found that the HAL neighborhood size was a reliable predictor of lexical decision reaction time (LDRT). Even after removing the contributions of orthographic variables and imageability, there was significant explanatory power from HAL neighborhood size. Siakaluk, Buchanan, and Westbury (2003) investigated the ability of HAL to predict performance in a categorization task. They found that HAL semantic density influenced the decision time on a go/no-go semantic decision task. Words with denser co-occurrence neighborhoods were processed faster. Yates, Locker, and Simpson (2003) found a similar facilitatory effect of high-density neighborhoods in a lexical decision task that included pseudohomophone foils.

Song and Bruza (2001), Song, Bruza, Huang, and Lau (2003), and Song, Bruza, and Cole (2004) have applied the HAL model to problems of concept learning, inference, and information flow. They were able to use HAL vectors as part of an intelligent software agent that makes "aboutness" judgments. For example, the sentence, "Welcome to the City of Red Deer, Alberta," has nothing to do with the ungulate known as *Cervus elaphus*. They are able to make such judgments by combining the vectors for all of the words in the sentence and then comparing that combination to the vector for the concept in question (in this case, "deer").

The performance on language tasks of some conceptually related models has also been studied. Rohde, Gonnerman, and Plaut (2005) created the correlated occurrence analogue to lexical semantic (COALS) model. It is identical in design to HAL, except in the following respects: It uses a correlation operation for both vector normalization and similarity measures, it removes closed-class words from the model, and it uses singular value decomposition (SVD) to reduce the dimensionality of the co-occurrence matrix. SVD is a factorization technique that can be used to calculate a lower dimensionality approximation of the original, larger matrix. Rohde et al. showed that HAL performs very well on word-similarity tasks, such as those in the TOEFL exam and other, similar tests when SVD is applied to the model.

Bullinaria and Levy (2007) analyzed different influences of excluded closed-class words, corpus size, window size, and distance metrics. They proposed using an information-theoretic metric, pointwise mutual information (PMI), instead of Euclidean distance and found that PMI improved the accuracy of their model in their semantic task simulations. PMI is a measure of association that is calculated as the ratio between the probability of two words co-occurring, given their joint distribution, versus the probability of their co-occurrence, given only their individual distributions and assuming independence.

The most recent HAL-type model to be reported is one created by Durda and Buchanan (2008) called "Windsor improved norms of distance and similarity of representations of semantics" (WINDSORS). In this version of HAL, the influence of word frequency on the model's output is mitigated through the use of various statistical and mathematical methods. Despite the removal of any real correlation with orthographic frequency, the vectors produced by WINDSORS are capable of modeling semantic priming experiments and word similarity norms. Using WINDSORS as a starting point, Durda, Buchanan, and Caron (2009) took a first step toward building a relationship between the vectors in vector models and perceptual/motor features. They used a feedforward neural network to provide a mapping from co-occurrence vectors of concepts (such as BIRD) to feature norms (such as HAS-WINGS). They then generalized this mapping and produced a list of features from the co-occurrence vector of a novel concept. This result supports the notion that co-occurrence vectors contain enormous amounts of information about the words they represent, including perceptually grounded information.

Recently Jones, Kintsch, and Mewhort (2006) and Jones and Mewhort (2007) built a holographic model of lexical memory, more complex than HAL, that they call "bound encoding of the aggregate language environment" (BEAGLE). It encodes the co-occurrence and word-order information into vectors, using a convolution function as a way to model verbal associative memory (Murdock, 1982). Convolution is a mathematical operation that can be applied to any type of vector to encode it into a memory-trace vector. Later, the information can be extracted from the memory trace by calculating the correlation between a probe item and the combined memory trace. In BEAGLE, this function is applied to language in such a way that word-order information and global co-occurrence information are simultaneously encoded into each vector. BEAGLE has been used to account for many different types of semantic priming effects when the prime–target pairs are related by both pure semantic relationships and associations (Jones et al., 2006). It has

also been used to model sentence completion and semantic categorization (Jones & Mewhort, 2007).

HAL's Parameter Space

Lexicon choice. The lexicon we chose was derived from the CELEX database (Baayen, Piepenbrock, & Gulikers, 1995) by choosing all of the words that had an orthographic frequency of 2 occurrences per million or greater. This lexicon contains approximately 45,000 words, which is less that the 70,000-word lexicon used by Lund and Burgess (1996). The choice to reduce the lexicon size was made for two reasons: (1) The amount of information contained in the contexts of low-frequency words is small and does not have much influence on the distances between most words in the space, and (2) the computational complexity of the model increases greatly with the size of the lexicon.

Corpus choice. Lund and Burgess (1996) used a corpus of 160 million words of USENET (Fristrup, 1994) text. It is well known that the balance of registers and genres in a corpus has a strong effect on the HAL vectors produced (Bullinaria & Levy, 2007; Rohde et al., 2005; Shaoul & Westbury, 2006b). In order to make our results comparable to the majority of studies done on the HAL model, we chose to replicate as closely as possible the USENET corpora used by Lund and Burgess (1996), Burgess and Lund (1997), Burgess (1998), Burgess and Livesay (1998), Burgess, Livesay, and Lund (1998), and Burgess and Lund (2000). We collected 12 billion words of USENET text from 2005 to 2007 (Shaoul & Westbury, 2009) and used a 1-billion-word subset of this corpus to build our models. The same benefits described by Lund and Burgess (1996) are true for this corpus: USENET text contains a very broad variety of genres and topics, and most of the text is in a very conversational style, similar in some ways to spoken language. We chose not to use a corpus of 160 million words because we found that there were many words in our 50,000-word lexicon that had one occurrence or no occurrences in this corpus. To obtain observations of multiple occurrences of all the words in our lexicon, it was necessary to use a larger corpus. In addition, Bullinaria and Levy did a very thorough analysis of the impact of corpus size on HAL. They found that their measures of performance increased as corpus size increased, but the amount of improvement was mostly at ceiling for corpora of 90 million words or greater. Recchia and Jones (2009) found that the amount of data was more important than the type of algorithm used when the quality of the corpus was held constant, with the larger corpus producing better performance. These results led us to believe that our choice of corpus source and size would allow us to compare them to those in previous work with USENET corpora of equal or smaller size.

Frequency issues and normalization. Shaoul and Westbury (2006b) showed that there was a problem with the original HAL model that allowed a word's orthographic frequency (OF) to influence its neighborhood density. If HAL neighborhood density is used to predict psycholinguistic phenomena, it would be unfortunate if HAL density measures covaried with OF, one of the most powerful predictors of lexical access (Balota, Black, & Cheney, 1992). Shaoul and Westbury (2006b) found that the normalization procedure used in the original HAL—dividing each vector by its variance—did not eliminate frequency effects. Buchanan et al. (2001) proposed using the OF of each word as the word-vector's divisor, and Shaoul and Westbury (2006b) did exactly that. For words with high frequency, co-occurrence values shrank, and for words with low frequency, co-occurrence values were amplified. Shaoul and Westbury (2006b) found that the neighborhood densities made with this new normalization technique were no longer correlated with OF.

Weighting and window size. As we mentioned above, Lund and Burgess (1996) assigned weights to the co-occurrence counts by weighting them in a symmetric 10-word window (10 words in front of and behind the target word) with a linear ramp, which multiplied values by their distance from the end of the window. This meant that the count for the word appearing directly adjacent would be multiplied by 10, the next one out would be multiplied by 9, and so on. There was no a priori justification for the window size, its symmetry, or the linear-ramp weighting scheme, and there was a very limited exploration of the window-size parameter. We chose to vary the window size and weighting scheme simultaneously to empirically investigate the impact of window size on the model. We chose to introduce alternative weighting schemes that would reflect the variety of weighting schemes being used by other investigators who have worked with HAL models. The weighting schemes that we tested are listed in Table 1.

Table 1
List of Weighting Functions Implemented in HiDEx

Function Name	Weighting Function	Sample Vector of Weights
Flat weights	$x = 1$	[1 1 1 1 1 1 1 1]
Linear ramp	$x = (w - p + 1)$	[1 2 3 4 4 3 2 1]
Quadratic ramp	$x = (w - p + 1)^2$	[1 4 9 16 16 9 4 1]
Forward linear ramp, backward flat ramp	$x = 1, x = (w - p + 1)$	[1 1 1 1 4 3 2 1]
Forward flat weights, backward linear ramp	$x = (w - p + 1), x = 1$	[1 2 3 4 1 1 1 1]
Inverse linear ramp	$x = p$	[4 3 2 1 1 2 3 4]
Inverse quadratic ramp	$x = p^2$	[16 9 4 1 1 4 9 16]
Second-word weighting	if $p = 2, x = 10$, else $x = 1$	[1 1 10 1 1 10 1 1]
Third-word weighting	if $p = 3, x = 10$, else $x = 1$	[1 10 1 1 1 1 10 1]
Fourth-word weighting	if $p = 4, x = 10$, else $x = 1$	[10 1 1 1 1 1 1 10]

Note—Function w, window size; p, position (1 to w).

Context size. One key aspect of implementing the HAL model is the reduction of the size of the global co-occurrence matrix after the weighting scheme has been applied and the windows have been summed. The original HAL model sorted all the vectors by variance and retained only the most variant word vectors (Lund & Burgess, 1996). Shaoul and Westbury (2006b) chose to use the most frequent word's vectors rather than the most variant.

Neighborhood size and neighborhood membership threshold. Another extension to HAL proposed by Shaoul and Westbury (2006b) was an extension of the concept of a neighborhood membership threshold. Unlike HAL, which used a fixed number of the closest neighbors as the neighborhood, we calculated a distance in co-occurrence space, called the *membership threshold*, which was used as the criterion for neighborhood membership. This threshold is calculated by randomly sampling many millions (usually billions) of word pairs and calculating their interword distances to find the mean and standard deviation of this distance distribution. The neighborhood membership threshold was set to 1.5 *SD*s below the mean distance, which is about 6.7% of the average distance between any two words. Most word pairs have only a weak or no relationship. Setting the cutoff point to cover this small fraction of the average distance between words will ensure that words count as neighbors only if they are at least as close as the closest 6.7% of the billions of sampled pairs. A consequence of using this definition of neighborhood membership is that some words may have more neighbors than others, and some words may have no neighbors at all. We will use this thresholded neighborhood measure for all future neighborhood calculations. Note that this threshold has to be recalculated every time any other parameter in the model is changed, since the average distance between words will be affected by any parameter change.

Two new measures: ARC and NCOUNT. Shaoul and Westbury (2006b) introduced two new measures of semantic density that depend on this neighborhood threshold. The first measure, average radius of co-occurrence (ARC), is the mean of the distances between the target word and all the neighbors that fall within its threshold. The second measure, neighbor count (NCOUNT), is the number of neighbor words within that threshold. These two new measures relate information about the density of the context neighborhood of a word (see Figure 2). In later sections, we will be doing analyses of a value called NCOUNT-INV, which is defined as the reciprocal of NCOUNT+1, the 1 being added to allow the measure to be well defined for words that have zero neighbors. NCOUNT-INV has a value of 1 for words with no neighbors and smaller values for words with more neighbors.

HiDEx

As of yet, there has been no implementation of HAL made available at no cost to investigators. In 2004, we began developing our implementation of HAL, and we have used it to conduct research on the HAL model (Shaoul & Westbury, 2006b). We released this software as an open-source project in December 2008, under the GNU General Public License (Stallman, 2009), and the program is available at www.psych.ualberta.ca/~westburylab/downloads.html.

HiDEx is capable of executing the HAL model using the identical calculations that were specified in the work of Lund and Burgess (1996), but it is also able to use different algorithms and parameter sets by setting various options. These alternative algorithms include new normalization algorithms, new weighting algorithms, and new neighborhood membership algorithms. Alternative parameters include new window sizes and context sizes. In this section, we describe HiDEx in detail, show what kind of input it can use, and explain what kinds of processing it can do and what kinds of output it can produce.

HiDEx is a command-line program written in C++ that is configured using a text file containing all the settings

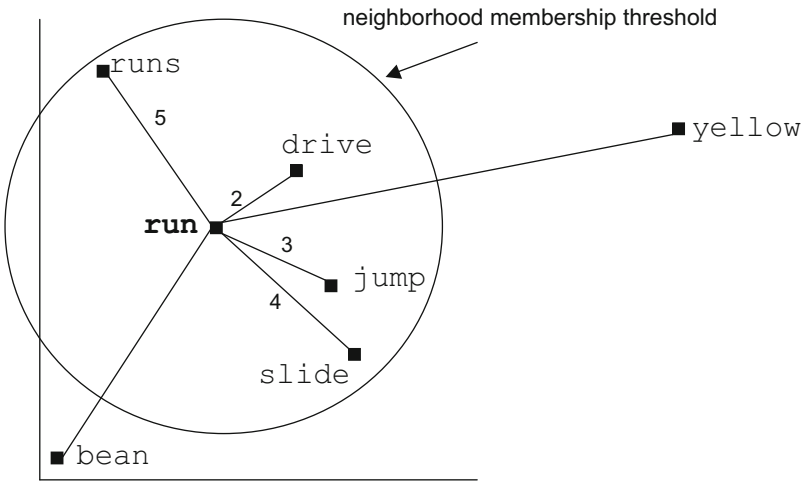


Figure 2. A 2-D visualization of the neighborhood membership threshold. The word *run* in this example has four neighbors, giving *run* a neighbor count of 4. The average distance between *run* and its neighbors, the average radius of co-occurrence measure, is 10.25.

used by the program. The format of this configuration file is described in the HiDEx user manual, which is distributed with the software (Shaoul & Westbury, 2008).

There are three stages of processing when using HiDEx. In Stage 1, one obtains and prepares the corpus for use, and then processes the corpus with HiDEx to create the global co-occurrence matrix. In Stage 2, one configures the desired settings for all the available parameters and processes the global matrix with HiDEx to create word vectors. In Stage 3, one uses HiDEx to measure the contextual similarity between words and then saves the output.

Here, in greater detail, are the operations completed in each stage. During Stage 1, HiDEx performs the initial preprocessing of the corpus. HiDEx will accept corpora in any UTF-8 encoded language. The corpus should be in the form of one large file that includes user-defined document separators. HiDEx will automatically capitalize lowercase letters in the corpus and remove any punctuation (preserving the English apostrophe). Once the corpus is processed, HiDEx stores a large, sparse matrix that contains all co-occurrence information for every word in the lexicon. This sparse matrix is saved to disk for use in Stage 2.

During Stage 2, the sparse matrix is processed into a denser matrix. HiDEx can be configured to use any of the following four settings during Stage 2. (1) The sizes of the windows ahead and behind the target word are set independently. (2) The weighting scheme is set. There are nine possible weighting schemes that can be applied to the co-occurrence window. The nine weighting functions available are described in Table 1. (3) The number of dimensions to retain (the context size) is set. When set to a value *N*, only the *N* most frequent word's co-occurrence data are included in the final calculations in Stage 3. (4) The vector normalization method can be set to Default: Ratio (co-occurrence divided by target-word frequency; Shaoul & Westbury, 2006b); PPMI (positive pointwise mutual information; Bullinaria & Levy, 2007); or Correlation (Rohde et al., 2005).

In Stage 3, the word vectors are used to calculate context similarity. The similarity metrics that are available for use are Default: (inverse) Euclidean distance (Lund & Burgess, 1996); Cosine (Bullinaria & Levy, 2007); City Block (Rohde et al., 2005); and Correlation (Rohde et al., 2005).

Two types of measurements can be made with HiDEx. (1) Get a word's neighbors: Find the *N* most similar words using the similarity metric specified. There is also a threshold-based option that allows the number of neighbors to vary due to the density of the neighborhood. If this similarity threshold is used, some words will have many neighbors, and others will have none. (2) Measure the similarity between pairs of words: Given a list of word pairs, the similarity of the contexts of these two words is calculated and returned. This can be done for any number of word pairs.

One of the attractions of co-occurrence models like HAL is that they are simple to understand and implement. However, notwithstanding its simplicity, the algorithm that generates HAL is computationally expensive. To provide a perspective on the scale of calculations performed by HiDEx, in this section, we provide a broad, slightly simplified explanation of the process.

At the beginning of a set of HiDEx experiments, we must obtain a corpus. In our case, we obtained a large amount of freely available USENET text (Shaoul & Westbury, 2009). Stage 1, the process of building a local co-occurrence data set, currently takes approximately 4 days of continuous processing on a large computer (with 16 CPUs and 128GB RAM). The actual calculations involve counting the words that co-occur within the window size for every word in the corpus and incrementing the corresponding values in the co-occurrence matrix. For a matrix the size of ours, this will create a data set of approximately 63 GB. This data set is the raw co-occurrence data set from which a HAL matrix of word vectors can be computed for a given HAL parameter set.

Using a computer with multiple cores, HiDEx can process a corpus of many billions of words. It can be configured to use lexicons that have well over 50,000 terms in them and to compute neighborhoods from vectors in a matter of hours. This performance is achieved by symmetric parallel processing that is implemented by using OpenMP (Chapman, Jost, van der Pas, & Kuck, 2007).

In Stage 2, we use HiDEx to build a HAL model with a given set of parameters. First, we apply the window size and weighting scheme to a raw co-occurrence data set, consolidating it and creating the global co-occurrence vectors. The dimensionality of this matrix is then reduced by retaining only the vectors that meet our criterion for inclusion: the *N* vectors for words with the highest OF. These vectors are normalized and then used to calculate the neighborhood membership threshold. We calculate this from a sample of a small percentage of all possible pairwise distances, usually around 2 billion distances. To measure how many words fall inside a word's membership threshold, it is necessary to compute the distance between that target word and every other word in the lexicon. The process of calculating neighborhoods takes approximately 2 h per parameter set for a word list of about 500 words.

As is now clear, the sheer number of calculations required to run these models is enormous. It is because the numbers of these calculations are so large that it is rare to find the resources to explore HAL's parameter space. HiDEx has been designed to take advantage of multiCPU supercomputers with large amounts of memory to enable this exploration.

Exploring Parameter Space With HiDEx

As we have noted above, the parameters used in the original HAL work were chosen without empirical or theoretical justification. The questions we want to address are, Is the model sensitive to these parameters, with respect to its ability to predict human performance on psycholinguistic tasks? Is there a new set of parameters that will create a better model of word meaning? Will this new parameter set give HAL-type models more explanatory power?

One problem for this kind of work is that the parameter space of the HAL model is very large, even if we limit ourselves to a small number of the infinite possible weighting schemes and keep the window size fairly small. The parameters we described above (corpus types, corpus sizes, window sizes, weighting functions, context sizes, etc.) can

be combined in a huge number of ways. As our summary of related work above shows, to evaluate the set of parameter combinations remains computationally intractable on the supercomputers currently available to us. We have, therefore, been able to undertake only a coarse-grained traversal of the parameter space. We then used an automated fitness function to measure how well each model fit a large set of experimental data. We varied only two parameters at a time, holding all other parameters constant at their default HAL values. This strategy allowed us to find out how these two parameters influenced the model individually and how they interacted in a pairwise fashion. Although this research is undeniably exploratory in nature, in constitutes a necessary set of preliminary steps toward being able to justify parameter values used in HAL-influenced co-occurrence. The two parameters that were varied in this initial exploration of parameter space were window size and weighting function. We chose these two parameters because they are the two HAL parameters that have the most potential to change the contextual information stored about words.

The size of the window is the only parameter that can change which words are considered to share context. Smaller windows will prevent long-distance contextual relationships from forming. For example, consider the following sentence:

> Mathematics, rightly viewed, possesses not only truth, but supreme beauty—a beauty cold and austere, without appeal to any part of our weaker nature, without the gorgeous trappings of painting or music, yet sublimely pure, and capable of a stern perfection such as only the greatest art can show. (Russell, 1910, p. 73)

Were the window size less than nine words ahead, there would not be any co-occurrence trace from this sentence for the words "mathematics" and "beauty," and HAL would never be able to convey that these words are related. The weighting scheme that is used to aggregate local co-occurrence across the co-occurrence window has a slightly different influence on the model's structure. By emphasizing the contextual importance of different parts of the window, it can boost or shrink the influence of proximally co-occurring words. In the above quote, the co-occurrence frequency of "painting" and "music" would be weighted by a factor of 10, if we used the second-word scheme, or by a factor of 2, if we used the inverse-ramp scheme (see Table 1). The difference in weighting can significantly change the distance between a target word and other words in context space, altering the neighborhood and the neighborhood density.

Some work comparing different model parameter sets has been done by Bullinaria and Levy (2007) and Lifchitz, Jhean-Larose, and Denhière (2009). Bullinaria and Levy used four measures to compare the fitness of their models: (1) the score on the TOEFL test (using HAL to choose the one word as the correct answer in a multiple-choice exam), (2) a distance comparison test (comparing inter-word distances between known, semantically related pairs and random pairs), (3) a semantic category test (testing to see whether words are closer to the name of their category than to the names of other categories), and (4) a syntactic categorization test (testing to see whether a word is closer to its syntactic category center than to other syntactic category centers). These tests provide some information about how the model performs in capturing the structure of the human semantic space. These tests depend on handpicked word lists, and, for that reason, they might not generalize beyond the test stimuli.

Bullinaria and Levy (2007) varied many different parameters to their HAL-like model to see which parameter settings performed best on each of their four tests of fitness. Their search of parameter space was fairly exhaustive, varying each parameter alone or in combination with another parameter. The first parameter varied was the type of co-occurrence frequency measure. The positive pointwise mutual information (PPMI) measure performed better than all the other measures they tested. The second parameter they looked at was the similarity metric, and they found that the cosine measure (the cosine of the two vectors) performed better than the Euclidean distance metric, the city block metric, and all of the other metrics they compared. The third parameter they looked at was window size, and they found that a symmetrical window of one word ahead and one word behind produced the highest accuracy. On the basis of these results, Bullinaria and Levy proposed that the optimum parameters of the HAL model are window size = 1, weighting scheme = flat, similarity measure = cosine, and co-occurrence measure = PPMI. This is very different from the original HAL parameters used by Burgess (1998) (window size = 10, weighting scheme = linear ramp, similarity measure = euclidean distance, co-occurrence measure = raw frequency).

More recently, Lifchitz et al. (2009) explored the parameter space of LSA (Landauer & Dumais, 1997) to find the optimal dimensionality of their model across various corpora of different sizes. They found that their optimal tuning of lemmatization, stop-word lists, term weighting, pseudodocuments, and normalization of document vectors in LSA allowed their model to outperform seventh- and eighth-grade students on a multiple-choice biology test. Both of these studies (Bullinaria & Levy, 2007; Lifchitz et al., 2009) show how exploring the parameter space of a high-dimensional model can lead to new insights and unexpected optimal-parameter sets.

In this study, we chose to focus our exploration on the HAL parameters that have the most relevance to our theoretical interests: window size and window weighting. To explore the influence of these parameters, we created a list of all of the possible combinations of forward and backward window sizes (0, 5, or 10) and all of the weighting functions listed in Table 1. This list contained 73 sets of parameter combinations. In Experiment 1, we looked at how varying these four parameters influences the model's ability to predict mean LDRT for a large set of randomly chosen words. In Experiment 2, we attempted to model semantic decision performance for a set of English words.

EXPERIMENT 1
Predicting Visual LDRTs

We chose estimation of LDRTs as our measure of each parameter set's goodness of fit. LDRTs have two main advantages that make them a suitable choice for this role. Most important, it is widely accepted that LDRTs are sensitive to many well-defined lexical measures (for reviews, see Balota et al., 1992; Hollis, Westbury, & Peterson, 2006), including semantic measures that are most closely related to our co-occurrence measures.[1] Second, LDRTs have the advantage of being available in large numbers collected in a systematic way (Balota et al., 2002), allowing us to simulate large experiments with random sampling to avoid the problem of over-fitting our model to a small and/or constant data set.

Method

Our first fitness function was the correlation between words' ARCs and their average LDRTs. We obtained averaged LDRT data for 40,481 words (averaged for each word across participants) and used these data to run simulations of lexical decision experiments.

In each simulation, a random subset of 500 words was sampled from the 40,481 in the list. HiDEx was then used to compute the ARC and NCOUNT-INV for these 500 words using one set of parameters. Our fitness measure for each parameter set was the correlation of the 500 words' measures computed using those parameters with the 500 LDRTs. This process was repeated 73 times with different random lists of 500 words and all the parameter sets.

Results and Discussion

One important question is, To what degree is HAL sensitive to changes in these two parameters? We found that it was highly sensitive. There was a large amount of variation in the correlations between LDRT and ARC and NCOUNT-INV for the sets of parameters we tested. Across all 73 parameter sets, the mean squared correlation of LDRTs with ARC was .14 (SD = .04), with a range from .04 to .25. The mean squared correlation of LDRTs with NCOUNT-INV was .17 (SD = .035), with a range from .08 to .24. Figure 3 shows the average ARC and N COUNT correlations for different weighting functions and window types. Because this figure would be too busy if it were included, the variance in these correlations is graphed separately. Figures 4B and 4D, respectively, show the variance in correlations between LDRT and ARC for different window sizes and weighting functions. Figures 4A and 4C, respectively, present the same information for NCOUNT-INV. These figures show that there is great variance in the predictive value of ARC and N COUNT for different weighting schemes, across window sizes, and for the different window sizes across weighting schemes.

Another important question is, Are the original HAL parameters optimal choices by this measure? We will refer to windows using the notation "xByA," where x is the number of included words behind the target word and y is the number of included words in front of the target word.

The original HAL parameters were 10B10A,[2] with a linear-ramp weighting function. The squared correlation of LDRT with the ARCs produced by this set was .11. The best combination of parameters for ARC predicting LDRT was inverse quadratic 10B0A, which accounted for more than twice as much variance, with a squared correlation of .26.

The squared correlation of LDRT with the NCOUNT-INV values produced by the original HAL parameters was .14. The optimal value found among the parameter sets searched was inverse quadratic 0B10A, with a squared correlation of .25 between the NCOUNT-INV values and the LDRTs.

The fact that these parameter sets were not identical complicates the problem of choosing an optimal set. Both used the inverse-quadratic weighting scheme, which weights each word as the square of the distance in words from the target word. Across all window sizes, this weighting scheme is the fourth best at predicting LDRTs from ARCs (Figure 4C), and it also shows by far the least variance across all weighting schemes. It is the fifth best at predicting LDRTs from NCOUNT-INV (Figure 4D). In the case of both predictors, the R^2 difference between the best weighting scheme and the inverse-quadratic weighting scheme is small, less than .03.

We also note that, in both cases, the weighting schemes that performed slightly better included the inverse-linear weighting scheme, which weights each word linearly with its distance in words from the target word. Taken together, these observations offer some support for schemes that weight words distant from the target word more highly than close words, which is the opposite of the original HAL linear-ramp parameter. This may be due to the ability of the inverse ramp to minimize the weight given to words that are directly adjacent to the target word. When the target word is a noun or verb, nearby words may often be closed-class function words. By deemphasizing the minimal semantics of closed-class words, inverse weighting schemes may improve the categorical relationships between words. We examine this possibility in more detail below.

As noted above, the two parameter sets that produced factors most predictive of LDRT used different window sizes: 10B0A (with ARC as the predictor) and 0B10A (with NCOUNT-INV as the predictor). As shown in Figures 4A and 4B, the results with these windows sizes were the second-best predictors of LDRT after collapsing across all weighting schemes. In both cases, the window size that produced the best predictors across all weight schemes was the original HAL weighting, 10B10A; in both cases, the difference between the best and second-best predictors was very small, with an average R^2 difference of less than .02.

It is impossible to reconcile the finding that the optimal windows were the nonoverlapping 0B10A and 10B0A windows or to reconcile these findings with the earlier finding by Bullinaria and Levy (2007) of an optimal window size of 1A1B. We are, therefore, unable to draw any firm conclusions about the optimal window size. It may be that different window sizes are optimal for different tasks, which raises difficult questions for theorists who wish to put forward co-occurrence models as models of human processing. Since both the 0B10A and 10B0A windows showed only small differences in the cases of both of our predictors from the best average predictor (10A10B), we

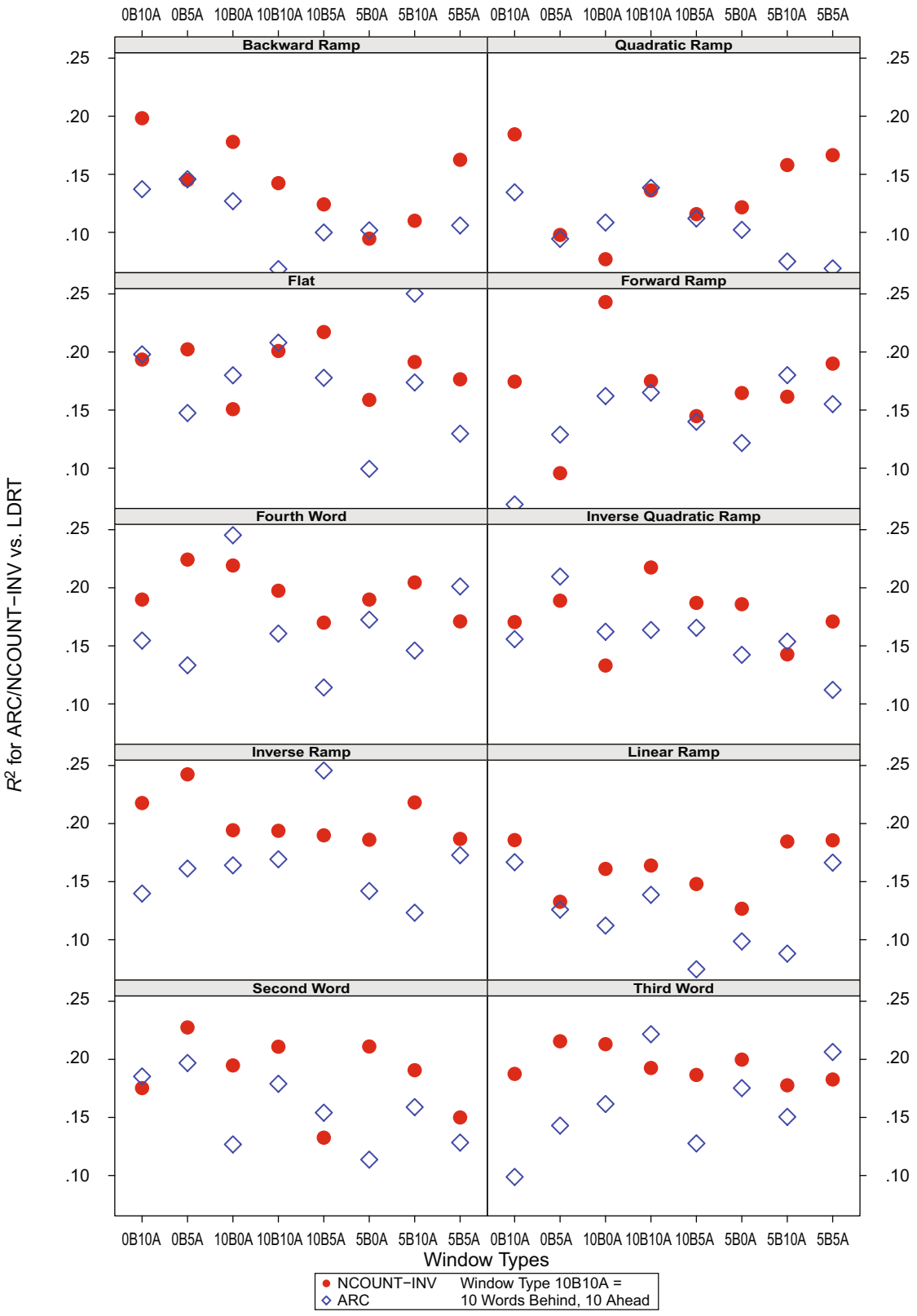


Figure 3. Overview of R^2 of the inverted neighbor count (NCOUNT-INV) and average radius of co-occurrence (ARC), with lexical decision reaction times (LDRTs), for different weighting functions and window types. Circles represent NCOUNT-INV correlations, and diamonds represent ARC correlations. All correlations are significant ($p < .001$).

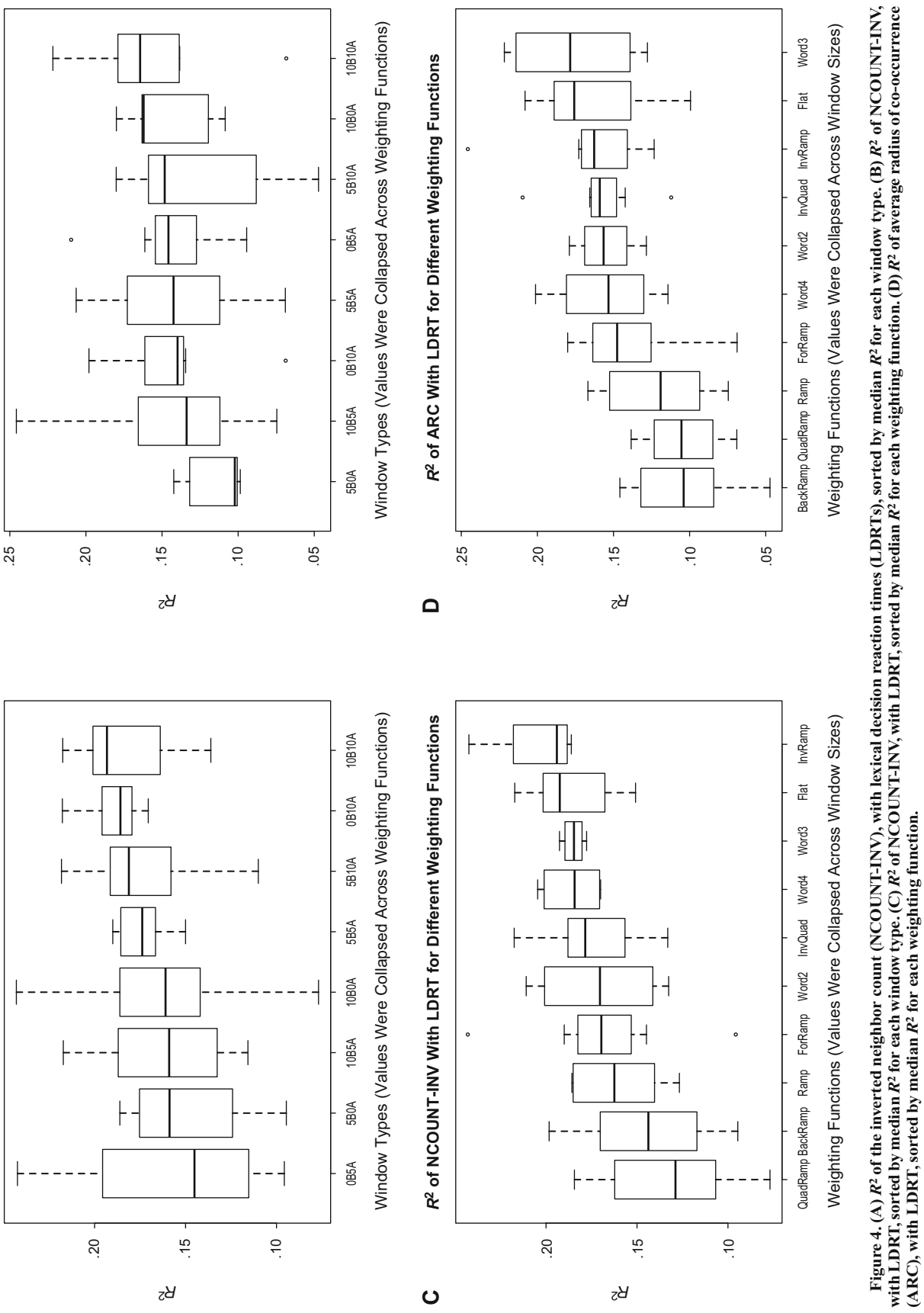


Figure 4. (A) R^2 of the inverted neighbor count (NCOUNT-INV), with lexical decision reaction times (LDRTs), sorted by median R^2 for each window type. (B) R^2 of NCOUNT-INV, with LDRT, sorted by median R^2 for each window type. (C) R^2 of NCOUNT-INV, with LDRT, sorted by median R^2 for each weighting function. (D) R^2 of average radius of co-occurrence (ARC), with LDRT, sorted by median R^2 for each weighting function.

tentatively suggest as a purely operational decision that the best HiDEx window size among those we have examined here may be the HAL default, 10A10B. Future work with other, possibly dynamic weighting schemes may help clarify what role the window size plays in modeling human behavioral results.

Since we used a random sample of words in each of these simulations to demonstrate the generalizability of HAL to the whole lexicon, we cannot directly compare the correlations that we have calculated to test for statistical significance. In order to confirm the increase in correlation for our parameter sets, we chose a random subset of 5,000 words from the lexicon and calculated the neighborhoods for these words using two parameter sets of interest: the original HAL parameter set and one of the best of our 73 parameter sets (inverse ramp, 10B10A).

The results of this comparison are shown in Table 2. The difference between the explanatory power of the two models can be seen in the size of the Akaike information criterion (AIC) value (Akaike, 1974). There is a difference of more than 100 between the AIC scores for the ARC and NCOUNT-INV models, meaning that the LDRT data are millions of times more likely, given the predictors from the optimized HAL model.[3]

To avoid any contamination of the results by spurious correlations, and to validate the results we obtained, we had to be certain that our results were stable across different subsets of the lexicon. We measured the stability of these correlations across two different sets of words. The same sets of parameters were re-run with different random sets of 500 words. The average absolute difference in R^2 between runs was small: .04 for the predictor ARC and .03 for the predictor NCOUNT-INV. This small amount of difference between runs shows that the correlations were stable across different random samples of words.

The goal of this experiment was to explore the HAL parameter space and find the set of parameters that produced ARC and NCOUNT-INV measures that best predicted a human measure of lexical accessibility. We found that there were large differences between the 73 parameter sets we tested. The best parameter set produced measures that can explain about 15% more of the variance in LDRTs than the worst, which is an encouraging result.

One constant across all of the results was that the slopes produced by a linear regression of both ARC and NCOUNT-INV on LDRT were consistently greater than zero: That is, the denser the neighborhood or the larger the number of neighbors, the faster the RT. This facilitatory effect was reported by Buchanan et al. (2001). In their Experiment 4, Buchanan et al. (2001) used a factorial design (words with dense neighborhoods vs. words with sparse neighborhoods) to investigate the relationship between neighborhood density and LDRT. They found that words with denser semantic neighborhoods had faster RTs in a lexical decision task.

EXPERIMENT 2
Predicting Semantic Decision RTs

Experiment 1 used LDRTs for the reasons noted above; they are widely used as a measure of lexical access and are available in large quantities. However, co-occurrence measures are often considered to be an analogue to semantic measures, and LD requires a shallow depth of semantic retrieval and processing. In the next two experiments, we replicated the work above using a semantic decision task that we hoped might be more sensitive to the representation and organization of lexical semantic memory.

We collected our own semantic decision reaction time (SDRT) data from two semantic decision tasks (described in more detail below), in which we asked participants to decide whether or how closely two words were semantically related. There has been no previous research on how the co-occurrence models might be used to predict RTs on this task. We proposed the following HAL measures as potential candidates that might be predictive of SDRTs.

1. ARCs and NCOUNTs: These are the ARC and NCOUNT measures for each word in the pair of words. From previous work with LDRTs (Buchanan et al., 2001; Shaoul & Westbury, 2006b), we know that words with sparser neighborhoods show faster LDRTs if all other lexical factors are held equal. This implies that the ARC or NCOUNT of either word could influence SDRTs.

2. ARCSUM and NCOUNTSUM: These are the sums of the ARCs and NCOUNTs for the words that make up the pair. The summed ARC and NCOUNT capture the combined densities of the words in the pair

3. Interword distance: This is the distance between the two words in co-occurrence space. This measure may influence RTs by facilitating the retrieval of lexical semantics due to the priming effects of the retrieval of two words with similar contextual histories.

We devised two experiments to gather data relating to the semantic processing of words and cognitive load of a semantic decision task. Experiment 2A was a speeded forced choice semantic decision task that required participants to decide whether words were related or unrelated. Experiment 2B was a judgment task in which participants

Table 2
Comparison of Correlations Between Lexical Decision Reaction Time and ARC/NCOUNT-INV for the Original HAL Parameters and an Optimized Set of Parameters

Parameter Set	Weighting Function	Window Size	R^2_{ARC}	AIC (ARC)	$R^2_{NCOUNT\text{-}INV}$	AIC (NCOUNT)
Original HAL	Linear ramp	10B10A	.12	61,660	.17	61,349
Optimized HAL	Inverse ramp	10B10A	.15	61,440	.18	61,231

Note—AIC, Akaike information criterion; ARC, average radius of co-occurrence; HAL, hyperspace analog to language; NCOUNT-INV, inverse of neighbor count (having a value of 1 for words with no neighbors and smaller values for words with more neighbors).

were asked to rate how related two words were. We used the same 73 parameter sets as in Experiment 1 to calculate the seven predictors described above and tested the strength of the relationships between these predictors and the mean SDRT of each item.

Method

Participants. Sixty-four undergraduate students (37 female) enrolled in introductory psychology courses at the University of Alberta participated in this study for partial course credit. Their mean age was 19.4 years, and the standard deviation was 4.4 years.

Stimuli. Three hundred pairs of words in three sets were chosen as stimuli. One hundred pairs were listed as associates in the University of South Florida's Nelson association norms database (Nelson, McEvoy, & Schreiber, 1998). Another 100 were from the idiosyncratic (low-frequency) responses list from the Nelson database. The third 100 word pairs were not listed as being related in the Nelson database.

To ensure that our stimuli would cover a broad range of associative relationships and not reflect the influence of nonsemantic lexical variables, we selected our stimuli very carefully. We built a very large set of pairs, from which we chose smaller stimuli sets using a criterion of noncorrelation. Two large sets of word pairs were chosen from the full lists of word pairs from the Nelson associated norms database. We started with the full list of 69,000 associated pairs (hereafter, ASSOC) and the full list of 112,000 idiosyncratic responses (hereafter, IDIO). The third large set was a list of 200,000 word pairs that we generated ourselves by picking words randomly without replacement from a dictionary of English words (Shaoul & Westbury, 2006a) with a frequency greater than 10 words per million (hereafter, UNREL).

We measured the OF, orthographic neighborhood (ON), and number of letters (LEN) for all of the words in these 287,000 pairs. Using the default HAL parameters, we also calculated the interword distance in co-occurrence space. We then matched subsets of the three sets of word pairs so that, for each ASSOC word pair, there would be an IDIO and an UNREL pair matched for OF, ON, LEN, and HAL distance. The matching algorithm first converted all measures into standard scores and then calculated the Euclidian distance of these standardized scores between all pairs. The pairs with the smallest distance in *z*-score space were stored as a match and were immediately removed from the input lists. By this method, we created three lists of approximately 1,000 entries each. From these lists, we picked the final 300 pairs by selecting pairs that satisfied the following criteria: The UNREL pairs were not judged by either author to be semantically related, and the HAL distance between all pairs was distributed evenly across the range of interword distance values.

Finally, we split the 300 pairs into two equal sets of 150 pairs to counterbalance the order of presentation. Equal numbers of participants saw each of the sets in each of the experiments in each presentation order. Please see Appendixes A, B, and C for the full stimulus sets.

Procedure. In the relatedness decision task, stimuli were presented on an LCD display connected to a Macintosh computer (running Mac OS X ver. 10.3.9) using the presentation software ACTUATE (Westbury, 2007). All words were displayed in lowercase letters in the Times New Roman font. Participants were asked to make a judgment about two words that were to appear sequentially. The first word appeared in black at the top of a 500 × 500 pixel white square for a duration that varied randomly between 2,000 and 3,500 msec between trials. This period of time, 2,500 to 5,000 msec, was intended to provide sufficient time for the participants to read the first word and access its meaning. After this, at the bottom of this square, a crossbar ("+") appeared for a duration of 500 to 1,500 msec (again, varying randomly), to be replaced with the second word in the pair.

When the second word appeared, participants were requested to make the following semantic decision as quickly and as accurately as possible (as explained in the instructions): "In your opinion, are these two words related?" One of two keys on the computer keyboard ("X" for *no* and "M" for *yes*) was to be pressed, and the RT was measured.

In the relatedness rating task, participants were asked to perform a slightly different task using the same apparatus and software as described above. They were shown word pairs, with both words appearing simultaneously in Times New Roman font. They were asked to rate the relatedness of the words by using the mouse to drag a sliding marker on an undivided Likert scale on the screen. This scale had the word "unrelated" over the left end of the line and the words "highly related" over the right side of the line. The participants were asked to take as much time as needed to accurately rate each pair of words. The software measured the position of the marker on the line and recorded 0 for unrelated and 100 for highly related, as well as the time taken to do the rating.

Results and Discussion

We removed observations from Experiment 2A that had an RT less than 300 msec or greater than 4,500 msec (2 standard deviations from the mean). These outliers made up 1% of all of the observations. We did not consider RTs for ASSOC word pairs when the participant considered them unrelated or RTs for UNREL items when the participant considered them to be related.

The results for the semantic decision tasks are shown in Table 3. Most (81%) of the ASSOC pairs were judged to be related, a smaller majority (66%) of the IDIO pairs were judged to be related, and most (87%) of the UNREL pairs were judged to be unrelated. Similar results were found in the related rating task (Experiment 2B). ASSOC pairs had a higher rating ($M = 80$, $SD = 24$) than IDIO pairs ($M = 67$, $SD = 31$), and UNREL pairs had the lowest rating ($M = 21$, $SD = 26$). These results support our

Table 3
Distribution of the Responses for the 300 Word Pairs, With Descriptive Statistics

Category	Semantic Decision	Percentage of Observations	Semantic Decision Reaction Time	
			M	*SD*
Associated pairs	Related	81	1,006.2	466.6
	Unrelated	19	1,210.1	544.5
Idiosyncratic responses	Related	66	1,116.2	513.0
	Unrelated	34	1,257.5	578.8
Unrelated pairs	Related	13	1,426.3	725.9
	Unrelated	87	1,152.2	506.4

choice of stimuli, clearly showing that our participants' understanding of relatedness matched our categories.

Using HiDEx, we calculated all the predictor measures described above for all of the word pairs in the stimulus set, using the most promising 40 parameter sets from the 73 parameter sets used in Experiment 1. These were the parameter sets that included the following weighting schemes: linear ramp (baseline), inverse ramp, inverse quadratic, third word, and fourth word.

The only predictor measure that had a reliable correlation with SDRT was the semantic density for the first word, NCOUNT1-INV. As shown in Figure 5, the only two weighting functions to achieve statistically significant correlations were fourth word and inverse ramp. The most consistent window type was the 10B0A window. The best result overall was obtained from the combination of these two parameters: fourth-word weighting and 10B0A window ($r = -.14$, $R^2 = .02$, $p = .01$). Since the slope of this relationship is negative, SDRT is predicted to decrease as the semantic density around the first word decreases (i.e., as NCOUNT1-INV increases). This result is congruent with the results from Experiment 1 because the time required to access the meaning of the word pair was shorter for words that had sparser neighborhoods.

This relationship is much weaker than that for the LDRT data in Experiment 1. One possible reason for this is that the task we used in this experiment was very complex. From the presentation of the first word to the presentation of the second, the participants presumably accessed semantic information about the first word. When the second word was displayed, the RT captured the time it took to do at least two things: retrieve the semantic information about the second item, and make a semantic decision. The SDRT we captured should be a function of the lexical retrieval of

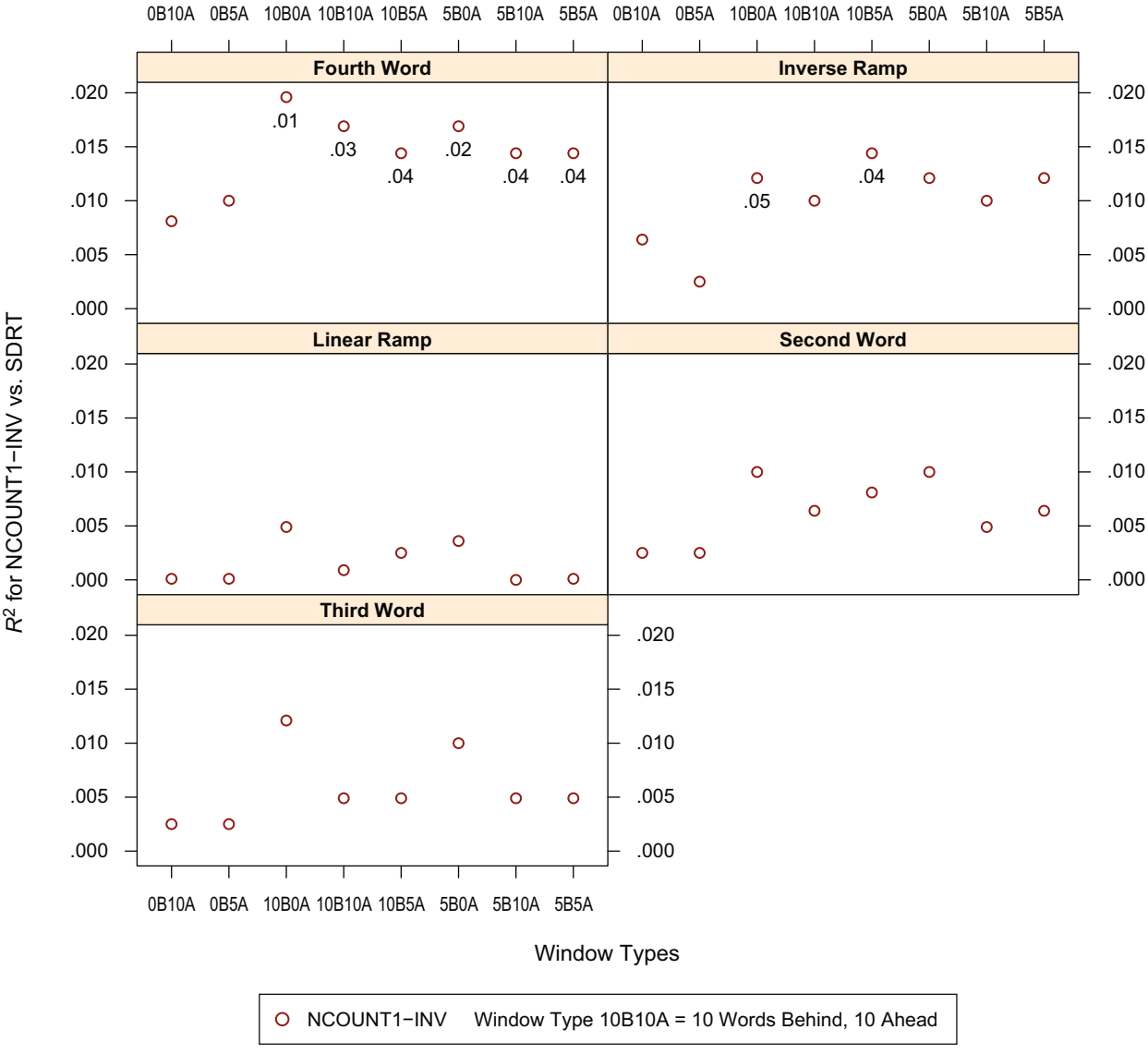


Figure 5. Overview of R^2 of the inverse neighbor count (NCOUNT1-INV), with semantic decision reaction times (SDRTs), for different weighting functions and window types. The *p* value for each reliable regression is shown below each point.

the second word, the complexity of the semantic memory traces for the two words, the type and number of relationships between the words, and the strategy/strategies that the participant used. We hypothesized that our HAL measures would provide an indirect measure of the complexity of the semantic traces and the relationship between the words (through their shared context). We have no data to help us predict which strategies would be used to make these semantic decisions. This opens the door for variability that we will not be able to account for in our model.

These concerns about the appropriateness of this forced choice task gave us the incentive to seek a better semantic decision task. This experiment and its results are described in the next section. We reserve further discussion of the findings of Experiment 2 for the General Discussion.

EXPERIMENT 3
A Go/No-Go Semantic Decision Experiment

In Experiment 3, we studied semantic decision using a slightly different task. We chose a task that Siakaluk et al. (2003) found to be superior in eliciting semantic distance effects: the go/no-go semantic decision task. Siakaluk et al. used both a forced choice and a go/no-go task in a semantic decision experiment (in their case, animacy: "Is this alive or not?"), and they noted that the time-constrained nature of the go/no-go task made it superior to other tasks for semantic decisions.

Method

Participants. Thirty-five undergraduate students (21 female) enrolled in introductory psychology courses at the University of Alberta participated in this study for course credit. Their mean age was 20.7 years, and the standard deviation was 2.9 years. None of the participants had participated in the previous experiments.

Stimuli. The stimuli used in this experiment were identical to those used in Experiment 2.

Procedure. The laboratory equipment was identical to that used in the semantic decision experiment described above. The only part of the procedure that was changed in Experiment 3 was the type of response we requested of the participants after the second word appeared on the screen. The participants were instructed to press the spacebar only if the words were related. The participants were instructed to do nothing if the words were unrelated. If no input was detected after 3,500 msec, the next trial was initiated, and a no-go result was recorded. Seventeen participants were shown 150 pairs, and 18 participants were shown the remaining 150 pairs. Order of presentation was randomized for each participant.

Results and Discussion

Two hundred sixty-three of the 300 word pairs were given at least one go response. There is a possibility that some participants pressed the spacebar hastily or unintentionally during the experiment. Our method for detecting unintended of responses was to look for the items that produced the fewest of responses. Any words with responses from less than 20% of the participants (equivalent to 3 or fewer participants) were removed. After removing observations for these 56 items (3% of the total number of observations), we analyzed the data for the remaining 207 word pairs.

The descriptive statistics for RTs are presented in Table 4. As with the forced choice task, the ASSOC RTs were the fastest, with the IDIO pairs having slower RTs and greater variability. Due to the nature of the task and to nonresponses, we were able to collect reliable RTs for the few erroneously accepted UNREL words only. These RTs were the longest and had the most variability.

To understand the relationship between the tasks in Experiments 2 and 3, we looked at the relationship for the mean SDRT for the 207 items that both experiments shared. There was a strong correlation between the logged RTs in the two experiments ($R^2 = .46$, $p < .001$). Between the two experiments, there was no significant difference between the mean SDRTs for the ASSOC stimuli [$t(198) = 1.195$, $p = .23$] or IDIO stimuli [$t(198) = 1.73$, $p = .09$]. However, there was a significant difference of 386 msec between the means for the UNREL stimuli in Experiment 2A and the means for the UNREL stimuli in Experiment 3 [$t(198) = 3.62$, $p < .001$], showing the predicted increased depth of processing for unrelated, difficult-to-process words.

In order to study the relationship between the SDRTs in Experiment 3 and the measures calculated by HiDEx, we calculated the item regressions for mean SDRT and our seven predictor measures for the 40 parameter sets that we analyzed in Experiment 2. The results are shown in Figure 6. There was no significant correlation between mean item SDRT and the majority of the parameter sets. In particular, the original HAL parameters (10B10A, with the linear-ramp weighting function) did not produce predictors that correlated reliably with go/no-go RTs. As in Experiment 2, the only measure that had any significant correlation with an SDRT was NCOUNT1-INV. The only two weighting functions that produced significant correlations with this measure were the inverse ramp and the fourth word functions, which also performed well in our analysis in Experiment 2. The window types involved in the parameter sets that produced predictors with reliable correlations with RT were 10B0A and 10B5A for the inverse ramp and 0B5A for the fourth word.

In summary, the results obtained in Experiment 3, like those obtained in Experiment 2, can be predicted by the HAL model. There were three parameter sets that had significant linear relationships between neighborhood density and mean SDRT, but the amount of variance explained by these relationships was small. The implications of these results are discussed in the following section.

Table 4
Distribution of Reaction Times for the 207 Word Pairs With Sufficient Numbers of "Go" Responses in Experiment 3

Category	Percentage of Observations	Semantic Decision Reaction Time	
		M	*SD*
Associated pairs	55	1,089.5	515.4
Idiosyncratic responses	43	1,249.6	573.8
Unrelated pairs	2	1,812.8	783.3

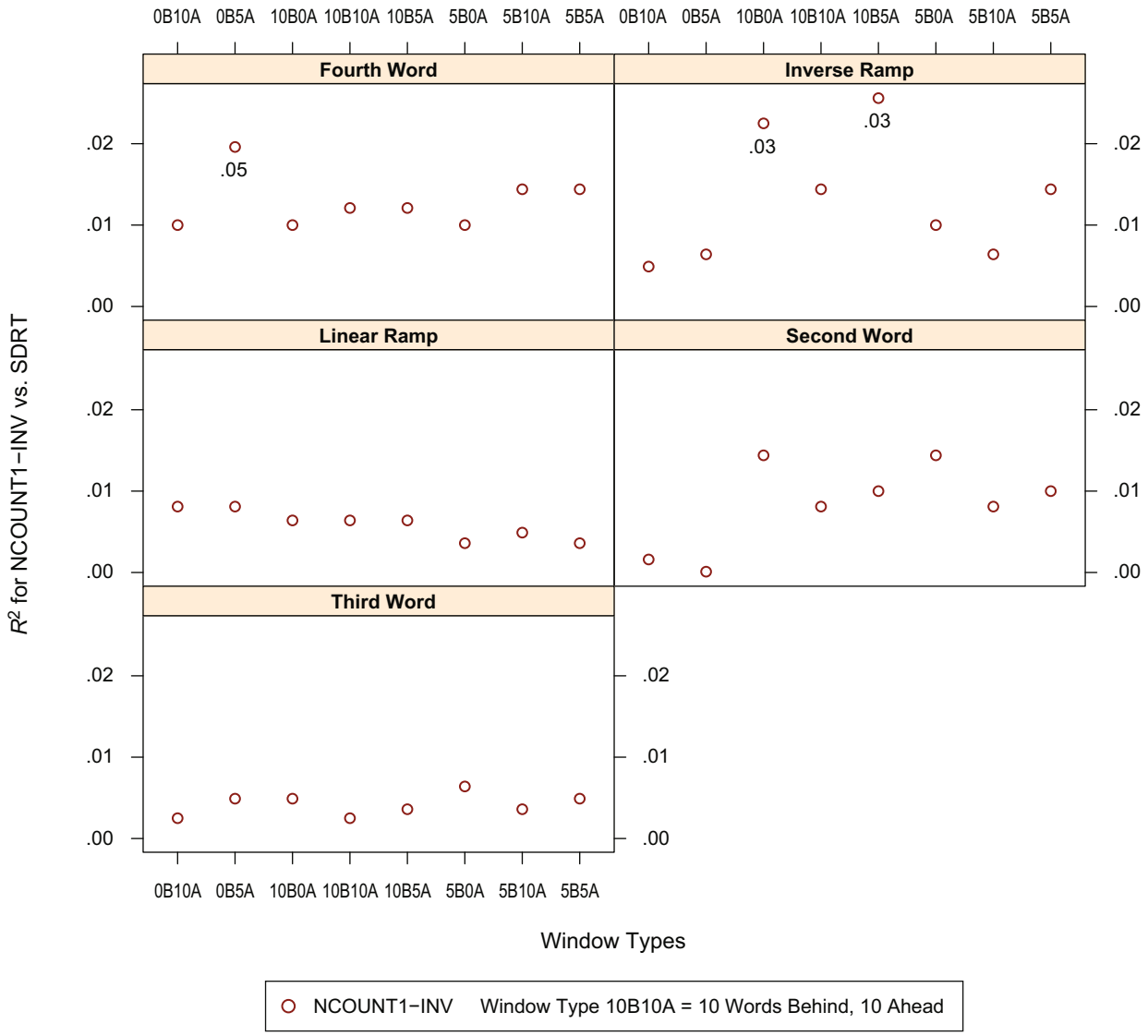


Figure 6. Overview of R^2 of the inverse neighbor count (NCOUNT1-INV), with semantic decision reaction times (SDRTs), in the go/no-go task for different weighting functions and window types. The *p* value for each reliable regression is shown below each point.

GENERAL DISCUSSION

In Experiments 1, 2, and 3, we started with LDRT and SDRT data and searched for the optimal parameter sets that predicted RTs. We found that certain weighting functions and window sizes fared much better than others at predicting RTs. There was a clear consensus across all the experiments: The original HAL parameters do not create the best measures of neighborhood density for predicting lexical-semantic access time. There was an encouraging convergence in Experiments 2 and 3 that found that a small number of parameter sets produced the strongest correlations with a semantic decision task. The best weighting function for Experiments 2 and 3, the inverse ramp, was also the best for Experiment 1 (shown in Figure 3). The best window types were those that contained 10 words behind and 0 or 5 words ahead. These results suggest that there is a very important function served by the inverse-ramp weighting function. What could be the reason for its superiority?

Unlike the linear ramp, the inverse-ramp weighting function gives greater significance to words that are located farther away in the window from the word in question. This has the effect of reducing the impact of the words closest to the target word. What kinds of words are usually found in these positions? The intuitive answer is *function words*, also-called *closed-class words*. Unlike nouns, adjectives, adverbs, and verbs, these are words that cannot have any new members to their class. They contain little semantic information about the words they appear next to but do create semantic relationships between words in a sentence.

Are our intuitions correct about closed-class words? In a cursory analysis of a 1-trillion-word corpus of English culled from Web pages (Brants & Franz, 2006), we found

some evidence that they are. Using a corpus-specific list of the 114 most frequent closed-class words in this corpus (see Appendix D for the full list), we counted the number of times a closed-class word appeared near an open-class word. We looked at the full set of 1 trillion 5-grams and found large differences in the probability of one of these 114 closed-class words appearing in each position in the window. Table 5 shows the probabilities for the four positions ahead of and behind any open-class word. All probabilities were reliably different from each other when tested with a proportion *z*-score test (in all cases, $p < .0001$). Since Positions −3, −2, 1, and 2 had the highest proportion of closed-class words, we propose that the weighting schemes that reduce the weight given to closed-class word contexts may be better at capturing semantic context relationships because of the decrease in closed-class contextual information. This makes sense when we look at an example: The similarity between the contexts of "cats" and "dogs" is more informed by "dog lovers and cat lovers" and "cat lovers and dog lovers" than by "the cat is going to" and "the dog is going to." By changing the weighting scheme, we changed the relative importance of closed-class word context and made the model better. We note here that Bullinaria and Levy (2007) found that their model performed best on their tasks when they used a 1-word window, whereas we found that giving weight to adjacent words did not help our models predict behavioral measures. This may be due to the different type of behavior being modeled.

Note that, in the semantic decision experiments, the neighborhood density of the first word (and never of the second word) in our word pairs produced the only significant relationships with SDRT. The denser the neighborhood of the first word, the longer the time needed to make a semantic decision. This is evidence that the contextual richness of the first word influences the processing of the semantic decision. Surprisingly, this relationship is in the opposite direction of the relationship reported by Buchanan et al. (2001) and Siakaluk et al. (2003). In these studies, a higher contextual neighborhood density was found to facilitate lexical access. Our results indicate that a higher co-occurrence neighborhood density facilitates lexical access while simultaneously increasing the cognitive load of semantic decision processing.

The apparent contradiction between facilitatory and inhibitory effects of dense neighborhoods has recently been analyzed by Mirman and Magnuson (2008). They compared different models of semantic neighborhood density to determine whether there were consistent facilitatory/inhibitory effects across different neighborhood density measures. They compared feature-based models using data from Cree, McNorgan, and McRae (2006); association-space models using data from Nelson et al. (1998); and co-occurrence models using data from COALS (Rohde et al., 2005). They found that certain neighborhood measurements were correlated with facilitation, whereas others were correlated with inhibition in both lexical decision and semantic decision tasks (living/nonliving and abstract/concrete, respectively).

In particular, they found that a single measure of neighborhood density was unable to account for the pattern of results. Instead, they found that both the number of neighbors and the distance between those neighbors was needed to understand the seemingly contradictory results. They reported that words with many near neighbors were categorized more slowly in a semantic decision task than were words with few near neighbors. They also found that words with many distant neighbors were categorized in the same task more quickly than were words with few distant neighbors. Mirman and Magnuson (2008) then went on to model this phenomenon with a feature vector-based attractor model (a type of neural network model; for the model's architecture see Cree et al., 2006). In light of this work by Mirman and Magnuson, we propose an alternative interpretation of our results from Experiments 2 and 3: Since our NCOUNT-INV measure is built using a threshold and it only counts the nearest neighbors, it also captures data about how many near neighbors a word has. Independently, we have found an inhibitory effect for neighborhood density identical to the effect found by Mirman and Magnuson, despite the fact that we used a relatedness judgment task, whereas they used other semantic categorization tasks.

How does this result relate to previous research into lexical semantic processing? Our semantic decision task is unlike most semantic psycholinguistic tasks. An extensive amount of research has been done on semantic priming (for a review, see Moss & Tyler, 1995). Unlike most semantic priming experiments, the semantic task we developed is not a lexical decision task. There is no implicit, subliminal semantic activation. The facilitation or inhibition in our experiments was the result of a combination of the semantic relationship between the words in the pair and the participant's strategies. This difference in methodology makes comparisons of effect size between our experiments and lexical decision semantic priming experiments difficult. What about semantic categorization/semantic decision tasks? The difference between traditional semantic decision tasks and our tasks is that, in most semantic decision tasks, a category, such as "concrete words," or an exemplar, such as "an animal," is used throughout the experiment. The task for the participant is usually a category membership decision that stays constant throughout the experiment. In our task, the category or exemplar is different in each and every trial. This makes it difficult to compare our results with those from traditional semantic decision experiments.

Table 5
Probabilities of Encountering Closed-Class Words Around an Open-Class Word

−4	−3	−2	−1	Center of Window	+1	+2	+3	+4
.22	.24	.25	.23	Any open-class word	.26	.24	.21	.20

For example, it may explain why our RTs were much slower than those in experiments that used semantic decisions for concreteness (Binder, Westbury, McKiernan, Possing, & Medler, 2005) and animacy (Siakaluk et al., 2003).

There is at least one study that used a task very similar to ours (a forced choice relatedness task) to study semantic processing: Pexman, Hino, and Lupker (2004) used a relatedness task to investigate ambiguity in semantic processing. They found that for *no* trials (trials on which the two words were not related), there was no ambiguity effect, and on related trials, there was an ambiguity disadvantage. They also proposed that this disadvantage was due to the semantic decision task itself and not to the process of retrieving the semantic representations for the words. We did not collect ambiguity ratings for the words in our stimulus set. This makes it difficult to compare our experiments with those in Pexman et al. The relationship between ambiguity and co-occurrence neighborhood density merits further study.

Conclusions

We have presented three experiments that explore the effect of varying two HAL parameters on modeling semantic processing tasks. Experiment 1 compared the LDRT predictions of 73 different HAL parameter sets. Experiments 2 and 3 used, respectively, a forced choice task and a go/no-go task to determine whether a task with an increased semantic load would show a predictive pattern for the HAL model. We found that, for certain optimized parameter sets, ARC and NCOUNT-INV were able to account for a large amount of variability in LDRTs. We then tested the power of these near-optimal parameter sets to predict SDRT in a novel task. The amount of variability explained by the optimal parameter sets in the semantic decision model was small in comparison to that in the lexical decision model but converged on the same parameter settings that were found in the LDRT experiments. We have shown that changing the weighting function and window size parameters of the HAL model can have a powerful impact on the ability of HAL to predict LDRT and SDRT. Additionally, the best set of parameters found were not those used in the original HAL model by Lund and Burgess (1996).

Finally, we found that the best set of parameters for predicting RTs were convergent for the SDRT and LDRT data. This finding opens the door to more research using HAL as a model for predicting behavioral data. Linguistic tasks that have a large semantic component could be modeled with a HAL-like representation of semantic information. If these models had their best fit using the same parameters discussed here, it would point to a general applicability of these parameter settings.

We have looked at how the local co-occurrence frequencies are weighted before being input into the model. The optimal weighting schemes—inverse ramp (LDRT) and fourth word (SDRT)—reduce the influence of the words directly preceding or following a word in its context. We analyzed a very large corpus of English and found that the vast majority of adjacent words are closed-class words.

We speculate that these closed-class words can act like superfluous context in our model, whereas the contextual information in the open-class words nearby are valuable contexts. In practical terms, the co-occurrence values for closed-class words will be smaller relative to those for open-class words when using the inverse-ramp or fourth-word weighting schemes. The weighting schemes that allow us to best predict behavior are the ones that filter out information about co-occurrence with closed-class words.

If the weighting scheme parameter has a potential psycholinguistic link, what about the window size parameter? Why is the optimal window setting that we found, 10B0A, better than the others? The relative importance of the backward window over the forward window might be due to the way that working memory stores recently perceived language. Only the most recently heard words are kept in the phonological loop (Baddeley, 2003), in much the same way that only the most recently seen words are kept in the 10B5A window. Furthermore, specific language impairment has been linked to impairments of working memory and is suspected of leading to problems in acquiring the meanings of words (Baddeley, 2003). If the concept of working-memory span can be considered analogous to the idea of window size, then perhaps the optimum size of a person's working-memory span for learning the meanings of words can be modeled using HAL. We note that removing the influence of preceding words removes half of the information from the original HAL global co-occurrence matrix, shrinking the actual dimensionality and, therefore, the size of high-dimensional space.

HiDEx is a powerful tool for investigators who need to measure the contextual similarity of words. It allows the user the flexibility to choose any corpus in any language as the input to the model. Many different variants of the HAL model are available, and each model can be configured so that all permutations of parameters can be tested. The output is also flexible enough to accommodate the needs of the investigator. By comparing the similarity measures produced by HiDEx with behavioral data, experimentalists can better understand the influence of co-occurrence measures on performance. We have demonstrated this by analyzing LDRT/SDRT data using HiDEx output. Experiment designers can also use HiDEx to create stimulus sets with particular co-occurrence neighborhood properties. Finally, our software, HiDEx, can be used as a reference implementation for other researchers seeking to compare results produced from identical corpora using different HAL implementations.

AUTHOR NOTE

This research was supported by grants to the authors from the Natural Sciences and Engineering Research Council of Canada (to C.S. and C.W.) and the Alberta Heritage Foundation for Medical Research (to C.W.). Correspondence concerning this article should be addressed to C. Shaoul, Department of Psychology, University of Alberta, Biological Sciences Building, Edmonton, AB, T6G 2E9 Canada (e-mail: cyrus.shaoul@ualberta.ca).

REFERENCES

AKAIKE, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, **19**, 716-723.

BAAYEN, R. H. (2001). *Word frequency distributions*. Boston: Kluwer.

BAAYEN, R. H., PIEPENBROCK, R., & GULIKERS, L. (1995). The CELEX Lexical Database (Release 2) [CD-ROM]. Philadelphia: University of Pennsylvania, Linguistic Data Consortium.

BADDELEY, A. (2003). Working memory and language: An overview. *Journal of Communication Disorders*, **36**, 189-208.

BALOTA, D. A., BLACK, S. R., & CHENEY, M. (1992). Automatic and attentional priming in young and older adults: Reevaluation of the two-process model. *Journal of Experimental Psychology: Human Perception & Performance*, **18**, 485-502.

BALOTA, D. A., CORTESE, M. J., HUTCHISON, K. A., NEELY, J. H., NELSON, D., SIMPSON, G. B., & TREIMAN, R. (2002). *The English Lexicon Project: A Web-based repository of descriptive and behavioral measures for 40,481 English words and nonwords*. Retrieved October 5, 2005, from http://elexicon.wustl.edu/.

BINDER, J. R., WESTBURY, C. F., MCKIERNAN, K. A., POSSING, E. T., & MEDLER, D. A. (2005). Distinct brain systems for processing concrete and abstract concepts. *Journal of Cognitive Neuroscience*, **17**, 905-917.

BRANTS, T., & FRANZ, A. (2006). Web 1T 5-Gram Corpus (Version 1). Philadelphia: University of Pennsylvania, Linguistic Data Consortium.

BUCHANAN, L., BURGESS, C., & LUND, K. (1996). Overcrowding in semantic neighborhoods: Modeling deep dyslexia. *Brain & Cognition*, **32**, 111-114.

BUCHANAN, L., WESTBURY, C., & BURGESS, C. (2001). Characterizing semantic space: Neighborhood effects in word recognition. *Psychonomic Bulletin & Review*, **8**, 531-544.

BULLINARIA, J. A., & LEVY, J. P. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, **39**, 510-526.

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

BURGESS, C., & LIVESAY, K. (1998). The effect of corpus size in predicting reaction time in a basic word recognition task: Moving on from Kučera 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.

BURGESS, C., & LUND, K. (2000). The dynamics of meaning in memory. In E. Dietrich & A. B. Markman (Eds.), *Cognitive dynamics: Conceptual and representational change in humans and machines* (pp. 117-156). Mahwah, NJ: Erlbaum.

CHAPMAN, B., JOST, G., VAN DER PAS, R., & KUCK, D. (2007). *Using OpenMP: Portable shared memory parallel programming*. Cambridge, MA: MIT Press.

CREE, G. S., MCNORGAN, C., & MCRAE, K. (2006). Distinctive features hold a privileged status in the computation of word meaning: Implications for theories of semantic memory. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **32**, 643-658.

DURDA, K., & BUCHANAN, L. (2008). WINDSORS: Windsor improved norms of distance and similarity of representations of semantics. *Behavior Research Methods*, **40**, 705-712.

DURDA, K., BUCHANAN, L., & CARON, R. (2009). Grounding co-occurrence: Identifying features in a lexical co-occurrence model of semantic memory. *Behavior Research Methods*, **41**, 1210-1223.

FRISTRUP, J. A. (1994). *USENET: Netnews for everyone.* Englewood Cliffs, NJ: Prentice Hall.

HOLLIS, G., WESTBURY, C. F., & PETERSON, J. B. (2006). NUANCE 3.0: Using genetic programming to model variable relationships. *Behavior Research Methods*, **38**, 218-228.

JONES, M. N., KINTSCH, W., & MEWHORT, D. J. K. (2006). High-dimensional semantic space accounts of priming. *Journal of Memory & Language*, **55**, 534-552.

JONES, M. N., & MEWHORT, D. J. K. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, **114**, 1-37.

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**, 211-240.

LIFCHITZ, A., JHEAN-LAROSE, S., & DENHIÈRE, G. (2009). Effect of tuned parameters on an LSA multiple choice questions answering model. *Behavior Research Methods*, **41**, 1201-1209.

LUND, K., & BURGESS, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instrumentation, & Computers*, **28**, 203-208.

MIRMAN, D., & MAGNUSON, J. S. (2008). Attractor dynamics and semantic neighborhood density: Processing is slowed by near neighbors and speeded by distant neighbors. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **34**, 65-79.

MOSS, H. E., & TYLER, L. K. (1995). Investigating semantic memory impairments: The contribution of semantic priming. *Memory*, **3**, 359-395.

MURDOCK, B. B. (1982). A theory for the storage and retrieval of item and associative information. *Psychological Review*, **89**, 609-626.

NELSON, D. L., MCEVOY, C. L., & SCHREIBER, T. A. (1998). *The University of South Florida word association, rhyme, and word fragment norms.* Available from www.usf.edu/FreeAssociation/.

PEXMAN, P. M., HINO, Y., & LUPKER, S. J. (2004). Semantic ambiguity and the process of generating meaning from print. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **30**, 1252-1270.

RECCHIA, G., & JONES, M. N. (2009). More data trumps smarter algorithms: Comparing pointwise mutual information with latent semantic analysis. *Behavior Research Methods*, **41**, 647-656.

ROHDE, D. L. T., GONNERMAN, L. M., & PLAUT, D. C. (2005). *An improved method model of semantic similarity based on lexical co-occurrence.* Unpublished manuscript. Retrieved April 20, 2007, from http://tedlab.mit.edu/~dr/.

RUSSELL, B. (1910). The study of mathematics. In *Philosophical essays*. London: Longmans, Green.

SHAOUL, C., & WESTBURY, C. (2006a). *USENET orthographic frequencies for the 40,481 words in the English lexicon project* [Data file]. Available from the University of Alberta Web site: www.psych.ualberta.ca/~westburylab/downloads.html.

SHAOUL, C., & WESTBURY, C. (2006b). Word frequency effects in high-dimensional co-occurrence models: A new approach. *Behavior Research Methods*, **38**, 190-195.

SHAOUL, C., & WESTBURY, C. (2008). HiDEx: High Dimensional Explorer [Software]. Available from the University of Alberta Web site: www.psych.ualberta.ca/~westburylab/downloads.usenetcorpus.html.

SHAOUL, C., & WESTBURY, C. (2009). *A USENET corpus* (2005–2009). Available from the University of Alberta Web site: www.psych.ualberta.ca/~westburylab/downloads.usenetcorpus.html.

SIAKALUK, P. D., BUCHANAN, L., & WESTBURY, C. (2003). The effect of semantic distance in yes/no and go/no-go semantic categorization tasks. *Memory & Cognition*, **31**, 100-113.

SONG, D., & BRUZA, P. (2001, September 10). *Discovering information flow using a high dimensional conceptual space*. Paper presented at the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans.

SONG, D., BRUZA, P., & COLE, R. (2004, July 30). *Concept learning and information inferencing on a high-dimensional semantic space*. Paper presented at the ACM SIGIR 2004 Workshop on Mathematical/Formal Methods in Information Retrieval, Sheffield, U.K.

SONG, D., BRUZA, P., HUANG, Z., & LAU, R. K. (2003). Classifying document titles based on information inference. In J. G. Carbonell & J. Siekmann (Eds.), *Foundations of intelligent systems* (pp. 297-306). Berlin: Springer.

STALLMAN, R. (2009). *GNU General Public License.* Available from www.fsf.org/licensing/.

WESTBURY, C. (2007). *ACTUATE: Assessing Cases, The University of Alberta Testing Environment*. Available from the University of Alberta Web site: www.psych.ualberta.ca/~westburylab.

YATES, M., LOCKER, L., JR., & SIMPSON, G. B. (2003). Semantic and phonological influences on the processing of words and pseudohomophones. *Memory & Cognition*, **31**, 856-866.

ZIPF, G. K. (1935). *The psycho-biology of language: An introduction to dynamic philology.* Boston: Houghton Mifflin.

ZIPF, G. K. (1949). *Human behavior and the principle of least effort*. New York: Addison-Wesley.

NOTES

1. Buchanan et al. (1996) called their measure of co-occurrence density "semantic neighborhood."

2. This notation, 10B10A, is a condensed expression of "10 words behind, 10 words ahead."

3. We also calculated the Bayesian information criterion (BIC) for these models and saw the same result: The optimized models had smaller BIC values. For ARCs, the BICs were 61,680 and 61,460, and for NCOUNT-INV, the BICs were 61,369 and 61,251.

APPENDIX A
Associated Words Sorted by Interword HAL distance

Word Pair	HAL Distance	OF (Word 1)	OF (Word 2)	Letters (Word 1)	Letters (Word 2)	ON (Word 1)	ON (Word 2)
essay–English	41.97	6.58	93.71	5	7	1	0
factory–labor	42.15	21.42	38.80	7	5	1	1
arrest–suspect	42.55	20.87	42.56	6	7	1	0
differ–similar	43.79	9.30	110.97	6	7	2	1
average–regular	43.89	66.71	52.13	7	7	1	0
biology–cell	43.99	11.80	43.69	7	4	0	16
cell–biology	43.99	43.69	11.80	4	7	16	0
legs–stretch	44.11	24.48	10.13	4	7	15	0
bother–trouble	44.20	29.37	64.90	6	7	4	0
experts–panel	44.21	32.77	21.73	7	5	3	2
decency–respect	44.40	16.87	65.89	7	7	1	0
worry–panic	45.87	37.31	8.72	5	5	4	1
knight–horse	46.09	11.39	24.55	6	5	0	10
corrupt–lawyer	48.11	21.84	27.26	7	6	0	1
kingdom–queen	49.96	27.20	22.20	7	5	0	3
fake–pretend	50.20	22.34	18.81	4	7	16	1
liquid–drink	51.62	10.81	29.06	6	5	0	4
fence–chain	54.20	11.57	32.74	5	5	2	1
engine–loud	54.84	45.61	17.12	6	4	0	6
doctor–operate	55.88	45.30	23.32	6	7	0	1
cloud–weather	56.76	27.77	24.91	5	7	2	3
sports–stadium	57.23	24.41	18.39	6	7	7	0
counsel–lawyer	58.17	14.17	27.26	7	6	0	1
pipe–valve	59.95	15.14	7.33	4	5	9	6
cowboy–range	60.20	5.76	70.58	6	5	1	3
depart–airport	61.01	47.06	20.05	6	7	1	0
depth–width	63.90	15.75	87.42	5	5	2	0
beach–relax	64.69	29.81	5.61	5	5	7	2
corrupt–destroy	65.74	21.84	40.24	7	7	0	0
artist–talent	66.76	14.10	19.96	6	6	0	0
recipe–mixture	67.82	10.07	8.12	6	7	1	1
Mercury–planets	69.05	17.50	11.22	7	7	1	1
opening–valve	69.80	49.03	7.33	7	5	1	6
fantasy–pretend	71.08	20.84	18.81	7	7	2	1
fortune–fame	71.54	23.52	9.70	7	4	0	15
deliver–truck	72.02	32.08	22.17	7	5	1	4
license–permit	74.70	36.26	14.27	7	6	1	1
courage–coward	77.16	12.84	16.95	7	6	0	2
bondage–chain	79.88	5.05	32.74	7	5	1	1
launch–missile	81.01	20.95	14.85	6	7	2	3
doorway–portal	82.36	14.02	14.40	7	6	0	3
economy–deficit	88.44	47.30	10.45	7	7	0	0
window–shield	91.07	62.29	8.65	6	6	1	0
English–poetry	94.08	93.71	11.75	7	6	0	0
chicken–recipe	96.47	17.67	10.07	7	6	1	1
ancient–temple	101.83	29.78	19.87	7	6	1	0
mailbox–empty	102.09	7.01	65.53	7	5	0	1
worship–temple	112.10	27.05	19.87	7	6	1	0
helpful–useless	136.19	30.30	18.69	7	7	0	0
session–therapy	138.77	21.90	12.60	7	7	1	0
Means	66.56	26.24	28.60	6.24	6	2.22	2.06

Note—HAL, hyperspace analog to language; OF, orthographic frequency; ON, orthographic neighborhood.

APPENDIX B
Idiosyncratic Words Sorted by Interword HAL Distance

Word Pair	HAL Distance	OF (Word 1)	OF (Word 2)	Letters (Word 1)	Letters (Word 2)	ON (Word 1)	ON (Word 2)
drama–serious	29.48	7.44	96.40	5	7	1	0
sharp–harsh	29.76	54.36	10.53	5	5	4	1
notion–purpose	32.91	18.00	67.18	6	7	4	0
routine–cycle	34.71	21.68	39.60	7	5	1	1
foolish–actions	36.50	11.77	63.83	7	7	1	0
mess–kitchen	38.56	21.12	13.40	4	7	15	0
Monday–busy	42.46	43.81	24.58	6	4	0	6
plastic–trend	42.58	21.00	13.83	7	5	3	2
apples–trees	42.86	5.73	36.47	6	5	0	10
priest–college	43.54	20.68	112.91	6	7	2	1
unite–apart	44.43	5.49	32.41	5	5	2	1
gravity–rock	46.24	18.23	44.05	7	4	0	16
luck–rainbow	47.83	46.22	6.92	4	7	16	0
park–reserve	48.67	39.98	15.60	4	7	16	1
select–careful	49.48	48.43	25.04	6	7	0	1
wild–beast	51.50	28.48	12.66	4	5	9	6
praise–destroy	51.52	16.85	40.24	6	7	1	0
succeed–lucky	52.19	13.58	28.90	7	5	0	3
citizen–arrest	52.45	30.55	20.87	7	6	0	1
imagine–suppose	53.20	52.52	42.64	7	7	1	0
panel–buttons	58.59	21.73	26.69	5	7	2	3
frigid–worry	58.90	5.04	37.31	6	5	0	4
express–hurry	62.19	70.76	6.63	7	5	1	6
morals–immoral	62.93	5.06	8.26	6	7	7	0
pilot–error	62.95	18.09	90.28	5	5	2	0
advise–console	63.44	12.27	9.26	6	7	1	1
collect–mess	64.02	16.63	21.12	7	4	0	15
succeed–wealth	64.35	13.58	26.69	7	6	0	1
bread–flour	65.67	14.23	5.57	5	5	7	2
cement–floor	67.61	5.66	57.32	6	5	1	3
ethics–virtue	69.22	17.97	9.78	6	6	0	0
curious–mystery	69.95	18.85	14.33	7	7	2	1
passage–mystery	70.93	15.04	14.33	7	7	1	1
realize–beauty	71.48	46.44	15.89	7	6	1	1
poverty–student	71.65	19.08	38.64	7	7	0	0
region–portion	73.86	48.12	19.97	6	7	1	0
severe–weather	75.25	18.96	24.91	6	7	2	3
servant–castle	78.03	7.24	19.66	7	6	0	2
capture–steal	83.82	17.23	17.07	7	5	1	4
failure–dismiss	84.92	51.42	6.76	7	7	0	0
defend–knight	86.49	35.53	11.39	6	6	1	0
capture–rescue	86.77	17.23	14.63	7	6	1	1
kingdom–Mickey	87.74	27.20	6.94	7	6	0	3
ancient–magic	89.82	29.78	29.47	7	5	1	1
freedom–escape	98.18	99.08	19.53	7	6	0	0
collect–items	100.43	16.63	55.93	7	5	0	1
capture–victim	102.03	17.23	29.58	7	6	1	0
kingdom–Britain	123.04	27.20	31.07	7	7	0	0
decency–ethics	123.29	16.87	17.97	7	6	1	0
surgery–miracle	128.94	17.14	8.74	7	7	1	0
Means	65.55	25.46	28.88	6.24	6	2.22	2.06

Note—HAL, hyperspace analog to language; OF, orthographic frequency; ON, orthographic neighborhood.

APPENDIX C
Unrelated Words Sorted by Interword HAL Distance

Word Pair	HAL Distance	OF (Word 1)	OF (Word 2)	Letters (Word 1)	Letters (Word 2)	ON (Word 1)	ON (Word 2)
fruits–towards	28.68	5.77	112.61	6	7	2	1
seldom–votes	33.06	6.12	24.64	6	5	0	10
blunt–react	36.97	44.69	11.75	5	5	4	1
fought–obvious	37.78	18.63	63.98	6	7	4	0
helmet–quote	37.98	6.27	68.23	6	5	1	3
rice–element	38.51	33.01	20.37	4	7	16	1
beloved–query	39.73	8.73	38.49	7	5	1	1

APPENDIX C (Continued)

Word Pair	HAL Distance	OF (Word 1)	OF (Word 2)	Letters (Word 1)	Letters (Word 2)	ON (Word 1)	ON (Word 2)
slowly–tube	40.49	43.56	14.47	6	4	0	6
payroll–diets	41.07	25.31	21.29	7	5	0	3
essence–park	41.57	11.51	39.98	7	4	0	16
involve–scheme	42.71	19.59	28.70	7	6	0	1
wind–signals	43.39	27.73	15.36	4	7	15	0
hint–sandy	43.74	13.00	8.04	4	5	9	6
circuit–aspects	44.02	22.77	38.59	7	7	0	0
broken–remarks	44.59	39.11	17.89	6	7	1	0
witness–broad	50.64	23.22	22.78	7	5	3	2
miss–patriot	51.81	39.46	13.47	4	7	16	0
knees–ended	52.78	8.90	30.91	5	5	2	1
treaty–animals	53.46	11.75	43.82	6	7	1	0
prophet–monthly	53.83	24.85	68.38	7	7	1	0
divorce–gotten	53.91	10.60	25.83	7	6	0	1
voted–knees	54.37	24.82	8.90	5	5	7	2
linked–popcorn	54.89	18.01	15.41	6	7	7	0
juice–perform	62.49	9.40	94.95	5	7	1	0
lemon–elected	62.78	21.93	27.41	5	7	2	3
awhile–suite	63.21	7.92	33.44	6	5	0	4
finest–symbol	63.96	10.78	17.62	6	6	0	0
smooth–clarify	64.37	11.84	7.56	6	7	1	1
appears–failure	65.22	61.30	51.42	7	7	1	0
compare–coast	65.88	24.68	25.98	7	5	1	4
arrives–penalty	66.52	20.49	16.98	7	7	2	1
female–entered	68.80	45.40	23.39	6	7	0	1
cruelly–explore	69.59	22.46	9.80	7	7	1	1
pacific–dozens	74.17	14.58	17.84	7	6	0	3
joined–crystal	75.24	23.91	17.75	6	7	2	3
shuttle–favor	75.60	6.10	33.87	7	5	1	1
pacific–pace	76.42	14.58	11.95	7	4	0	15
imagine–bench	77.85	52.52	7.24	7	5	1	6
knock–third	78.32	12.46	90.50	5	5	2	0
enforce–stream	84.38	9.51	14.04	7	6	0	2
classic–expand	87.31	28.89	16.19	7	6	1	1
friends–guards	93.91	96.63	12.71	7	6	0	0
package–mirrors	96.41	45.88	8.27	7	7	0	0
militia–upper	98.37	11.51	67.46	7	5	0	1
behave–poetry	99.60	53.28	11.75	6	6	1	0
element–scream	108.35	20.37	8.93	7	6	1	1
ordered–enigma	116.38	32.55	19.53	7	6	1	0
soldier–forums	122.25	20.54	19.75	7	6	1	0
weekend–tactics	130.50	30.34	18.66	7	7	0	0
chicken–deposit	152.59	17.67	14.28	7	7	1	0
Means	66.41	24.30	29.06	6.24	6	2.22	2.06

Note—HAL, hyperspace analog to language; OF, orthographic frequency; ON, orthographic neighborhood.

APPENDIX D
The 114 Most Frequent Closed-Class Words From the Web1T Corpus

a	about	all	also	more	most	my	no	with	would	you	your
an	and	any	are	not	now	of	on	</S>	<S>	@	=
as	at	be	been	one	only	or	other	>	?	’	’S
but	by	can	do	our	out	over	should	(	)	*	+
first	for	from	get	so	some	such	than	,	-	–	.
had	has	have	he	that	the	their	them	...	/	:	;
her	here	his	how	there	these	they	this	[	\	]	|
I	if	in	into	through	to	up	us	!	“	#	$
is	it	its	just	was	we	were	what	%	&		
like	make	may	me	when	which	who	will				

Note—All of these words were among the 200 most frequently used words in the Web1T corpus. <*S*> and </*S*> denote the beginning and end of a sentence.

(Manuscript received November 12, 2009;
revision accepted for publication February 26, 2010.)