

An Improved Model of Semantic Similarity Based on Lexical Co-Occurrence

Douglas L. T. Rohde

Massachusetts Institute of Technology, Department of Brain and Cognitive Sciences

Laura M. Gonnerman

Lehigh University, Department of Psychology

David C. Plaut

Carnegie Mellon University, Department of Psychology,
and the Center for the Neural Basis of Cognition

November 7, 2005

Abstract

The lexical semantic system is an important component of human language and cognitive processing. One approach to modeling semantic knowledge makes use of hand-constructed networks or trees of interconnected word senses (Miller, Beckwith, Fellbaum, Gross, & Miller, 1990; Jarmasz & Szpakowicz, 2003). An alternative approach seeks to model word meanings as high-dimensional vectors, which are derived from the co-occurrence of words in unlabeled text corpora (Landauer & Dumais, 1997; Burgess & Lund, 1997a). This paper introduces a new vector-space method for deriving word-meanings from large corpora that was inspired by the HAL and LSA models, but which achieves better and more consistent results in predicting human similarity judgments. We explain the new model, known as COALS, and how it relates to prior methods, and then evaluate the various models on a range of tasks, including a novel set of semantic similarity ratings involving both semantically and morphologically related terms.

1 Introduction

The study of lexical semantics remains a principal topic of interest in cognitive science. Lexical semantic models are typically evaluated on their ability to predict human judgments about the similarity of word pairs, expressed either as explicit synonymy ratings (Rubenstein & Goodenough, 1965; Miller & Charles, 1991) or implicitly through such measures as priming (Plaut & Booth, 2000; McDonald & Lowe, 1998). One well-known approach to modeling human lexical semantic knowledge is WordNet—a large network of word forms, their associated senses, and various

links that express the relationships between those senses (Miller et al., 1990). WordNet itself does not provide a word-pair similarity metric, but various metrics based on its structure have been developed (Rada, Mili, Bicknell, & Blettner, 1989; Budanitsky & Hirst, 2001; Patwardhan, Banerjee, & Pedersen, 2003). Metrics have also been built upon other lexical databases, such as Roget's Thesaurus (Jarmasz & Szpakowicz, 2003).

Another common approach to modeling lexical semantics is the derivation of high-dimensional vectors, representing word meanings, from the patterns of word co-occurrence in large corpora. Latent Semantic Analysis (LSA; Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990) derives its vectors from collections of segmented documents, while the Hyperspace Analogue to Language method (HAL; Lund & Burgess, 1996) makes use of unsegmented text corpora. These vector-space approaches are limited in that they do not model individual word senses, but they do have certain practical advantages over WordNet or thesaurus-based approaches. The vector-based methods do not rely on hand-designed datasets and the representations in which they encode semantic knowledge are quite flexible and easily employed in various tasks. Vector representations are also attractive from a cognitive modeling standpoint because they bear an obvious similarity to patterns of activation over collections of neurons.

Although one goal of research in this area is to directly model and understand human lexical semantics, another important goal is the development of semantic representations that are useful in studying other cognitive tasks that are thought to be dependent on lexical semantics, such as word reading (Plaut, Seidenberg, McClelland, & Patterson, 1996), lexical decision (Plaut, 1997), past

tense formation (Ramsar, 2001), and sentence processing (Rohde, 2002a). The HAL methodology has been used for such diverse applications as modeling contextual constraints in the parsing of ambiguous sentences (Burgess & Lund, 1997a), distinguishing semantic and associative word priming (Lund, Burgess, & Atchley, 1995; Lund, Burgess, & Audet, 1996), and modeling dissociations in the priming of abstract and emotion words (Burgess & Lund, 1997b), while LSA has been used in modeling categorization (Laham, 1971), textual coherence (Foltz, Kintsch, & Landauer, 1998), and metaphor comprehension (Kintsch, 2000).

In this paper, we introduce a new vector-space method—the Correlated Occurrence Analogue to Lexical Semantic, or COALS—which is based on HAL, but which achieves considerably better performance through improvements in normalization and other algorithmic details. A variant of the method, like LSA, uses the singular value decomposition to reduce the dimensionality of the resulting vectors and can also produce binary vectors, which are particularly useful as input or output representations in training neural networks.

In Section 2, we briefly review the methods used in the 11 other models against which COALS will be evaluated and then describe the model itself. In Section 3, we test the models on a variety of tasks involving semantic similarity rating and synonym matching. In doing so, we introduce a new empirical benchmark of human ratings of 400 word pairs, which makes use of a diverse set of lexical relationships. We also analyze the vectors produced by COALS using multidimensional scaling, hierarchical clustering, and studies of nearest-neighbor terms. Section 4 introduces some variations on the COALS method to investigate the effects of alternative normalization techniques and parameter choices and to better understand the differences between HAL and COALS.

2 Lexical semantic models

In this section we review the methods used in a variety of popular semantic models, including HAL, LSA, and several lexicon-based techniques. We then introduce the COALS model and some of its variants.

2.1 The HAL model

The HAL method for modeling semantic memory (Lund & Burgess, 1996; Burgess & Lund, 1997a) involves constructing a high-dimensional vector for each word such that, it is hoped, the pairwise distances between the points represented by these vectors reflect the similarity in meaning of the words. These semantic vectors are derived from the statistics of word co-occurrence in a large corpus of

Table 1

A sample text corpus.

How much wood would a woodchuck chuck , if a woodchuck could chuck wood ?
As much wood as a woodchuck would , if a woodchuck could chuck wood .

text.

For the purpose of illustration, we will explain the model using the simple text corpus shown in Table 1. The HAL method begins by producing a co-occurrence matrix. For each word, a , we count the number of times every other word, b , occurs in close proximity to a . The counting is actually done using weighted co-occurrences. If b occurs adjacent to a , it receives a weighting of 10. If b is separated from a by one word, it receives a weighting of 9, and so forth on down to a weighting of 1 for distance-10 neighbors. We call this a ramped window of size 10. The cell $w_{a,b}$ of the co-occurrence matrix (row a , column b) contains the weighted sum of all occurrences of b in proximity to a .

HAL actually uses two separate columns for each neighboring word, b : one for occurrences of b to the left of a and one for occurrences to the right of a . Table 2 depicts the weighted co-occurrence table for the Woodchuck example. Along the *would* row, the first *woodchuck* column has a value of 10 because *woodchuck* appears immediately before *would* once. The second *woodchuck* column has a value of 20 because *woodchuck* occurs two words after the first *would* (9 points), 7 words after it (4 points), and 4 words after the second *would* (7 points).

The HAL model, as reported in the literature, has typically been trained on either a 160-million-word or a 300-million-word corpus of English text drawn from the Usenet discussion group service. The co-occurrence matrix uses 140,000 columns representing the leftward and rightward occurrences of 70,000 different words. The rows in the HAL co-occurrence table form 140,000-element semantic vectors that represent the meanings of the corresponding words. The more similar in meaning two words are, the more similar their vectors should be. In the HAL methodology, the vectors are normalized to a constant length (see Table 4) and then the distance between two words' vectors is computed with any Minkowski metric. Normally, Minkowski-2, or Euclidean distance, is used.

Vectors of size 140,000 are rather large and cumbersome, and Burgess suggests that their dimensionality can be reduced by eliminating all but the k columns with the highest variance. In this way, it is hoped, the most informative columns are retained. However, as the magnitude of a set of values is scaled up, the variance of that set increases with the square of the magnitude. Thus, it hap-

Table 2

Sample co-occurrence table used in the HAL method, prior to row length normalization.

	a	as	chuck	could	how	if	much	wood	woodch.	would	,	.	?	a	as	chuck	could	how	if	much	wood	woodch.	would	,	.	?
a	13	24	12	3	9	20	22	31	16	23	18	0	7	13	7	31	26	0	14	4	21	50	9	16	7	7
as	7	8	15	11	0	5	9	25	10	0	3	0	17	24	8	2	3	0	9	10	10	20	13	11	0	0
chuck	31	2	5	20	5	14	6	9	36	15	12	0	0	12	15	5	6	0	9	8	30	10	2	11	9	12
could	26	3	6	0	0	16	2	4	30	9	14	0	0	3	11	20	0	0	0	6	23	2	1	0	8	8
how	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	5	0	0	3	10	9	7	8	4	0	0
if	14	9	9	0	3	0	8	11	16	15	20	0	2	20	5	14	16	0	0	3	14	18	0	0	5	5
much	4	10	8	6	10	3	0	8	5	0	2	0	9	22	9	6	2	0	8	0	20	18	15	10	0	0
wood	21	10	30	23	9	14	20	7	26	5	11	0	8	31	25	9	4	0	11	8	7	26	20	14	10	10
woodch.	50	20	10	2	7	18	18	26	13	20	16	0	5	16	10	36	30	0	16	5	26	13	10	18	9	9
would	9	13	2	1	8	0	15	20	10	0	0	0	4	23	0	15	9	0	15	0	5	20	0	17	3	0
,	16	11	11	0	4	0	10	14	18	17	0	0	3	18	3	12	14	0	20	2	11	16	0	0	4	4
.	7	0	9	8	0	5	0	10	9	3	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
?	7	0	12	8	0	5	0	10	9	0	4	0	0	7	17	0	0	0	2	9	8	5	4	3	0	0

Table 3

Several possible vector similarity measures.

Inv. Sq. City-block:	$S(a,b) = \frac{1}{(\sum_i a_i - b_i)^2 + 1}$
Inv. Sq. Euclidean:	$S(a,b) = \frac{1}{\sum_i (a_i - b_i)^2 + 1}$
Cosine:	$S(a,b) = \frac{\sum_i a_i b_i}{(\sum_i a_i^2 \sum_i b_i^2)^{1/2}}$
Correlation:	$S(a,b) = \frac{\sum_i (a_i - \bar{a})(b_i - \bar{b})}{(\sum_i (a_i - \bar{a})^2 \sum_i (b_i - \bar{b})^2)^{1/2}}$

pens to be the case that the most variant columns tend to correspond to the most common words and selecting the k columns with largest variance is similar in effect to selecting the k columns with largest mean value, or whose corresponding words are most frequent. It is also the case that these columns tend to dominate in the computation of the Euclidean distance between two vectors. For this reason, eliminating all but the few thousand columns with the largest or most variant values has little effect on the relative distance between HAL vectors.

When using the Euclidean distance function, HAL produces values that decrease with greater semantic similarity. In order to convert these distances into a positive measure of semantic relatedness, they must be inverted. One effective method for doing this is to use the Inverse Squared Euclidean distance function given in Table 3. Due to the +1 in the denominator, this function is bounded between 0 and 1, where perfect synonyms would score a 1 and unrelated words a value close to 0. In practice, it actually matters little whether the Euclidean distances used with HAL are squared or the +1 is used.

The HAL models tested here were trained on the same 1.2 billion word Usenet corpus used for COALS (see Section 2.7). In the HAL-14K version of the model, the vec-

Table 4

Several vector normalization procedures.

Row:	$w'_{a,b} = \frac{w_{a,b}}{\sum_j w_{a,j}}$
Column:	$w'_{a,b} = \frac{w_{a,b}}{\sum_i w_{i,b}}$
Length:	$w'_{a,b} = \frac{w_{a,b}}{(\sum_j w_{a,j}^2)^{1/2}}$
Correlation:	$w'_{a,b} = \frac{T w_{a,b} - \sum_j w_{a,j} \cdot \sum_i w_{i,b}}{(\sum_j w_{a,j} \cdot (T - \sum_j w_{a,j}) \cdot \sum_i w_{i,b} \cdot (T - \sum_i w_{i,b}))^{1/2}}$
	$T = \sum_i \sum_j w_{i,j}$
Entropy:	$w'_{a,b} = \log(w_{a,b} + 1) / H_a$
	$H_a = -\sum_j \frac{w_{a,b}}{\sum_j w_{a,j}} \log\left(\frac{w_{a,b}}{\sum_j w_{a,j}}\right)$

tors were composed from the 14,000 columns with highest variance. In the HAL-400 model, only the top 400 columns were retained, which is closer to the 200 dimensions commonly used with this model.

2.2 The LSA model

LSA (Deerwester et al., 1990; Landauer, Foltz, & Laham, 1998) is based not on an undifferentiated corpus of text but on a collection of discrete documents. It too begins by constructing a co-occurrence matrix in which the row vectors represent words, but in this case the columns do not correspond to neighboring words but to documents. Matrix component $w_{a,d}$ initially indicates the number of occurrences of word a in document d . The rows of this matrix are then normalized. An entropy-based normalization, such as the one given in Table 4 or a slight variation thereof, is often used with LSA. It involves taking logarithms of the raw counts and then dividing by H_a , the en-

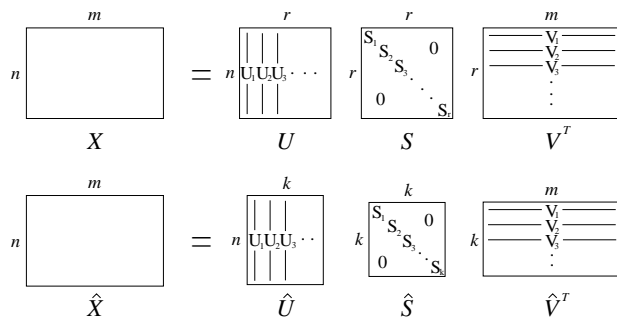


Figure 1: The singular value decomposition of matrix X . \hat{X} is the best rank k approximation to X , in terms of least squares.

trophy of the document distribution of row vector a . Words that are evenly distributed over documents will have high entropy and thus a low weighting, reflecting the intuition that such words are less interesting.

The critical step of the LSA algorithm is to compute the singular value decomposition (SVD) of the normalized co-occurrence matrix. An SVD is similar to an eigenvalue decomposition, but can be computed for rectangular matrices. As shown in Figure 1, the SVD is a product of three matrices, the first, U , containing orthonormal columns known as the *left singular vectors*, and the last, V^T containing orthonormal rows known as the *right singular vectors*, while the middle, S , is a diagonal matrix containing the *singular values*. The left and right singular vectors are akin to eigenvectors and the singular values are akin to eigenvalues and rate the importance of the vectors.¹ The singular vectors reflect principal components, or axes of greatest variance in the data.

If the matrices comprising the SVD are permuted such that the singular values are in decreasing order, they can be truncated to a much lower rank, k . It can be shown that the product of these reduced matrices is the best rank k approximation, in terms of sum squared error, to the original matrix X . The vector representing word a in the reduced-rank space is \hat{U}_a , the a th row of \hat{U} , while the vector representing document b is \hat{V}_b , the b th row of \hat{V} . If a new word, c , or a new document, d , is added after the computation of the SVD, their reduced-dimensionality vectors can be computed as follows:

$$\hat{U}_c = X_c \hat{V} \hat{S}^{-1}$$

$$\hat{V}_d = X_d^T \hat{U} \hat{S}^{-1}$$

The similarity of two words or two documents in LSA is usually computed using the cosine of their reduced-dimensionality vectors, the formula for which is given in

¹In fact, if the matrix is symmetric and positive semidefinite, the left and right singular vectors will be identical and equivalent to its eigenvectors and the singular values will be its eigenvalues.

Table 3. It is unclear whether the vectors are first scaled by the singular values, S , before computing the cosine, as implied in Deerwester, Dumais, Furnas, Landauer, and Harshman (1990).

Computing the SVD itself is not trivial. For a dense matrix with dimensions $n < m$, the SVD computation requires time proportional to n^2m . This is impractical for matrices with more than a few thousand dimensions. However, LSA co-occurrence matrices tend to be quite sparse and the SVD computation is much faster for sparse matrices, allowing the model to handle hundreds of thousands of words and documents. The LSA similarity ratings tested here were generated using the term-to-term pairwise comparison interface available on the LSA web site (<http://lsa.colorado.edu>).² The model was trained on the Touchstone Applied Science Associates (TASA) “general reading up to first year college” data set, with the top 300 dimensions retained.

2.3 WordNet-based models

WordNet is a network consisting of synonym sets, representing lexical concepts, linked together with various relations, such as synonym, hypernym, and hyponym (Miller et al., 1990). There have been several efforts to base a measure of semantic similarity on the WordNet database, some of which are reviewed in Budanitsky and Hirst (2001), Patwardhan, Banerjee, and Pedersen (2003), and Jarmasz and Szpakowicz (2003). Here we briefly summarize each of these methods. The similarity ratings reported in Section 3 were generated using version 0.06 of Ted Pedersen’s WordNet::Similarity module, along with WordNet version 2.0.

The WordNet methods have an advantage over HAL, LSA, and COALS in that they distinguish between multiple word senses. This raises the question, when judging the similarity of a pair of polysemous words, of which senses to use in the comparison. When given the pair *thick–stout*, most human subjects will judge them to be quite similar because *stout* means strong and sturdy, which may imply that something is thick. But the pair *lager–stout* is also likely to be considered similar because they denote types of beer. In this case, the rater may not even be consciously aware of the adjective sense of *stout*. Consider also *hammer–saw* versus *smelled–saw*. Whether or not we are aware of it, we tend to rate the similarity of a polysemous word pair on the basis of the senses that are most similar to one another. Therefore, the same was done with the WordNet models.

²The document-to-document LSA mode was also tested but the term-to-term method proved slightly better.

WN-EDGE: The Edge method

The simplest WordNet measure is edge counting (Rada et al., 1989), which involves counting the number of *is-a* edges (hypernym, hyponym, or troponym) in the shortest path between nodes. The edge count is actually incremented by one so it really measures the number of nodes found along the path. In order to turn this distance, $d(a,b)$, into a similarity measure, it must be inverted. The WordNet::Similarity package by default uses the multiplicative inverse, but somewhat better results were obtained with this additive inverse function:

$$S(a,b) = \max(21 - d(a,b), 0)$$

WN-HSO: The Hirst and St-Onge method

The Hirst and St-Onge (1998) method uses all of the semantic relationships in WordNet and classifies those relationships as upwards, downwards, or horizontal. It then takes into account both path length, $d(a,b)$, and the number of changes in direction, $c(a,b)$, of the path:

$$S(a,b) = 8 - d(a,b) - c(a,b)$$

Unrelated senses score a 0 and words that share the same sense score a 16.

WN-LCH: The Leacock and Chodorow method

The Leacock and Chodorow (1998) algorithm uses only *is-a* links, but scales the shortest path length by the overall depth of the hierarchy, $D = 16$, and adds a *log* transform:

$$S(a,b) = -\log \frac{d(a,b)}{2D}$$

WN-RES: The Resnik method

Resnik's (1995) measure assumes that the similarity of two concepts is related to the information content, or rarity, of their lowest common superordinate, $\text{lso}(a,b)$. It is defined as:

$$S(a,b) = -\log p(\text{lso}(a,b))$$

where $p()$ is a concept's lexical frequency ratio. If $\text{lso}(a,b)$ does not exist or has 0 probability, $S(a,b) = 0$.

WN-JCN: The Jiang and Conrath method

The Jiang and Conrath (1997) measure also uses the frequency of the $\text{lso}(a,b)$, but computes a distance measure by scaling it by the probabilities of the individual words:

$$d(a,b) = \log \frac{p(\text{lso}(a,b))^2}{p(a)p(b)}$$

The standard method for converting this into a similarity metric is the multiplicative inverse. However, this creates problems for words that share the same concept, and thus have $d(a,b) = 0$. As with WN-EDGE, we have obtained better performance with the additive inverse:

$$S(a,b) = \max(24 - d(a,b), 0)$$

WN-LIN: The Lin method

The Lin (1997) measure is very similar to that of Jiang and Conrath (1997) but combines the same terms in a different way:

$$S(a,b) = \frac{\log p(\text{lso}(a,b))^2}{\log(p(a)p(b))}$$

WN-WUP: The Wu and Palmer method

Wu and Palmer (1994) also make use of the lowest common superordinate of the two concepts, but their similarity formula takes the ratio of the depth of this node from the top of the tree, $d(\text{lso}(a,b))$, to the average depth of the concept nodes:

$$S(a,b) = \frac{2 d(\text{lso}(a,b))}{d(a) + d(b)}$$

WN-LESK: The Adapted Lesk method

The Adapted Lesk method (Banerjee & Pedersen, 2002) is a modified version of the Lesk algorithm (Lesk, 1986), for use with WordNet. Each concept in WordNet has a *gloss*, or brief definition, and the Lesk algorithm scores the similarity of two concepts by the number of term overlaps in their glosses. The adapted Lesk algorithm expands these glosses to include those for all concepts linked to by the original concepts, using most but not all of the link types. It also gives a greater weighting to multi-word sequences shared between glosses, by scoring each sequence according to the square of its length. WN-LESK differs from the other WordNet methods in that it does not make primary use of the network's link structure.

2.4 The Roget's Thesaurus model

Jarmasz and Szpakowicz (2003) have developed a semantic similarity measure that is akin to the edge-counting WordNet models, but which instead makes use of the 1987 edition of Penguin's *Roget's Thesaurus of English Words and Phrases*. Unlike WordNet, the organization of Roget's Thesaurus is more properly a taxonomy. At the highest taxonomic level are 8 classes, followed by 39 sections, 79 sub-sections, 596 head groups, and 990 heads. Each head is then divided into parts of speech, these into paragraphs, and paragraphs into semicolon groups. The

similarity between two concepts is simply determined by the level of the lowest common subtree that contains both concepts. Concepts that share a semicolon group have similarity 16. Those that share a paragraph have similarity 14, 12 for a common part of speech, 10 for a common head, on down to 2 for a common class and 0 otherwise. As in the WordNet models, polysemous words are judged on the basis of their most similar sense pair.

2.5 The COALS model

The main problem with HAL, as we'll see, is that the high frequency, or high variance, columns contribute disproportionately to the distance measure, relative to the amount of information they convey. In particular, the closed class or function words tend to be very frequent as neighbors, but they convey primarily syntactic, rather than semantic, information. In our corpus of written English (see Section 2.7), the frequency of the most common word, *the*, is 34 times that of the 100th most common word, *well*, and 411 times that of the 1000th most common word, *cross*. Under the HAL model, a moderate difference in the frequency with which two words co-occur with *the* will have a large effect on their inter-vector distance, while a large difference in their tendency to co-occur with *cross* may have a relatively tiny effect on their distance. In order to reduce the undue influence of high frequency neighbors, the COALS method employs a normalization strategy that largely factors out lexical frequency.

The process begins by compiling a co-occurrence table in much the same way as in HAL, except that we ignore the left/right distinction so there is just a single column for each word. We also prefer to use a narrower window in computing the weighted co-occurrences. Rather than a ramped, 10-word window, a ramped 4-word window is usually employed, although a flat 4-word window works equally well. Neighbor *b* receives a weighting of 4 if it is adjacent to *a*, 3 if it is two words from *a*, and so forth. Table 5 shows the initial co-occurrence table computed on the Woodchuck corpus using a size 4 window. Note that the table is symmetric. In actuality, we normally compute the table using 100,000 columns, representing the 100,000 most frequent words, and 1 million rows, also ordered by frequency. This large, sparse matrix is filled using dynamic hash tables to avoid excess memory usage.

As Burgess and Lund found, it is possible to eliminate the majority of these columns with little degradation in performance, and often some improvement. But rather than discarding columns on the basis of variance, we have found it simpler and more effective to discard columns on the basis of word frequency. Columns representing low-frequency words tend to be noisier because they involve fewer samples. As we will see in Section 4,

roughly equivalent performance is obtained by using anywhere from 14,000 to 100,000 columns. Performance declines slowly as we reduce the vectors to 6,000 columns and then more noticeably with just a few thousand. Unless otherwise noted, the results reported in Section 3 are based on vectors employing 14,000 columns.

Of primary interest in the co-occurrence data is not the raw rate of word-pair co-occurrence, but the conditional rate. That is, does word *b* occur more or less often in the vicinity of word *a* than it does in general? One way to express this tendency to co-occur is by computing Pearson's correlation between the occurrence of words *a* and *b*. Imagine that we were to make a series of observations, in which each observation involves choosing one word at random from the corpus and then choosing a second word from the weighted distribution over the first word's neighbors. Let x_a be a binary random variable that has value 1 whenever *a* is the first word chosen and let y_b be a binary random variable that has value 1 whenever *b* is the second word chosen. If $w_{a,b}$ records the number of co-occurrences of x_a and y_b , then the coefficient of correlation, or just *correlation* for short, between these variables can be computed using the formula given in Table 4.

When using this correlation normalization, the new cell values, $w'/a,b$, will range from -1 to 1. A correlation of 0 means that x_a and y_b are uncorrelated and word *b* is no more or less likely to occur in the neighborhood of *a* than it is to occur in the neighborhood of a random word. A positive correlation means that *b* is more likely to occur in the presence of *a* than it would otherwise. Table 6 shows the raw co-occurrence counts from Table 5 transformed into correlation values. Given a large corpus, the correlations thus computed tend to be quite small. It is rare for a correlation coefficient to be greater in magnitude than 0.01. Furthermore, the majority of correlations, 81.8%, are negative, but the positive values tend to be larger in magnitude (averaging $1.3e^{-4}$) than the negative ones (averaging $-2.8e^{-5}$).

It turns out that the negative correlation values actually carry very little information. This makes some sense if we think about the distribution of words in natural texts. Although some words are used quite broadly, most content words tend to occur in a limited set of topics. The occurrence of such a word will be strongly correlated, relatively speaking, with the occurrence of other words associated with its topics. But the majority of words are not associated with one of these topics and will tend to be mildly anti-correlated with the word in question. Knowing the identity of those words is not as helpful as knowing the identity of the positively correlated ones. To illustrate this point another way, imagine that we were to ask you to guess a word. Would you rather be told 10 words associated with the mystery word (*cat, bone, paw, collar...*), or 100 words that have nothing to do with the mystery

Table 5

Step 1 of the COALS method: The initial co-occurrence table with a ramped, 4-word window.

	a	as	chuck	could	how	if	much	wood	woodch.	would	,	.	?
a	0	5	9	6	1	10	4	8	18	9	10	0	0
as	5	4	2	1	0	0	7	10	3	2	1	0	5
chuck	9	2	0	8	0	5	1	9	11	2	4	3	3
could	6	1	8	0	0	4	0	6	8	0	2	2	2
how	1	0	0	0	0	0	4	3	0	2	0	0	0
if	10	0	5	4	0	0	0	0	10	3	8	0	0
much	4	7	1	0	4	0	0	10	2	3	0	0	3
wood	8	10	9	6	3	0	10	2	8	5	0	4	6
woodch.	18	3	11	8	0	10	2	8	0	8	10	1	1
would	9	2	2	0	2	3	3	5	8	0	5	0	0
,	10	1	4	2	0	8	0	0	10	5	0	0	0
.	0	0	3	2	0	0	0	4	1	0	0	0	0
?	0	5	3	2	0	0	3	6	1	0	0	0	0

Table 6

Step 2 of the COALS method: Raw counts are converted to correlations.

	a	as	chuck	could	how	if	much	wood	woodch.	would	,	.	?
a	-0.167	-0.014	0.014	0.009	-0.017	0.085	-0.018	-0.033	0.096	0.069	0.085	-0.055	-0.079
as	-0.014	0.031	-0.048	-0.049	-0.037	-0.077	0.133	0.103	-0.054	-0.021	-0.050	-0.037	0.133
chuck	0.014	-0.048	-0.113	0.094	-0.045	0.021	-0.061	0.031	0.048	-0.046	-0.002	0.088	0.031
could	0.009	-0.049	0.094	-0.075	-0.037	0.033	-0.070	0.022	0.049	-0.075	-0.021	0.069	0.023
how	-0.017	-0.037	-0.045	-0.037	-0.018	-0.037	0.192	0.070	-0.055	0.069	-0.037	-0.018	-0.026
if	0.085	-0.077	0.021	0.033	-0.037	-0.077	-0.071	-0.106	0.085	0.006	0.138	-0.037	-0.053
much	-0.018	0.133	-0.061	-0.070	0.192	-0.071	-0.065	0.128	-0.061	0.019	-0.071	-0.034	0.072
wood	-0.033	0.103	0.031	0.022	0.070	-0.106	0.128	-0.113	-0.033	0.001	-0.106	0.111	0.100
woodch.	0.096	-0.054	0.048	0.049	-0.055	0.085	-0.061	-0.033	-0.167	0.049	0.085	-0.017	-0.051
would	0.069	-0.021	-0.046	-0.075	0.069	0.006	0.019	0.001	0.049	-0.075	0.060	-0.037	-0.053
,	0.085	-0.050	-0.002	-0.021	-0.037	0.138	-0.071	-0.106	0.085	0.060	-0.077	-0.037	-0.053
.	-0.055	-0.037	0.088	0.069	-0.018	-0.037	-0.034	0.111	-0.017	-0.037	-0.037	-0.018	-0.026
?	-0.079	0.133	0.031	0.023	-0.026	-0.053	0.072	0.100	-0.051	-0.053	-0.053	-0.026	-0.037

Table 7

Step 3 of the COALS method: Negative values discarded and the positive values square rooted.

	a	as	chuck	could	how	if	much	wood	woodch.	would	,	.	?
a	0	0	0.120	0.093	0	0.291	0	0	0.310	0.262	0.291	0	0
as	0	0.175	0	0	0	0	0.364	0.320	0	0	0	0	0.365
chuck	0.120	0	0	0.306	0	0.146	0	0.177	0.220	0	0	0.297	0.175
could	0.093	0	0.306	0	0	0.182	0	0.149	0.221	0	0	0.263	0.151
how	0	0	0	0	0	0	0.438	0.265	0	0.263	0	0	0
if	0.291	0	0.146	0.182	0	0	0	0	0.291	0.076	0.372	0	0
much	0	0.364	0	0	0.438	0	0	0.358	0	0.136	0	0	0.268
wood	0	0.320	0.177	0.149	0.265	0	0.358	0	0	0.034	0	0.333	0.317
woodch.	0.310	0	0.220	0.221	0	0.291	0	0	0	0.221	0.291	0	0
would	0.262	0	0	0	0.263	0.076	0.136	0.034	0.221	0	0.246	0	0
,	0.291	0	0	0	0	0.372	0	0	0.291	0.246	0	0	0
.	0	0	0.297	0.263	0	0	0	0.333	0	0	0	0	0
?	0	0.365	0.175	0.151	0	0	0.268	0.317	0	0	0	0	0

word (*whilst, missile, suitable, cloud...*)? Odds are, the 10 positively correlated words would be much more helpful.

Another problem with negative correlation values that are large in magnitude is that they are based on a small number of observations, or, more appropriately, on a small number of non-observations. The presence or absence of a single word-pair observation may be the difference between a strongly negative correlation and a slightly negative correlation. Therefore, strongly negative correlations are generally less reliable than strongly positive ones. If we simply eliminate all of the negative correlations, setting them to 0, the performance of the model actually improves somewhat. This also has the effect of making the normalized co-occurrence matrix sparser than the matrices produced by the other normalization strategies, which will be a useful property when we wish to compute the SVD.

The next problem is that most of the positive correlation values are quite small. The interesting variation is occurring in the $1e-5$ to $1e-3$ range. Therefore, rather than using the straight positive correlation values, their square roots are used. This is not a terribly principled maneuver, but it has the beneficial effect of magnifying the importance of the many small values relative to the few large ones. Figure 7 shows the example table once the negative values have been set to 0 and the positive values have been square rooted.

If one is interested in obtaining vectors that best reflect lexical semantics, it may be necessary to reduce the influence of information related to other factors, such as syntax. As we'll see, when human subjects perform semantic similarity judgment tasks, they rely only moderately on syntactic properties such as word class. One key source of syntactic information in the co-occurrence table is carried by the columns associated with the function words, including punctuation. The pattern with which a word co-occurs with these closed class words mainly reflects syntactic type, rather than "purely semantic" information. If one were interested in classifying words by their roles, a reasonable approach would be to focus specifically on these columns. But by eliminating the closed class columns, a slightly better match to human semantic similarity judgments is obtained. The closed class list that we use contains 157 words, including some punctuation and special symbols. Therefore, we actually make use of the top 14,000 *open-class* columns.

In order to produce similarity ratings between pairs of word vectors, the HAL method uses Euclidean or sometimes city-block distance (see Table 3), but these measures do not translate well into similarities, even under a variety of non-linear transformations. LSA, on the other hand, uses vector cosines, which are naturally bounded in the range $[-1, 1]$, with high values indicating similar vectors. For COALS, we have found the correlation measure to

be somewhat better. Correlation is identical to cosine except that the mean value is subtracted from each vector component. In cases such as this where the vectors are confined to the positive hyperquadrant, correlations tend to be more sensitive than cosines. Therefore, the COALS method makes use of correlation both for normalization and for measuring vector similarity.

Summary of the COALS method

1. Gather co-occurrence counts, typically ignoring closed-class neighbors and using a ramped, size 4 window:

1	2	3	4	0	4	3	2	1
---	---	---	---	---	---	---	---	---
2. Discard all but the m (14,000, in this case) columns reflecting the most common open-class words.
3. Convert counts to word pair correlations, set negative values to 0, and take square roots of positive ones.
4. The semantic similarity between two words is given by the correlation of their vectors.

2.6 COALS-SVD: Reduced-dimensionality and binary vectors

For many applications, word vectors with more than a few hundred dimensions are impractical. In some cases, such as the training of certain neural networks, binary-valued vectors are needed. Therefore, COALS has been extended to produce relatively low-dimensional real-valued and binary vectors.

Rohde (2002b) designed and analyzed several methods for binary multi-dimensional scaling and found the most effective methods to be those based on gradient descent and on bit flipping. The one method that proved least effective made use of the singular value decomposition, which is the basis for LSA. However, the current tasks require that we be able to scale the vectors of several hundred thousand words and the problem is therefore considerably larger than those tested in Rohde (2002b). The running time of the gradient descent and bit flipping algorithms is at least quadratic in the number of words, and using them on a problem of this size would be impractical. Therefore, like LSA, we rely on the SVD to generate reduced-dimensionality real-valued and binary vectors.³

Ideally, the SVD would be computed using the full matrix of COALS word vectors. However, this is computationally difficult and isn't necessary. Good results can be obtained using several thousand of the most frequent words. In the results reported here, we use 15,000

³To compute the SVD of large, sparse matrices, we use the SVDLIBC programming library, which was adapted by the first author from the SVDPACKC library (Berry, Do, O'Brien, Krishna, & Varadhan, 1996).

word vectors, or rows, with 14,000 (open-class) columns in each vector.

Once the SVD has been computed, a k -dimensionality vector for word c is generated by computing $X_c \hat{V} \hat{S}^{-1}$, where X_c is the COALS vector for word c (see Figure 1). The \hat{S}^{-1} term removes the influence of the singular values from the resulting vector and failing to include it places too much weight on the first few components. Although we are not certain, it is possible that LSA does not include this term. Our method for producing reduced-dimensionality vectors in this way will be referred to as COALS-SVD.

Although LSA measures similarity using the cosine function, we again prefer correlation (see Table 3). In this case, it actually makes only a small difference because the vector components tend to be fairly evenly distributed around 0. This fact also makes possible a simple method for producing binary-valued vectors. To convert a real-valued k -dimensional vector to a bit vector, negative components are converted to 0 and positive components to 1. This binary variant will be labeled the COALS-SVDB model. The effect of dimensionality in COALS-SVD and COALS-SVDB is explored in Section 4.3.

2.7 Preparing the corpus

COALS could be applied to virtually any text corpus. However, for the purpose of these experiments it seemed most appropriate to train the model on a large and diverse corpus of everyday English. Following Burgess and Lund (1997a), we chose to construct the corpus by sampling from the online newsgroup service known as Usenet. This service carries discussions on a wide range of topics with many different authors. Although Usenet topics are skewed towards computer issues (mice are more likely to be hardware than rodents) and are not necessarily representative of the full range of topics people encounter on a daily basis, it is certainly better in this respect than most other corpora.

Unfortunately, the data available on Usenet, although copious, requires some filtering, starting with the choice of which newsgroups to include. We tried to include as many groups as possible, although certain ones were eliminated if their names indicated they were likely to contain primarily binaries, were sex related, dealt with purely technical computer issues, or were not in English. Approximately one month's worth of Usenet data was collected from several public servers. The text was then cleaned up in the following sequence of steps:

1. Removing images, non-ascii codes, and HTML tags.
2. Removing all non-standard punctuation and separating other punctuation from adjacent words.
3. Removing words over 20 characters in length.

4. Splitting words joined by certain punctuation marks and removing other punctuation from within words.
5. Converting to upper case.
6. Converting \$5 to 5 *DOLLARS*.
7. Replacing URLs, email addresses, IP addresses, numbers greater than 9, and emoticons with special word markers, such as `<URL>`.
8. Discarding articles with fewer than 80% real words, based on a large English word list. This has the effect of filtering out foreign text and articles that primarily contain computer code.
9. Discarding duplicate articles. This was done by computing a 128-bit hash of the contents of each article. Articles with identical hash values were assumed to be duplicates.
10. Performing automatic spelling correction.
11. Splitting the hyphenated or concatenated words that do not have their own entries in a large dictionary but whose components do.

The automatic spelling correction itself is a somewhat involved procedure. The process begins by feeding in a large list of valid English words. Then, for each misspelled word in the corpus, the strongest candidate replacement is determined based on the probability that the intended word would have been misspelled to produce the actual word. Computing this probability involves a dynamic programming algorithm in which various mistakes—dropping or inserting a letter, replacing one letter with another, or transposing letters—are each estimated to occur with a particular probability, which is a function of their phonological similarity and proximity on the keyboard. The candidate replacement word is suggested along with the computed probability of making the given error, the ratio of this probability to the sum of the probabilities from all candidate replacements—a measure of the certainty that this is the correct replacement—and the candidate word's frequency.

Of course, not all candidate replacements are correct. In fact, the majority are not because the Usenet text contains a large number of proper names, acronyms, and other obscure words. Therefore, we must decide when it is appropriate to replace an unfamiliar word and when it should be left alone. This was done by means of a neural network which was trained to discriminate between good and bad replacements on the basis of the three statistics computed when the candidate word was generated. One thousand candidate replacements were hand coded as good or bad replacements to serve as training data for this network. Only when the trained network judged a candidate to be good enough was a word actually corrected. In the future, this spelling correction method could probably be

improved with the addition of a language model or the use of something like COALS itself.

At the end of the cleanup process, the resulting corpus contains 9 million articles and a total of 1.2 billion word tokens, representing 2.1 million different word types. Of these, 995,000 occurred more than once and 325,000 occurred 10 or more times. The 100,000th word occurred 98 times.

3 Evaluation of the methods

In this section, we evaluate the performance of COALS, alongside the other models described in Section 2, on several tasks. Section 3.1 compares the models' ratings to those of humans on four word-pair similarity tasks. Section 3.2 evaluates the models on multiple-choice synonym matching vocabulary tests. Then, to provide a better understanding of some of the properties of COALS, Section 3.4 presents visualizations of word similarities employing multi-dimensional scaling and hierarchical clustering and Section 3.5 examines the nearest neighbors of some selected words.

3.1 Word-pair similarity ratings

One measure of the ability of these methods to model human lexical semantic knowledge is a comparison of a method's similarity ratings with those of humans on a set of word pairs. Of particular interest is the relative similarity humans and the models will find between word pairs exhibiting various semantic, morphological, and syntactic differences. In this section, we test the models on four word-pair similarity tasks: the well-known Rubenstein and Goodenough (1965) and Miller and Charles (1991) lists, the WordSimilarity-353 Test Collection (Finkelstein et al., 2002), and a new survey we have conducted involving 400 word pairs exhibiting 20 types of semantic or morphological pairings.

On each of these sets, a model's ratings will be scored on the basis of their correlation with the average human ratings of the same items. However, different models or raters may apportion their scales differently. Even if two models agree on the rank order of word pairs, one may be more sensitive at the high end, predicting greater differences between nearly synonymous pairs than between nearly unrelated pairs, and others more sensitive at the low end. Therefore, because humans are also likely to apportion their ratings scale differently given its resolution or the average similarity of the experimental items, it isn't clear what the proper scaling should be and a straight correlation could unfairly bias the results against certain models. One solution is to test them using rank-order correlation, which relies only on the relative ordering of the

pairs. However, this requirement seems a bit too loose, as models are much more useful and informative if they are able to produce an exact numerical estimate of the similarity of a given word pair.

Therefore, the models were measured using the best-fit exponential scaling of their similarity scores.⁴ Any scores less than 0 were set to 0,⁵ while positive scores were replaced by $S(a,b)^t$, where $S(a,b)$ is the model's predicted similarity of words a and b and t is the exponent that maximizes the model's correlation with the human ratings, subject to the bounds $t \in [1/16, 16]$. If $t = 1$, this is the identity function. If $t > 1$, this increases the sensitivity at the high end of the ratings scale and if $t < 1$ it increases the sensitivity at the low end. COALS typically uses t values between 1/2 (square root) and 1/3 (cube root), while LSA uses t values between 1/1.5 and 1/2.5. HAL, on the other hand, typically prefers t values between 4 and 8. Scaling preferences for the WordNet-based models range from strongly positive for WN-EDGE, to moderately negative for WN-LESK, with most in positive territory. For easier comparison with other scores, the correlation coefficients throughout this paper will be expressed as percentages (multiplied by 100).

In cases in which a word was not recognized by a model, any ratings involving that word were assigned the mean rating given by the model to all other word pairs. We will note how often this was necessary for each model and task.

Table 8 gives the scores of all of the models on the word similarity tasks discussed here and the vocabulary tasks discussed in Section 3.2. COALS-14K is the COALS model using 14,000 dimensions per vector, while COALS-SVD-800 is the COALS-SVD model with 800 dimensions per vector. We will explain each of the tasks in turn. The overall score, given in the right-most column, is a weighted average of the scores on all of the tasks. These weightings, given in the top row of Table 8, are based on the relative sizes of the tasks and our own interest in them. Although these weightings are subjective, the weighted scores may be a convenient reference given the large number of models and tasks.

WS-RG: The Rubenstein and Goodenough ratings

One of the most popular word similarity benchmarks is the Rubenstein and Goodenough (1965) set, in which 65 noun pairs were scored on a 0 to 4 scale by 51 human raters. The words used in this task are moderately com-

⁴Correlations based on the best-fit exponential or on rank order actually agree quite well. If rank order were used instead, the overall scores given in Table 8 would decrease by less than 2 points for all of the models except WN-EDGE, WN-LCH, WN-RES, WN-JCN, WN-WUD, and WN-LIN, whose scores all decrease by 5–7 points.

⁵Only COALS and LSA can produce negative similarity scores and these are relatively rare.

Table 8

Performance of various models on lexical-semantic tasks. WS- tasks involve word similarity rating and scores represent exponential best-fit percent correlation with human ratings. VT- tasks are multiple-choice synonym-matching vocabulary tests and scores indicate percent correct. The Overall column is a weighted average of the other scores.

Model	WS-RG	WS-RG-ND	WS-MC	WS-MC-ND	WS-353	WS-400	WS-400-NV	WS-400-ND	WS-400-PT	VT-TOEFL	VT-TOEFL-NV	VT-ESL	VT-ESL-NV	VT-RDWP	VT-RDWP-NV	VT-ALL	VT-ALL-NV	Overall
Task Weightings	8%	3%	8%	3%	12%	10%	5%	5%	10%	6%	4%	6%	4%	10%	6%	–	–	=100%
COALS-14K	68.2	79.1	67.1	85.2	62.6	62.4	60.9	71.2	88.0	86.2	81.6	52.0	52.5	65.5	67.4	67.8	67.2	69.2
COALS-SVD-800	67.3	77.9	72.7	82.5	65.7	68.4	67.5	74.2	91.6	88.8	86.8	68.0	67.5	66.8	69.2	70.8	71.3	73.4
COALS-SVD-200	64.7	76.6	68.0	82.7	65.3	64.9	65.0	73.8	88.4	86.2	84.2	58.0	55.0	60.8	60.3	65.0	62.7	69.4
HAL-14K	14.6	23.1	25.6	35.6	28.2	13.6	11.4	21.5	24.8	56.2	47.4	26.0	27.5	37.9	37.5	39.7	37.4	27.8
HAL-400	15.3	25.7	31.9	41.9	31.1	8.8	7.1	15.2	14.0	53.8	47.4	26.0	27.5	35.7	35.0	37.7	35.6	26.4
LSA	65.6	70.7	73.1	79.5	59.9	66.0	66.7	71.1	90.9	53.4	48.7	43.0	47.4	40.6	42.1	43.2	43.9	61.6
WN-EDGE	86.4	86.1	82.8	80.9	39.0	38.4	47.9	41.3	55.1	44.9	68.6	66.5	78.2	53.8	66.0	53.6	68.1	58.9
WN-HSO	80.4	78.6	74.9	72.2	34.6	46.0	46.8	48.7	72.5	67.8	63.2	67.7	73.9	57.6	56.6	60.6	60.0	60.5
WN-LCH	85.2	84.5	81.6	79.5	37.9	38.2	46.6	41.1	55.2	45.5	68.6	66.5	78.2	54.2	66.5	54.0	68.4	58.5
WN-RES	81.3	80.8	78.4	73.6	37.7	34.0	42.1	37.5	52.6	43.0	64.7	58.5	68.0	47.8	57.5	48.2	59.6	54.2
WN-JCN	76.6	76.6	77.1	78.4	35.1	38.7	48.1	42.2	57.2	42.0	63.8	63.5	76.9	39.5	45.7	42.8	52.4	53.9
WN-LIN	76.7	76.4	76.4	76.8	37.1	38.3	47.6	42.0	57.5	41.1	62.1	63.5	76.9	38.2	44.2	41.7	51.1	53.7
WN-WUP	83.7	84.0	80.5	78.6	37.6	33.0	41.9	39.2	48.6	43.0	63.4	52.2	59.8	50.6	61.1	49.4	61.3	54.1
WN-LESK	70.1	70.2	74.2	75.5	36.6	40.6	44.1	43.9	65.5	79.7	71.0	58.0	64.1	68.0	68.7	69.0	68.5	59.9
ROGET	84.6	86.2	80.3	83.1	51.0	57.0	58.9	57.0	76.6	74.6	76.1	78.7	82.5	69.7	68.9	71.6	71.6	70.2

mon, with a geometric mean (GM) frequency of 9.75 per million in our Usenet corpus. According to the exponential best-fit correlation measure, COALS-14K scores a moderately strong 68.2% on this task. The COALS-SVD models are somewhat worse, as is the LSA model. However, the HAL models score very poorly, at just 14.6% and 15.3%. The WordNet and Roget’s models all perform very well on these pairs, with the simple WN-EDGE actually achieving the best score of 86.4%. These results are shown in graphical form in Figure 2.

A potential source of confusion for COALS and the other vector-based models is the existence of multiple senses, and even multiple syntactic categories, for many orthographically-defined words. As language users, we are generally untroubled by—and often unaware of—the existence of multiple possible meanings of the words we use because we are able to rely on context to properly disambiguate a word’s meaning without necessarily calling to mind the competing senses. The lexicon-based models have the advantage of starting with knowledge about distinct word senses and they can compare words by pairing their most similar senses, as people seem to do. In contrast, a vector produced by COALS, HAL, and LSA necessarily reflects an averaging of all of a word’s senses. It turns out, as we will discuss later, that this is generally not a significant problem if the desired sense is dominant or even if the frequency of the various senses is balanced. But when the correct sense is much less frequent than an

incorrect sense, the models’ performance is likely to drop.

We tested this by identifying and removing five words (and the 13 pairs involving them) from the WS-RG set, producing the 52-pair WS-RG-ND set. Words were removed whose dominant sense in the Usenet corpus (given here in parentheses) is not the sense most likely employed in the synonymy judgments, including: *jewel* (the singer), *crane* (a machine, rather than a bird), *oracle* (the company), *madhouse* (connoting general mayhem, rather than an asylum), and *cock* (not a rooster). As expected, this resulted in little change in performance for the WordNet and Roget’s methods, but significant improvements for the vector-based methods. On this reduced set, COALS-14K outperforms four of the eight WordNet models.

WS-MC: The Miller and Charles ratings

Another commonly used similarity benchmark is that of Miller and Charles (1991), who selected 30 of the word pairs from the Rubenstein and Goodenough (1965) set and collected ratings from 38 subjects. The words in this subset are somewhat higher in frequency than the full WS-RG set, with a GM frequency of 11.88 per million.

As we might expect, the results with this set largely parallel those for the WS-RG task. The performance of most of the vector-based models as well as that of ROGET improves somewhat, while that of the WordNet models declines, although WN-EDGE is still the best. The WS-

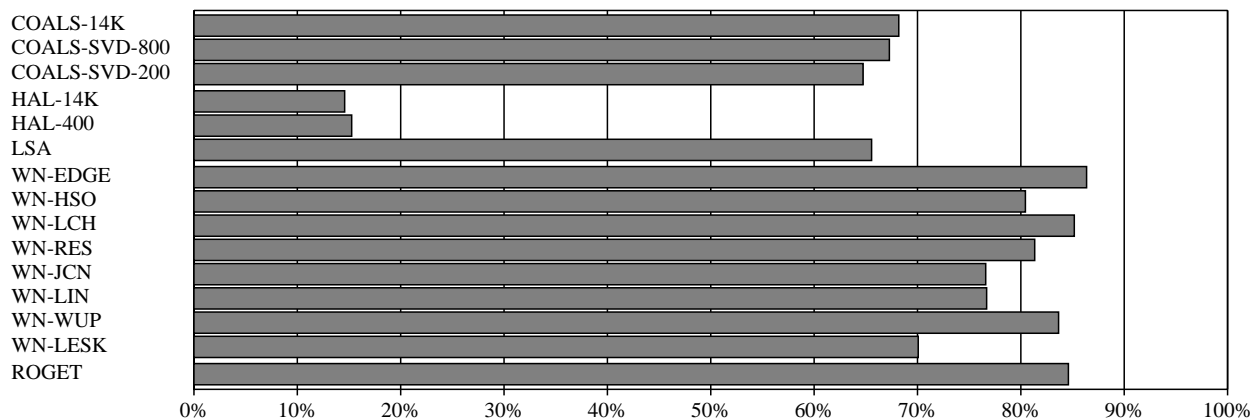


Figure 2: Performance of the models on the Rubenstein and Goodenough (WS-RG) task.

MC-ND set was produced by eliminating the six pairs involving the same five problematic words dominated by an incorrect sense. Again, the performance of the lexicon-based models remains about the same, but the vector-based models all improve considerably. On this sub-task, COALS-14K actually achieves the highest score, followed by ROGET and then the two COALS-SVD models. HAL, although better than on the full WS-MC task, still lags well behind the others.

WS-353: The Finkelstein et al. ratings

The WordSimilarity-353 Test Collection (Finkelstein et al., 2002) is a set of 353 pairs, including the 30 WS-MC pairs, rated by either 13 or 16 subjects on a 0 to 10 scale. This collection includes some proper names and cultural references, such as *Arafat-terror*, and also some word associates that are not necessarily synonymous, such as *tennis-racket*. Although most of the words are nouns, there are some adjectives and a few gerunds. The words in the WS-353 set tend to be more common than the Rubenstein and Goodenough words, with a GM frequency of 25.69 per million. A word was unfamiliar in seven of the WS-353 pairs for LSA, in one for the WordNet models, and in 22 for the Roget’s model. In each of these cases, missing data was replaced with a model’s average rating.

The models’ results on the WS-353 task are shown in Figure 3 and listed in Table 8. The COALS and LSA models all perform quite well on this task, with the highest score of 65.7% achieved by COALS-SVD-800. However, the WordNet and Roget’s models perform much worse on the WS-353 set than they did on the previous tasks, with scores in the 34% to 39% range, although ROGET is well ahead of the others. The WordNet models tend to underestimate human judgments of the semantic similarity of associated or domain-related word-pairs, such as *psychology-Freud*, *closet-clothes*, and *computer-software*.

WS-400: Morphological and semantic word-pair classes

The three word similarity tasks that we have just discussed were limited mainly to noun-noun pairs of varying synonymy. However, we were also interested in the degree to which human and model similarity judgments are affected by other factors, including syntactic role, morphology, and phonology.

Therefore, a survey was developed to determine the semantic similarity of 400 pairs of words representing a range of lexical relationships (as shown in Table 9). Twenty different types of relationship were included, with 20 pairs of words for each type. Some pairs were morphologically related (e.g., *teacher-teach*), some were members of the same taxonomic category (*apple-pear*), some were synonyms (*dog-hound*), some shared only phonological similarity (*catalog-cat*), and others were dissimilar both in meaning and in sound (*steering-cotton*).

The word pairs were divided into 10 lists with 40 words on each list, 2 from each category of lexical relationship. The 10 lists were administered to 333 Carnegie Mellon undergraduates, such that each word pair was rated by an average of 33 participants. Participants were asked to rate the word pairs on a scale from 1 (very dissimilar) to 9 (very similar) and were encouraged to use the entire scale. The instructions included examples of highly similar, moderately similar, and dissimilar pairs, and reminded participants that some words sound alike but nevertheless have quite different meanings (e.g., *ponder-pond*). Table 9 shows the mean similarity ratings and frequency for each type of word pair (Kucera & Francis, 1967).

The WS-400 words tend to be relatively common, with a GM frequency of 18.46 per million, just under twice that of the Rubenstein and Goodenough (1965) words. The scores of the models using all 400 of these pairs are given in Figure 4 and Table 8. The outcome is similar to that of the WS-353 task. COALS and LSA perform the best,

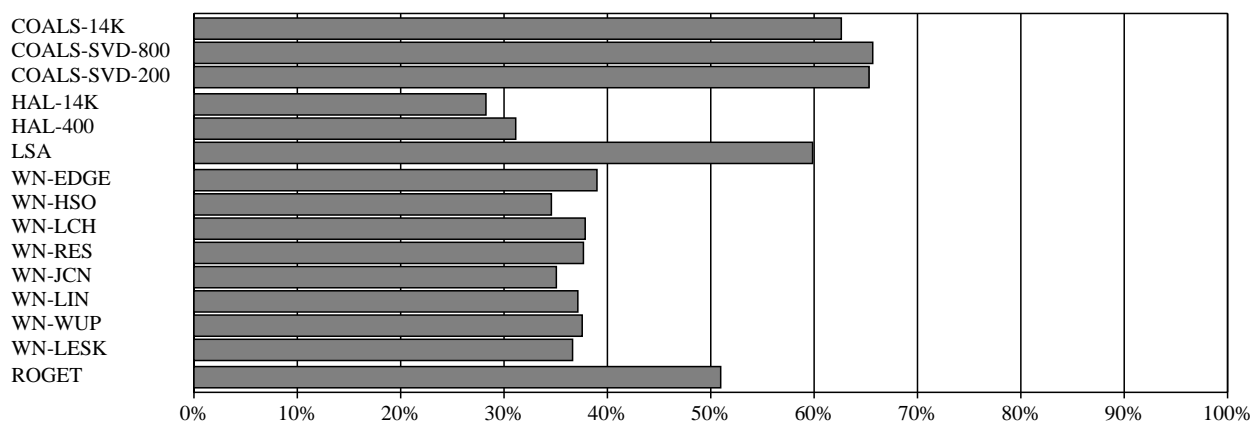


Figure 3: Performance of the models on the Finkelstein et al. (WS-353) task.

Table 9

The 20 WS-400 word pair types for which human similarity ratings were obtained.

Type	Example	Description	Mean Frequency	Mean Similarity Rating (std. dev.)
A.	<i>similarity–talented</i>	unrelated filler pair	61.8	1.54 (1.04)
B.	<i>fasten–fast</i>	orthographically similar but unrelated foil	49.8	1.63 (0.99)
C.	<i>fish–monkey</i>	distantly related coordinate nouns	23.2	3.08 (1.61)
D.	<i>smoke–eat</i>	distantly related coordinate verbs	47.0	3.64 (1.72)
E.	<i>lion–mane</i>	object and one of its parts	68.3	4.66 (1.81)
F.	<i>cow–goat</i>	closely related coordinate nouns	39.1	5.19 (2.36)
G.	<i>mailman–mail</i>	noun ending in <i>-man</i> and related noun or verb	40.2	5.34 (1.86)
H.	<i>fly–drive</i>	closely related coordinate verbs	41.2	5.43 (1.75)
I.	<i>entertain–sing</i>	superordinate-subordinate verb pair	64.3	5.48 (1.63)
J.	<i>scientist–science</i>	noun or verb ending in <i>-ist</i> and related noun or verb	43.5	5.54 (1.74)
K.	<i>stove–heat</i>	instrument and its associated action	45.4	5.91 (1.77)
L.	<i>musician–play</i>	noun ending in <i>-ian</i> and related noun or verb	58.5	6.22 (1.73)
M.	<i>weapon–knife</i>	superordinate-subordinate noun pair	47.6	6.30 (1.48)
N.	<i>doctor–treat</i>	human agent and associated action	49.1	6.45 (1.65)
O.	<i>famous–fame</i>	adjective and its related noun	50.2	6.73 (1.56)
P.	<i>cry–weep</i>	synonymous verbs	48.9	6.88 (1.55)
Q.	<i>rough–uneven</i>	synonymous adjectives	60.6	7.13 (1.66)
R.	<i>farm–ranch</i>	synonymous nouns	74.6	7.37 (1.34)
S.	<i>speak–spoken</i>	irregular noun or verb inflection	60.0	7.52 (1.55)
T.	<i>monster–monsters</i>	regular noun or verb inflection	48.6	7.71 (1.42)

with COALS-SVD-800 scoring 68.4%, followed by ROGET and then the WordNet models. WN-HSO performs somewhat better than the other WordNet methods on this task. The HAL model performs very poorly on the WS-400 task.

This test is a bit unfair for the WordNet models because they were unable to handle 45 of the word pairs, mainly those containing adjectives. ROGET was unable to handle 12 and the LSA model two. To verify that the inclusion of adjectives was not responsible for the relatively low scores of the WordNet models, the WS-400-NV sub-task was constructed, involving 338 of the 400 pairs using only nouns and verbs. All of these pairs were recognized by the models with the exception of ROGET and LSA, which were each unfamiliar with two pairs. This change increased the performance of the WordNet models by 8 to 10 points, with the exception of WN-HSO and WN-LESK. Nevertheless, none of the WordNet models scored over 50% on the WS-400-NV set.

Some of the WS-400 pairs are more difficult for the vector-based models because they induce human raters to make use of a non-dominant word sense. In order to identify these pairs, three judges were provided with the list of pairs along with a list of the ten nearest neighbors (most similar words) of each word, according to COALS. The judges indicated if the word senses they would use in comparing the two words differ substantially from the senses suggested by the nearest neighbors list. On 90 of the pairs, two of the three judges agreed that a non-dominant sense was involved. The resulting 310 pairs form the WS-400-ND set. Once again, as we might expect, the performance of the vector-based models increases significantly, while that of the WordNet models increases by just a few points, with no change for ROGET.

These results would seem to suggest that the vector based methods are substantially hindered by their lack of word sense distinctions. While that is true to some extent, the overall performance of the COALS and LSA models remains quite high despite this limitation. If these methods were substantially hindered by the interference of inappropriate word senses, we should expect them to perform very poorly on the 90 pairs relying on non-dominant senses that were eliminated in forming the WS-400-ND set. If we test the models on just these 90 pairs, the HAL scores do indeed drop to about -10%, but the COALS and LSA models remain quite strong. COALS-SVD-800 achieves a score of 42.5%, with LSA scoring 41.3% and COALS-14K scoring 37.0%. In contrast, the WordNet models all score between 10.3% (WN-WUP) and 32.8% (WN-HSO). The only model to outperform COALS and LSA is ROGET, scoring 56.0%. Therefore, even on word pairs invoking a non-dominant sense, COALS and LSA continue to perform reasonably well.

A principal motivation for collecting the WS-400 rat-

ings was to discover how similar, overall, human raters find the pairs from each of the 20 pair types listed in Table 9, which manipulate the word class and morphological properties of the words. Therefore, another interesting test of the models is the degree to which they predict the average similarity of each of these 20 pair types. This eliminates much of the noise in the individual pair data and reveals the models' broader trends. A model was scored on the WS-400-PT task by first taking its raw ratings of the WS-400 pairs and computing the exponent, t , that maximizes the correlation with human judgments. Then the adjusted pairwise ratings, $S(a,b)^t$, computed using the optimal t value, were averaged over the 20 pairs in each of the 20 pair types. The correlation between the average human ratings and the average adjusted model ratings over the 20 types determines a model's score. The results are shown in Figure 5 and under the WS-400-PT column in Table 8.

With some of the noise having been removed, the performance of most of the models increases substantially on this task relative to the basic WS-400 test. COALS-SVD-800 and LSA both score over 90%, followed closely by the other COALS models. These models, therefore, are capturing nearly all of the variance in similarity over the word pair classes. ROGET and WN-EDGE score in the 70–80% range, with most of the other WordNet models in the 50–60% range.

Figure 6 shows the individual averaged human ratings for the 20 pair types alongside those of COALS-SVD-800 and WN-EDGE, included because its performance is representative of the distance-based WordNet methods. The models' ratings have been linearly transformed for best alignment. The bars are sorted by increasing human rating.

As is evident from the comparison of the gray and black bars in Figure 6, COALS is most likely to under-rate the similarity of pair types P, N, I, and Q, followed by J, L, O, and K. P are nearly synonymous verbs, I are superordinate-subordinate verbs, and Q are nearly-synonymous adjectives. Therefore, COALS tends to under-rate the similarity of verbs and adjectives. Types N, J, L, O, and K all primarily involve nouns paired with verbs or adjectives. Thus, the model also tends to under-rate the similarity of associates, which may be surprising given that it is based on co-occurring words and nouns are likely to co-occur with related verbs and adjectives.

The pairs whose similarity the model most over-rates are F, E, C, B, and S. Types F, E, and C are all distantly related nouns. Interestingly, unlike the human raters, the model rates set S (irregular inflections) higher than set T (regular inflections). Thus, it is possible that the human raters are influenced by the morphological transparency of the regular inflections in their similarity judgments. It is interesting that most of the model's underestimates occur

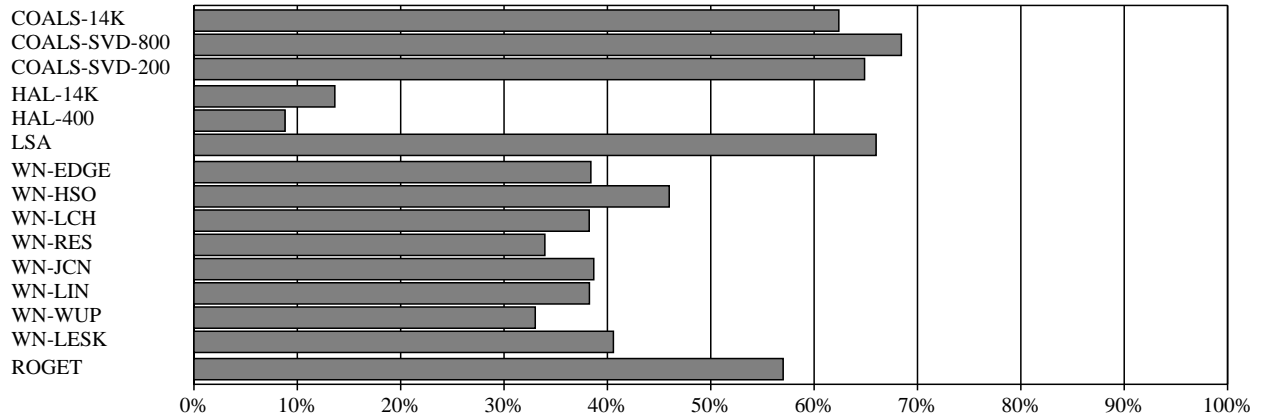


Figure 4: Performance of the models across all word pairs on the WS-400 task.

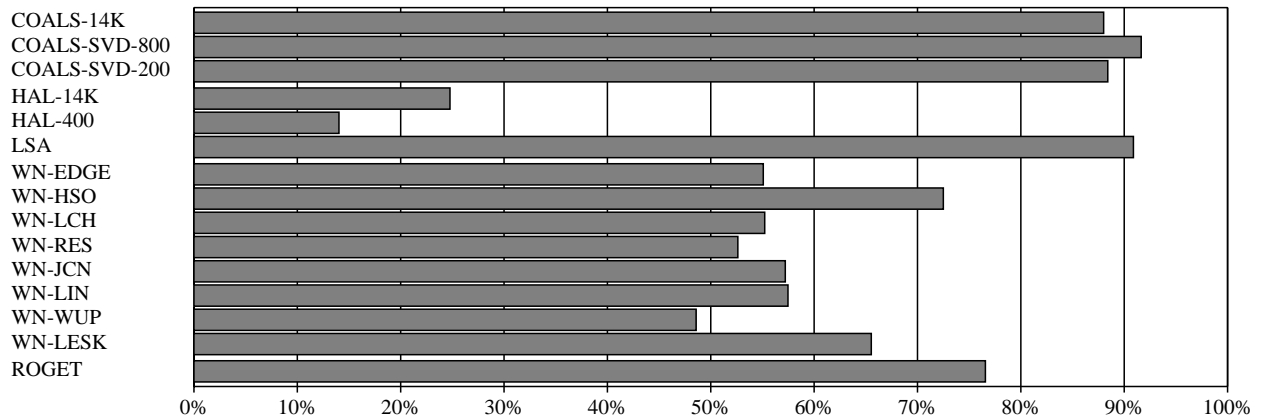


Figure 5: Performance of the models with the ratings averaged over each of the 20 word pair types on the WS-400 task.

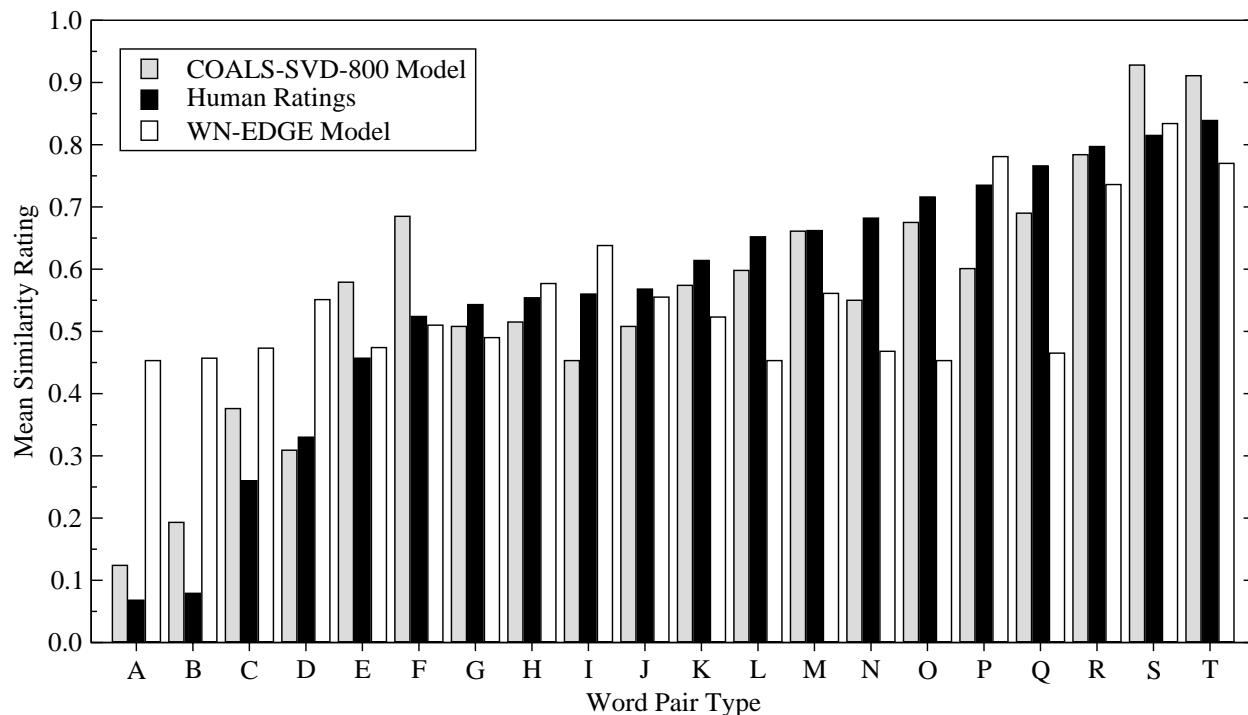


Figure 6: Average similarity ratings (linearly transformed for best alignment) of the 20 word pair types for human subjects, the COALS model, and the WS-EDGE model.

in the middle of the range of pair types and the overestimates at the extremes. This indicates that reducing the t value used in transforming the models' ratings would improve its score in this task. Indeed, reducing t from 0.37 to 0.15 increases the model's correlation score from 91.6% to 93.9%.

The white bars in Figure 6 result from WN-EDGE, which is fairly representative of the WordNet-based methods. This model most underestimates types Q, O, N, L, M, and K. Q and O involve adjectives, which the model does not really handle. N, L, and K involve nouns and their associated verbs. Again, it should not be too surprising that WN-EDGE underestimates the similarity of such pairs because the WordNet 2.0 database has few links between nouns and verbs. However, the underestimate of type M is surprising, as these are superordinate/subordinate nouns, for which WordNet is specialized. The most over-rated pair types are the four at the low end: A, B, D, and C. This would seem to indicate that the WordNet database includes some relatively short pathways between concepts that most people would feel are unrelated or distantly related, but it could be a secondary effect of the low similarity scores between words of different syntactic or semantic class that humans nevertheless feel are moderately related.

3.2 Multiple-choice vocabulary tests

Another form of task previously used in evaluating semantic models is the multiple choice vocabulary test. The items in these tests all consist of a target word or phrase followed by four other words or phrases, and the test taker must choose which of the four options is most similar in meaning to the target. In this section, we present three such tasks, drawn from the Test of English as a Foreign Language, the English as a Second Language test, and the Reader's Digest Word Power quizzes.

The models performed the tests much as a human might. They rated the similarity of each of the choices to the target word and the most similar choice was selected. Correct answers scored 1 and incorrect answers scored 0. In cases where one or more of the options was unfamiliar to the model, it was assumed that the model would choose randomly from among the unfamiliar options along with the best of the familiar ones, and the item was scored according to the expected likelihood of guessing correctly. So if the correct answer and one other option were unknown, the model would score 0.333 for that item.

VT-TOEFL: Test of English as a Foreign Language

The first of the vocabulary tests consists of 80 items drawn from the Educational Testing Service's Test of English as a Foreign Language (TOEFL), first used for model testing

by Landauer and Dumais (1997). The items in this test all consisted of single-word targets and options, such as *concisely*, *prolific*, or *hue*, with a fairly low GM frequency of 6.94 per million. According to Landauer and Dumais (1997), a large sample of foreign college applicants taking this or similar tests scored an average of 64.5% of the items correct.

Table 8 shows the models' results on the VT-TOEFL task. The COALS models achieved the highest scores, with 88.8% for COALS-SVD-800, and 86.2% for the others. WN-LESK and ROGET also did well, with 79.7% and 74.6%, respectively, followed by WN-HSO at 67.8%. The HAL and LSA models scored in the mid 50–60% range and the other WordNet models scored in the low to mid 40's. Landauer and Dumais (1997) reported a score of 64.4% for their model, higher than the 53.4% that we found. The difference may be due to changes in the training corpus, algorithm, or number of dimensions used.

The VT-TOEFL task is not quite fair to the WordNet methods because it includes items involving adjectives and adverbs that are not well-represented in WordNet. Therefore we also tested the VT-TOEFL-NV subset, consisting of just the 38 noun and verb items. In using this subset, the performance of the vector-based models declines slightly, although COALS-SVD-800 is still the best. As expected, the performance of the WordNet models, with the exception of WN-LESK, increases significantly. But they remain well behind ROGET and the COALS models.

VT-ESL: English as a Second Language tests

The second vocabulary test consists of 50 items drawn from the English as a Second Language (ESL) test (Turney, 2001). The ESL words tend to be shorter and higher in frequency (GM frequency 14.17 per million), but the test relies on more subtle discriminations of meaning, such as the fact that *passage* is more similar to *hallway* than to *entrance* or that *stem* is more similar to *stalk* than to *trunk*. In the actual ESL questions, the target words were placed in a sentence context, which often helps disambiguate its meaning, but these contexts were not made available to the models. Therefore, some items were difficult or impossible, such as one in which the correct synonym for *mass* was *lump*, rather than *service* or *worship*.

The vector-based models did not perform as well on the VT-ESL task as they did on VT-TOEFL. This may be because the ESL items often play on the distinction between different senses of a word. The best performance was achieved by ROGET followed by some of the WordNet models and COALS-SVD-800. LSA was at 43.0% and the HAL models were at chance. Once again, a reduced set of the 40 items using only nouns and verbs was also tested. This resulted in improved performance for the

WordNet models and a smaller improvement for ROGET, which again earned the highest score.

VT-RDWP: Reader's Digest Word Power tests

The final vocabulary test consists of 300 items taken from the Reader's Digest Word Power (RDWP) quizzes (Jarmasz & Szpakowicz, 2003). The GM frequency of the words in these quizzes, 6.28 per million, is relatively low and, unlike the other tests, the RDWP targets and options are often multi-word phrases. The function words were removed from the phrases for our tests. These phrases were handled differently in the various models. The COALS and HAL models computed a vector for each phrase by averaging the vectors of its words. For the LSA model, each phrase was treated as a text and the term-by-term similarity of the texts was computed. The WordNet and ROGET models do not have a natural way of dealing with phrases. Therefore, the similarity between phrases was taken to be that of the most similar pair of words spanning them.

The VT-RDWP task proved harder for the models than did the VT-TOEFL and VT-ESL tasks. The best model was again ROGET, followed by WN-LESK and then COALS-SVD-800 and the other COALS models. The other WordNet models and the HAL and LSA models all fared quite poorly. On the subset of 213 items using only nouns and verbs, VT-RDWP-NV, the performance of most of the WordNet models as well as the better COALS models improved. In this case, COALS-SVD-800 had the highest score.

Overall Vocabulary Test Results

Figure 7 and Table 8 show the overall results on a combination of the 430 items in the three vocabulary tests. ROGET had the best score, followed closely by COALS-SVD-800 and WN-LESK. WN-HSO proved better than the other link-based WordNet methods, possibly because it makes use of a larger subset of the available links. The WordNet models were the only ones sensitive to the presence of adjectives and adverbs in the vocabulary tests. With the exception of WN-HSO and WN-LESK, their performance was 9% to 15% higher on the noun/verb items.

The VT-ALL-NV column in Table 8 lists the overall results on the 291 items using only nouns or verbs from the three vocabulary tests. The performance of most of the models on this subset is similar to their performance on the full set of items. Most of the WordNet models improve, but the best of them—WN-EDGE, WN-LCH, and WN-LESK—remain somewhat worse than ROGET and COALS-SVD-800.

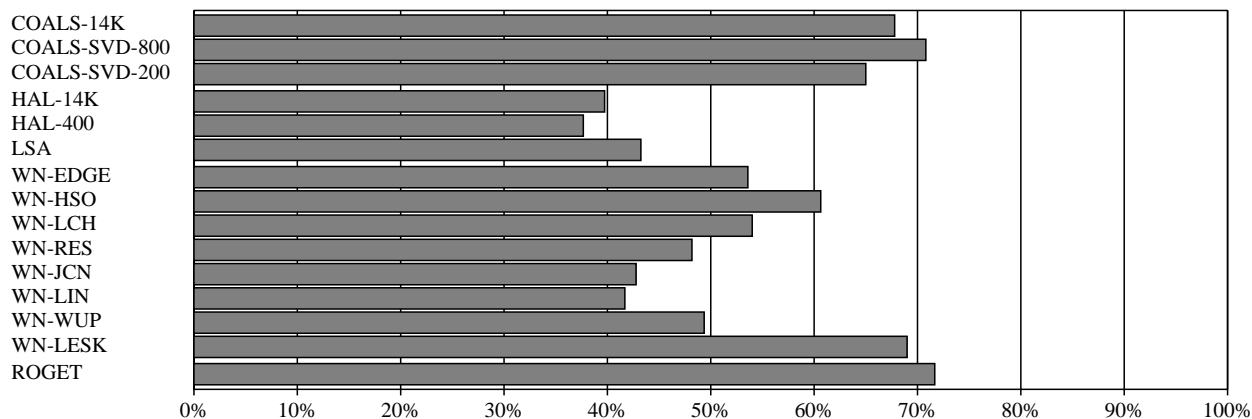


Figure 7: Overall performance of the models on the combined vocabulary tests.

3.3 Empirical Evaluation Summary

The COALS models, especially COALS-SVD-800, though not always the best were consistently good on all of the empirical tests. ROGET also performed very well. It was somewhat worse than COALS on the WS-353, WS-400, and VT-TOEFL tasks, but better on WS-RG, WS-MC, and VT-ESL. The WordNet methods did very well on WS-RG and WS-MC, but poorly on the large similarity tasks. WN-LESK was fairly good at the vocabulary tests, while WN-EDGE and WN-LCH did well on the noun/verb portions of the vocabulary tests.

LSA was comparable to COALS on the similarity rating tasks, although it only scored better than COALS-SVD-800 on WS-MC. However, LSA did rather poorly on the vocabulary tests. This may be because the TOEFL and RDWP tests used low-frequency words, while the ESL test relied on subtle distinctions or particular word senses. LSA may have performed better on these tests were it trained on a larger corpus. Finally, HAL was simply not comparable to the other models, except perhaps on the TOEFL tests.

3.4 Multi-dimensional scaling and clustering

We now turn to some visualization and analysis techniques to get a better understanding of the COALS model and its behavior on particular words and word types. In these experiments, we will make use of the COALS-14K model. Although we may have gotten better results with COALS-SVD-800, COALS-14K is more directly comparable to HAL, which has been evaluated in the past using similar analyses. One method employed by Lund and Burgess (1996), Burgess and Lund (1997a), and Burgess (1998) for visualizing the HAL vectors is multi-dimensional scaling (MDS). This involves projecting the high-dimensional space of word vectors into a much lower

dimensional space, typically having just two dimensions, while preserving the relative distance between vectors as much as possible.

There are many methods for performing MDS. One popular approach is to compute a principal components analysis on the matrix of pairwise vector distances or the SVD on the vector matrix itself. The visualization is usually done by selecting two dimensions from among the top three or four principal components. This gives the experimenter some leeway in deciding which information to present. An alternative approach is to choose a set of words to be examined and to assign each word an initial point in a two dimensional space. The points are then gradually adjusted using gradient descent to minimize some error function relating the new pairwise distances to the original ones. This approach is potentially capable of fitting more of the similarity structure into the available dimensions because no principal component information is discarded outright.

The actual method we employed is a version of ordinal, or non-metric, gradient descent (Shepard, 1962; Kruskal, 1964), which is described more fully in Rohde (2002b). The initial pairwise distances between the word vectors were computed using the following function, which will be referred to as *correlation distance*:

$$D(a,b) = 1 - \sqrt{\min(S(a,b), 0)}$$

where $S(a,b)$ is the correlation between vectors a and b , as defined in Table 3.

In this case, the square root is superfluous because the ordinal gradient descent method only retains the rank order of the pairwise distances. The gradient descent begins by projecting the vectors into two dimensional space using randomized basis vectors. The points are then adjusted to minimize *stress*, which is essentially the root mean squared difference between the actual pairwise Euclidean distances of the points in the new space and the

closest possible set of Euclidean distances that are constrained to share the rank order of the pairwise correlation distances of the original vectors. An automatic learning rate adjustment procedure is used to control the gradient descent. Because this technique does not always find the globally optimum possible arrangement, six trials were performed for each scaling problem and the solution with minimal stress was chosen. Typically, two or three of the trials achieved similar minimal stress values.

Noun types

In this experiment, COALS vectors were obtained for 40 nouns selected from three main classes: animals, body parts, and geographical locations. The MDS of these words is shown in Figure 8. Results for the HAL method using many of these same words are displayed in Figure 2 of Lund and Burgess (1996), Figure 2 of Burgess and Lund (1997a), and Figure 2 of Burgess (1998). Visual comparison with these figures should indicate that the current method obtains a much tighter clustering of these noun categories.

Body parts are all clustered in the upper left and animals in the lower left. It is likely that *mouse* separates from the other animals due to its computer sense, *oyster* because it is primarily a food, and *bull* because of its vulgar sense. Interestingly, the places are also separated into two or three clusters. The countries and continents are in the upper right and the cities and states are below. It appears that the North American cities have been separated from the other places. One could speculate that *Moscow* has grouped closer to the countries because it is a capital city and is therefore often referred to in much the same terms as a country, as in, "...easing the way for *Moscow* to export more to *Western Europe*."

An alternative way to visualize the similarity structure of a set of items is *hierarchical clustering* (Johnson, 1967). This general technique also has many variants. In the version used here, pairwise distances are again computed using the correlation distance function. Initially, each point forms its own cluster. The two closest clusters are merged and their centroid computed to form the new location of the cluster (average-link clustering). This process repeats until only one cluster remains. Figure 9 shows the hierarchical clustering of the 40 nouns. Each vertical line represents a cluster. The length of a horizontal line represents the correlation distance between the centroid of a cluster and the centroid of its parent cluster. Note that, according to the correlation distance function, two points are not necessarily equidistant from their centroid.

The primary division that the clustering algorithm finds is between the places and the other nouns. Within the places, there is a distinction between the cities and the

states, countries and continents (plus *Moscow*). Within the set of body parts there is little structure. *Wrist* and *ankle* are the closest pair, but the other body parts do not have much substructure, as indicated by the series of increasingly longer horizontal lines merging the words onto the main cluster one by one. Within the animals there is a cluster of domestic and farm animals. But these do not group with the other animals. *Turtle* and *oyster* are quite close, perhaps because they are foods.

It is notable that the multidimensional scaling and clustering techniques do not entirely agree. Both involve a considerable reduction, and therefore possible loss, of information. *Turtle* is close to *cow* and *lion* in the MDS plot, but that is not apparent in the clustering. On the other hand, the clustering distinguishes the (non-capital) cities from the other places whereas the MDS plot places *Hawaii* close to *Tokyo*. Although *France* appeared to group with *China* and *Russia* in the MDS plot, it doesn't in the hierarchical clustering. MDS has the potential to map quite different points onto nearby locations and clustering has the potential to display apparent structure where there is none or fail to find structure when it is too complex. Therefore, visualization techniques such as these should not be overly relied upon as indicators of precise pairwise similarity, but they can prove very useful for understanding general patterns in complex, high-dimensional data.

An alternative technique for quantitatively evaluating the quality of a set of predefined clusters is to compare the average distance between points in different clusters to the average distance between points that share a cluster. By computing these distances using correlation distance in the original space, rather than in a reduced dimensionality space, we can avoid the possible biases introduced by the dimensionality reduction. And rather than comparing average pairwise distances, we will actually use the root mean square (r.m.s.) between- and within-cluster distances, although this has little bearing on these results. An effective way to combine these distances into a *cluster score* is to divide their difference by their sum. The resulting ratio will fall in the [-1, 1] range, with random clusterings having an expected score of 0. Because nearest neighbors in the COALS space typically have a correlation distance of about 0.25, the maximum possible cluster score is roughly 0.6.

If we define the four clusters in the noun categorization task to be those indicated by the four different symbols used in Figure 8, the r.m.s. between cluster distance is 0.91, while the r.m.s. within cluster distance is 0.55, resulting in a cluster score of 0.247. The fact that this score is positive indicates that the clusters are in some sense meaningful. However, the absolute magnitude of this score has little meaning and it is best understood relative to other cluster scores. For example, if we instead

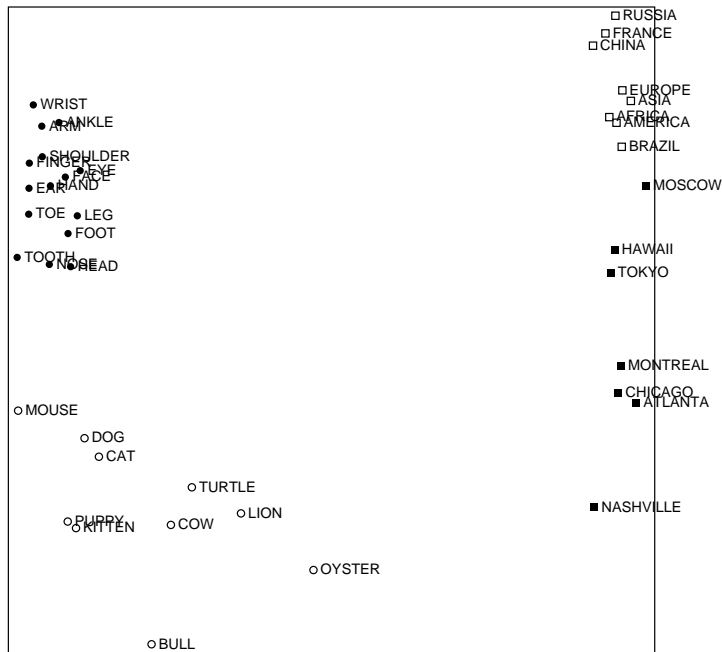


Figure 8: Multidimensional scaling for three noun classes.

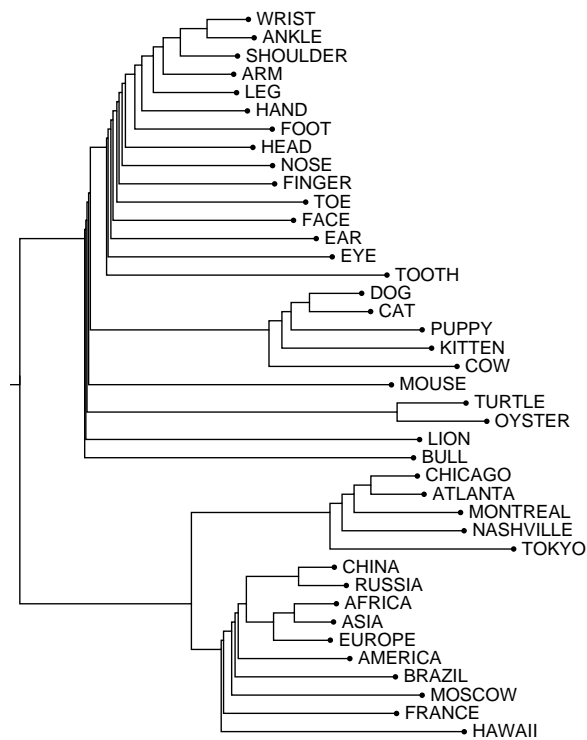


Figure 9: Hierarchical clustering for three noun classes using distances based on vector correlations.

define three clusters, with just one cluster containing all of the places, the cluster score falls to 0.222. The fact that placing small (cities and states) and large (countries and continents) geographical areas in distinct clusters increases the cluster score provides further evidence that the model is finding a substantive distinction between them.

Verb types

The next area we will look at are verb classes. Verbs are, arguably, more complex than nouns in that there are a number of dimensions, both syntactic and semantic, on which they can vary. Verbs can be distinguished by tense and aspect, argument structure, and on other syntactic levels, in addition to semantic differences. But there is often a noticeable correlation between verb meaning and argument structure (Levin, 1993). For example, verbs that involve giving or sending generally take a dative object, while thought verbs (*think*, *believe*, *know*) can usually take a sentential complement. If the model assigns those verbs similar representations, is it due to their syntactic or semantic similarity? Certainly, it will be some combination of the two and one interesting question is the degree to which COALS is sensitive to syntax versus semantics.

We begin by looking at eight common verbs selected from each of three semantic classes: violent action verbs, verbs of sensation, and verbs of communication. As shown in Figure 10, the verbs of each class do group together in the multidimensional scaling, but the clusters are not as tight as for the noun classes. Action verbs are in the lower right of the plot, communication in the lower left and sensation in the upper half. The points are not evenly distributed in the space because *detected* and *cut* are outliers that fall at the extremes of their classes. According to this plot, the model does not seem to be particularly sensitive to argument structure, as there is not a clear distinction between the communication verbs that are frequently ditransitive (*told*, *asked*) and those that usually take a single object (*explained*, *questioned*) or no object (*shouted*). However, the model does seem to be reasonably sensitive to the semantic distinctions between the verbs. The overall cluster score for this set of verbs is 0.078, which is considerably lower than the score on the noun clustering task.

To further investigate the model's bias towards semantic over syntactic distinctions, the next MDS experiment involved eight verb roots in each of four forms: present tense, past tense, progressive (*-ing*), and past participle. As shown in Figure 11, the verbs are tightly clustered on the basis of verb root, rather than on syntactic form. When divided according to root, the cluster score is a very strong 0.375. This suggests that the model may be purely driven by semantics, with no consistent contribution from syntax.

However, an additional experiment, shown in Figure 12, leads to a seemingly contradictory outcome. In this case, we have selected 12 verb roots, including the eight from Figure 11, and performed a multidimensional scaling using just their past tense and past participle forms, which are all unambiguous. Although the clustering is not particularly tight, with a score of 0.078, there is a clear and consistent effect of syntactic form. All of the past tenses align on the bottom and the past participles on top.

The MDS analyses in Figures 11 and 12 would appear to be contradictory. In the first case, verb root dominates, while in the second case verb type appears to be the dominant factor. Further analysis using the cluster score indicates that COALS itself is behaving in a relatively consistent manner. On the first task, shown in Figure 11, the score when clustering on the basis of verb root was 0.375. However, if we instead categorize the verbs on the basis of syntactic form, the clustering score is 0.061. While substantially lower, this score is still highly significant and the mean correlations within and across the clusters are statistically different ($F(1,494)=17.32$, $p < 0.0001$). Therefore, although the model's vectors primarily reflect properties of verb roots, there is also a consistent syntactic, or verb form, component. The same is true for the verb pairs shown in Figure 12. The clustering score computed on the basis of syntactic form was, in this case, 0.078, which is only a bit stronger than the 0.061 on the previous experiment. But if the clustering score is computed on the basis of verb root, it is 0.240. Therefore, in both cases the model's representations of the verbs primarily reflected verb root, while syntactic form plays a noticeable, though weaker, role.

How can two MDS analyses lead to such apparently contradictory results, such as those in Figures 11 and 12? The answer lies in the potentially misleading effect of information loss resulting from MDS. Vectors in a high-dimensional space can have a complex similarity structure. The vector representing some concept might, for example, be composed of features reflecting a variety of properties, such as its size, its shape, its appearance, its use, where it comes from, whether it is dangerous, and so forth. Thus, a stuffed bear and a toy car might be similar because they are both small toys that children play with. And a toy car and a real car are similar at different levels: in appearance and, metaphorically, in use. But a stuffed bear and a real car don't have all that much in common. When a set of vectors that differ on a variety of levels are rearranged in a two-dimensional space, there are simply not enough degrees of freedom to capture the full complexity the similarity structure. The best low-dimensional representation can be obtained by allowing the most consistently relevant components to dominate.

In Figure 11, the semantic distinctions between verb

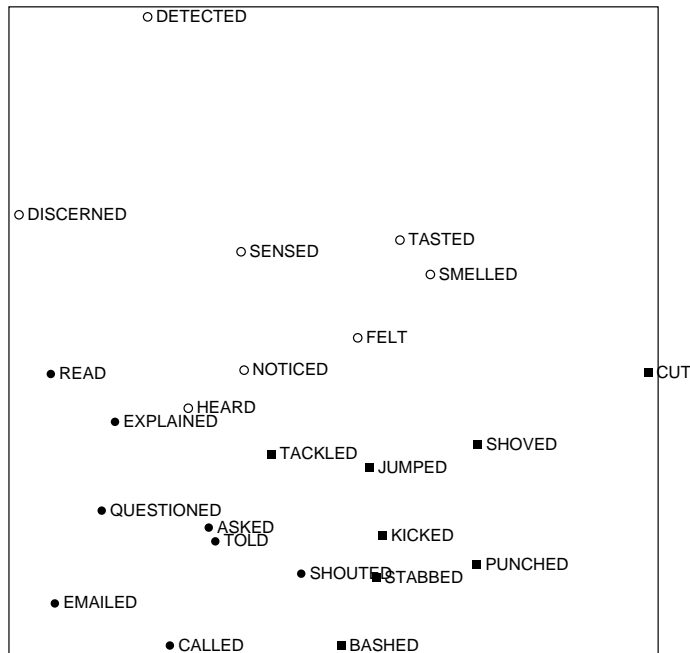


Figure 10: Multidimensional scaling of three verb semantic classes.

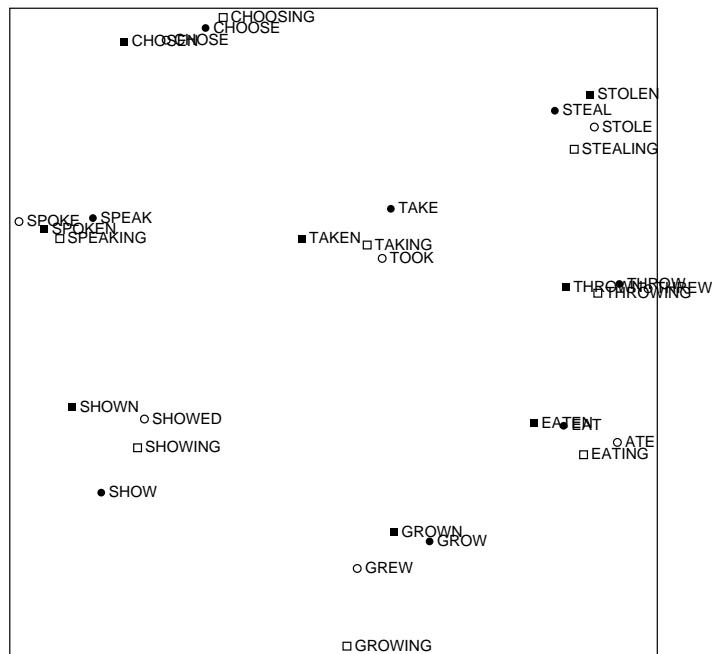


Figure 11: Multidimensional scaling of present, past, progressive, and past participle forms for eight verb families.



Figure 12: Multidimensional scaling of unambiguous past tense and past participle verb forms.

roots dominate and mask the weaker effects of syntactic similarity. With four different verb forms, there would be no way, in this constrained space, to place verb forms from the same root close together and also place verbs with the same syntactic form closer to one another than to the other verbs. On the other hand, in Figure 12, where there is a single syntactic contrast, the consistent effect of this component across all pairs balances the semantic components and draws the past tenses to one side and the past participles to the other. In this case, the stronger semantic similarity structure is not as apparent, though it is not completely lost. Although multidimensional scaling and hierarchical clustering, which suffers from a similar problem, are useful tools for visualization of complex data, they can be misleading and are best used in conjunction with other methods, such as the clustering score.

Noun/verb associates

This final study asks the question, “How much *teach* is in a *teacher*?” That is, how similar are the COALS representations for an agentive noun and its associated action, relative to the similarity between different agentive nouns. Eight word pairs were selected, each including a noun denoting a common human occupation or role and a verb denoting a typical action in which that person might engage. The MDS results for this data set are shown in Figure 13. The nouns occupy the upper right of the space and the verbs the lower left. However, the noun vs. verb clustering is not particularly tight, with a score of 0.065.

It is also apparent that nouns and their associated verbs are correlated in their position along the line separating nouns from verbs, with *driver/drive* in the upper left and *priest/pray* and *bride/marry* (as well as *priest/marry*) in the lower right. If each of the associated nouns and verbs are paired, the clustering score increases to 0.134. The average correlation distance between nouns and their associated verbs is 0.64, compared to a distance of 0.77 between nouns and 0.90 between nouns and non-associated verbs. Therefore, according to the model, semantic domain has a stronger effect than word class, so *teacher* includes somewhat more *teach* than it does *-er*.

3.5 Nearest neighbors

Another way to gain a visceral sense for what the COALS vectors are encoding is to look at the nearest neighbors, in semantic space, of various words. Table 10 shows the 10 nearest neighbors, out of the 100,000 most common words, for some representative nouns. As a point of comparison, similar lists based on HAL can be found for some of these words in Table 2 of Lund and Burgess (1996).

For unambiguous nouns with many near synonyms or subtypes, such as *gun*, the model is quite effective at finding those synonyms. Singular and plural noun forms tend to have very similar representations. For a word like *cardboard*, the model mainly identifies correlates, meaning other types of material. For *Leningrad*, it finds places in Russia, or words associated with it, rather than other cities of the world. But the similarity of these neighbors

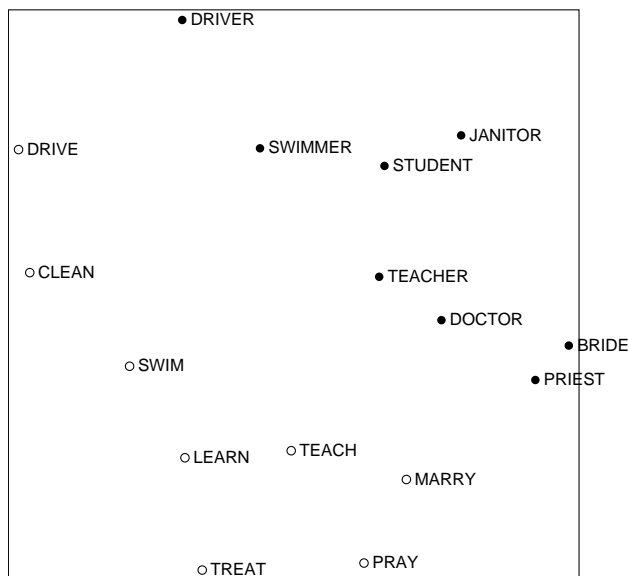


Figure 13: Multidimensional scaling for nouns and their associated verbs.

Table 10

The 10 nearest neighbors and their percent correlation similarities for a set of nouns, under the COALS-14K model.

	gun	point	mind	monopoly	cardboard	lipstick	leningrad	feet
1)	46.4 handgun	32.4 points	33.5 minds	39.9 monopolies	47.4 plastic	42.9 shimmer	24.0 moscow	59.5 inches
2)	41.1 firearms	29.2 argument	24.9 consciousness	27.8 monopolistic	37.2 foam	40.8 eyeliner	22.7 sevastopol	57.7 foot
3)	41.0 firearm	25.4 question	23.2 thoughts	26.5 corporations	36.7 plywood	38.8 clinique	22.7 petersburg	52.0 metres
4)	35.3 handguns	22.3 arguments	22.4 senses	25.0 government	35.6 paper	38.4 mascara	20.7 novosibirsk	45.7 legs
5)	35.0 guns	21.5 idea	22.2 subconscious	23.2 ownership	34.8 corrugated	37.2 revlon	20.3 russia	45.4 centimeters
6)	32.7 pistol	20.1 assertion	20.8 thinking	22.2 property	32.3 boxes	35.4 lipsticks	19.6 oblast	44.4 meters
7)	26.3 weapon	19.5 premise	20.6 perception	22.2 capitalism	31.3 wooden	35.3 gloss	19.5 minsk	40.2 inch
8)	24.4 rifles	19.3 moot	20.4 emotions	21.8 capitalist	31.0 glass	34.1 shimmer	19.2 stalingrad	38.4 shoulders
9)	24.2 shotgun	18.9 distinction	20.1 brain	21.6 authority	30.7 fabric	33.6 blush	19.1 ussr	37.8 knees
10)	23.6 weapons	18.7 statement	19.9 psyche	21.3 subsidies	30.5 aluminum	33.5 nars	19.0 soviet	36.9 toes

Table 11

The 10 nearest neighbors for a set of verbs, according to the COALS-14K model.

	need	buy	play	change	send	understand	explain	create
1)	50.4 want	53.5 buying	63.5 playing	56.9 changing	55.0 sending	56.3 comprehend	53.0 understand	58.2 creating
2)	50.2 needed	52.5 sell	55.5 played	55.3 changes	42.0 email	53.0 explain	46.3 describe	50.6 creates
3)	42.1 needing	49.1 bought	47.6 plays	48.9 changed	40.2 e-mail	49.5 understood	40.0 explaining	45.1 develop
4)	41.2 needs	41.8 purchase	37.2 players	32.2 adjust	39.8 unsubscribe	44.8 realize	39.8 comprehend	43.3 created
5)	41.1 can	40.3 purchased	35.4 player	30.2 affect	37.3 mail	40.9 grasp	39.7 explained	42.6 generate
6)	39.5 able	39.7 selling	33.8 game	29.5 modify	35.7 please	39.1 know	39.0 prove	37.8 build
7)	36.3 try	38.2 sells	32.3 games	28.3 different	33.3 subscribe	38.8 believe	38.2 clarify	36.4 maintain
8)	35.4 should	36.3 buys	29.0 listen	27.1 alter	33.1 receive	38.5 recognize	37.1 argue	36.4 produce
9)	35.3 do	34.0 sale	26.8 playable	25.6 shift	32.7 submit	38.0 misunderstand	37.0 refute	35.4 integrate
10)	34.7 necessary	31.5 cheap	25.0 beat	25.1 altering	31.5 address	37.9 understands	35.9 tell	35.2 implement

Table 12

The 10 nearest neighbors for a set of adjectives, according to the COALS-14K model.

	high	frightened	red	correct	similar	fast	evil	christian
1)	57.5 low	45.6 scared	53.7 blue	59.0 incorrect	44.9 similiar	43.1 faster	24.3 sinful	48.5 catholic
2)	51.9 higher	37.2 terrified	47.8 yellow	37.7 accurate	43.2 different	41.2 slow	23.4 wicked	48.1 protestant
3)	43.4 lower	33.7 confused	45.1 purple	37.5 proper	40.8 same	37.8 slower	23.2 vile	47.9 christians
4)	43.2 highest	33.3 frustrated	44.9 green	36.3 wrong	40.6 such	28.2 rapidly	22.5 demons	47.2 orthodox
5)	35.9 lowest	32.6 worried	43.2 white	34.1 precise	37.7 specific	27.3 quicker	22.3 satan	47.1 religious
6)	31.5 increases	32.4 embarrassed	42.8 black	32.9 exact	35.6 identical	26.8 quick	22.3 god	46.4 christianity
7)	30.7 increase	32.3 angry	36.8 colored	30.7 erroneous	34.6 these	25.9 speeds	22.3 sinister	43.8 fundamentalist
8)	29.2 increasing	31.6 afraid	35.6 orange	30.6 valid	34.4 unusual	25.8 quickly	22.0 immoral	43.5 jewish
9)	28.7 increased	30.4 upset	33.5 grey	30.6 inaccurate	34.1 certain	25.5 speed	21.5 hateful	43.2 evangelical
10)	28.3 lowering	30.3 annoyed	32.4 reddish	29.8 acceptable	32.7 various	24.3 easy	21.3 sadistic	41.2 mormon

Table 13

The 10 nearest neighbors for a set of closed class words, according to the COALS-14K model.

	he	could	where	after	three	although	into	?
1)	81.1 she	75.3 can	29.8 outside	49.6 before	79.0 two	56.2 though	42.9 onto	56.7 <EMO>
2)	51.6 he'd	60.2 would	28.3 somewhere	46.1 during	74.9 four	53.1 however	37.1 back	56.2 ...
3)	49.6 they	58.0 can't	27.9 nearby	41.7 later	71.7 five	52.8 but	36.7 down	53.9 lol
4)	48.5 him	56.6 might	26.7 in	41.0 last	65.6 six	35.6 quite	35.4 inside	49.1 eh
5)	48.0 who	56.5 couldn't	26.7 near	38.3 ago	64.9 eight	34.6 that	34.8 slowly	48.6 huh
6)	47.8 mulder	56.3 able	25.6 what	37.3 ended	60.9 seven	34.6 because	32.9 out	47.3 btw
7)	47.8 she'd	50.7 will	24.5 how	37.0 until	59.6 several	33.9 certainly	31.9 away	47.0 yeah
8)	47.3 someone	50.6 cannot	24.1 anywhere	36.7 afterwards	56.2 couple	32.3 seem	31.7 through	46.9 !
9)	46.7 it	49.1 didn't	23.4 there	36.2 since	54.8 few	31.6 also	31.1 around	45.2 hey
10)	46.6 nobody	48.4 you	23.2 secluded	35.7 started	48.8 nine	31.3 even	30.5 then	43.9 anyway

Table 14

The 3 nearest neighbors for a set of misspelled words, according to the COALS-14K model.

	paly	caribbean	thanx	dont	thier	referring	definatly	wierd
1)	24.5 play	51.7 caribbean	74.9 thanks	78.6 don't	59.1 their	38.5 referring	30.9 definitely	52.0 weird
2)	18.8 playing	29.8 bahamas	49.9 tia (thanks in advance)	57.8 didn't	36.7 own	22.5 talking	20.6 probably	36.7 strange
3)	17.0 played	27.5 mediterranean	47.5 cheers	53.0 wouldn't	35.0 our	22.3 refers	18.6 really	35.1 odd

is relatively weak. The nearest neighbors for *lipstick* are something of a mixed bag, with some correlates (*eyeliner*, *mascara*, *gloss*, *blush*), some companies that manufacture it (*Clinique*, *Revlon*, *NARS*), and some properties associated with it (*shimmery*, *shimmer*).

In the case where one word sense is much more frequent than the others, the weaker senses will be overwhelmed. Consider the words *point* and *mind*. *Point* could mean the sharp end of a pin, or it could mean a proposition. As seen in Table 10, the latter definition clearly dominates in our corpus. Likewise, *mind* could either be a noun or a verb (“Do you mind?”), but the noun sense appears to be dominant. Some words, on the other hand, have two roughly balanced senses, so the vector representing the word will be close to the average of the vectors that would reflect the distinct senses. Is this a serious problem for the model?

Consider generating two random vectors, x and y , in a 1000-dimensional space and averaging them to produce z . Vectors x and y will be uncorrelated, but they each will have a correlation with z of about 70%, which is quite strong, stronger than the typical similarity between a pair of close synonyms. Therefore, if a word has two evenly balanced senses, the vector average of those two meanings won't be nonsense, but will remain highly correlated with each of the two senses, and thus with synonyms of those senses (Kawamoto, 1993; Hinton & Shallice, 1991). The word *feet* is a good illustration of this. Its two main meanings, the things you walk on and a unit of measure, are roughly evenly balanced in frequency and the nearest neighbors to *feet*, according to the model, reflect one or the other of the two senses. Five of the ten neighbors are other units of measure, four are other body parts, and *foot* is ambiguous.

Eight verbs and their nearest neighbors are listed in Table 11. The verb neighbors seem to contain fewer asso-

ciates and more synonyms or near synonyms, although *sale* and *cheap* are neighbors of *buy*, and *different* is a neighbor of *change*. Other forms of the same verb are also generally high on the list. Obviously, the verb *send* is biased in our corpus towards an email or newsgroup context. It isn't clear if *email* is so high on its list because the noun form of *email* is associated with sending or because the verb form is somewhat synonymous. It is interesting that *game* and *games* appear as neighbors of *play*. One might view these as associates; that is, words that frequently appear together but are unrelated in meaning. However, they are not pure associates as they do seem to share some aspect of meaning—we might call it their playfulness. Overall, the verbs are more similar to their nearest neighbors than the nouns are to theirs.

Table 12 displays the nearest neighbors for some adjectives. In this case, the neighbors are mainly a mixture of synonyms, antonyms, and comparative forms (*high*, *higher*, *highest*). The color words are all deemed highly similar to one another. The word *evil* is ambiguous between the adjective and noun forms, and appears to have some associates as neighbors, such as *demons* and *satán*, although all of its neighbors are relatively weak.

Table 13 shows the nearest neighbors for some closed class words. Many of the nearest neighbors for the function words are of the same class as the target word, be it a pronoun, modal, preposition, conjunction, or punctuation. But there do appear to be some associated words on the list. *He* and *she* have a very high correlation similarity. *Who* is probably merely associated with *he* and *Mulder* is hard to explain, except that it and *Scully*, both characters in the popular X-Files television show, happen to have been very frequent in our corpus. The nearest neighbors of *after* are not all prepositions, but they are all temporal words. Interestingly, *three* was found to be similar not only to other numbers but to the quantifiers *several*, *cou-*

ple, and *few*. Presumably *one* is not as similar to *three* as the other digits are because it is less often used as a quantifier and more often to mean *someone*. The nearest neighbors of the question mark tend to be other symbols and interjections that newsgroup writers use to end or bridge their sentences, such as emoticons (<EMO>) and the abbreviations *lol* (laughing out loud) and *btw* (by the way).

It seems evident, on the basis of these nearest neighbor lists, that the COALS method produces quite reasonable responses on the whole. The nearest neighbors to a word are usually strongly semantically related, either synonyms, antonyms, coordinates, subordinates, or superordinates. But there is still some tendency for pure associates, such as *could-you* or *buy-cheap*, to be assigned similar representations. It may be inevitable that a model based on word co-occurrence will reflect pure association to some degree, but this tendency is minimized in COALS by the use of a narrow neighborhood window and correlation normalization.

In one final example, we chose eight common misspellings that were missed by our automatic spelling correction procedure and found their nearest neighbors, shown in Table 14. Although COALS makes no use of orthography and doesn't weight its nearest neighbors by frequency, the semantic nearest neighbor for each of these misspellings is the correct spelling of the word. Therefore, it is likely that COALS could be used in conjunction with some simple orthographic matching procedures to create an effective spelling correction system for common mistakes. It remains to be seen how well it would work on unique or infrequent mistakes for which little co-occurrence data is available.

4 Variations on the COALS method

In this section, we explore the components of the COALS model that distinguish it from other vector-based models and contribute to its success. In particular, we examine alternative matrix normalization methods, the differences between HAL and COALS, the effects of vector dimensionality on the performance of COALS, COALS-SVD, and COALS-SVDB, and the interaction between corpus size and the co-occurrence window size.

4.1 Vector normalization methods

HAL, LSA, and COALS all involve constructing a co-occurrence matrix and normalizing the cells of that matrix to accentuate the useful information it contains. Some possible normalization functions were given in Table 4. HAL normalizes the length of each row vector, and thus the sum of the squared components, to 1. A similar form

of row normalization scales the sum of the components to 1. Neither of these methods has any effect on the correlation or cosine similarity measures, except when averaging vectors to produce representations of phrases on the VT-RDWP task. Row normalizations do nothing to reduce the influence of columns representing high frequency words or large documents. An alternative method that does do this is column normalization, which scales each column to a sum of 1.

The entropy normalization often used with LSA is a modified form of row normalization. It replaces each entry with its log, which reduces but does not eliminate the influence of high frequency columns. Each row is then divided by its entropy, so that rows without much variation within them are discounted. This second step again has no bearing if the row-wise correlation or cosine is to be computed immediately, but it does affect a subsequent SVD.

The normalization method used in COALS initially replaces each cell in the matrix with the temporal correlation of the row and column items. This is relatively insensitive to the overall frequency of both the row and column items and is therefore something like a simultaneous row and column normalization. Negative values are then discarded and the positive values are square rooted to reduce the influence of large correlations. Along with improving the performance, this step has the added benefit of decreasing the density of the resulting matrix. Although they are not very sparse, COALS matrices usually have a density of about 15–20%, roughly half that of matrices produced with the other normalization procedures, making the computation of the SVD more efficient.

Table 15 shows the effects of these various normalization procedures when used in place of square root correlation on the COALS and COALS-SVD models. With the exception of the normalization procedure, the models in the top half of the table are all identical to COALS-14K and those in the bottom half are identical to COALS-SVD-800. Using no normalization (-NONE) and using row normalization (-ROW) results in relatively poor performance. Without the SVD, they are equivalent except that row normalization gives a slight improvement on the multi-word RDWP phrases. Because the correlation similarity function ignores the row magnitude, column normalization (-COL) works nearly as well as correlation on the COALS model. But if used in conjunction with the SVD, column normalization by itself is ineffective. The opposite is true for entropy normalization (-ENTR). It results in no improvement without the SVD, but with the SVD it is quite effective.

The -COR variants are similar to the COALS models in that they compute the correlation and discard the negative values, but they do not take the square root of the positive correlations. There is little difference between this

Table 15

Performance of the COALS and COALS-SVD models using various normalization procedures.

Model	WS-RG	WS-RG-ND	WS-MC	WS-MC-ND	WS-353	WS-400	WS-400-NV	WS-400-ND	WS-400-PT	VT-TOEFL	VT-TOEFL-NV	VT-ESL	VT-ESL-NV	VT-RDWP	VT-RDWP-NV	VT-ALL	VT-ALL-NV	Overall
COALS-NONE	45.9	54.1	50.5	63.6	37.3	42.6	44.2	52.6	82.6	66.2	65.8	32.0	37.5	46.8	48.1	48.7	49.0	50.7
COALS-ROW	45.9	54.1	50.5	63.6	37.3	42.6	44.2	52.6	82.6	66.2	65.8	32.0	37.5	50.8	52.8	51.5	52.4	51.4
COALS-COL	64.6	75.0	63.6	78.2	61.4	64.0	63.2	70.7	92.4	78.8	79.0	52.0	57.5	57.5	58.0	60.6	60.6	67.1
COALS-ENTR	46.1	51.3	48.9	67.1	35.7	38.4	37.7	45.2	77.5	66.2	65.8	40.0	45.0	53.8	51.9	54.3	52.8	50.5
COALS-COR	67.3	76.7	67.8	80.7	59.3	62.8	63.1	71.0	90.8	83.8	81.6	54.0	55.0	61.2	62.2	64.3	63.8	68.3
COALS-14K	68.2	79.1	67.1	85.2	62.6	62.4	60.9	71.2	88.0	86.2	81.6	52.0	52.5	65.5	67.4	67.8	67.2	69.2
COALS-SVD-NONE	60.6	67.1	82.8	85.4	43.1	47.8	48.9	58.3	81.6	75.0	65.8	42.0	42.5	45.8	47.2	51.0	49.0	58.3
COALS-SVD-ROW	60.3	69.1	61.2	75.0	52.3	56.4	57.1	64.5	90.0	70.0	60.5	46.0	50.0	50.8	50.0	53.8	51.4	60.5
COALS-SVD-COL	51.1	58.2	49.8	66.6	48.0	54.0	52.2	64.5	91.4	72.5	68.4	36.0	37.5	45.2	45.3	49.0	47.2	55.9
COALS-SVD-ENTR	67.1	76.6	73.3	83.3	66.1	68.1	67.3	72.6	88.4	86.2	84.2	66.0	65.0	61.5	60.3	66.4	64.1	71.5
COALS-SVD-COR	49.6	55.5	37.1	45.9	47.8	52.1	51.8	64.2	87.5	61.2	60.5	40.0	42.5	47.5	45.8	49.0	47.2	53.2
COALS-SVD-800	67.3	77.9	72.7	82.5	65.7	68.4	67.5	74.2	91.6	88.8	86.8	68.0	67.5	66.8	69.2	70.8	71.3	73.4

and the COALS procedure without the SVD. But with the SVD, taking the square roots of the correlations is much more effective. Both with and without the use of the SVD for dimensionality reduction, the COALS normalization procedure results in the best overall performance on our tasks.

4.2 Differences between the HAL and COALS methods

In this section we take a closer look at the differences between the HAL and COALS procedures and the effects these differences have on performance. Table 16 shows the major performance measures for 13 different models, ranging from the HAL procedure in model A to the COALS procedure in model M.

All of these models use vectors with 14,000 components. The models with Distinct L/R checked make a distinction between neighboring words that occur to the left and right of the target word when counting the co-occurrences. In these models, the 14,000 columns are chosen on the basis of highest variance. The other models do not distinguish between left and right neighbors and select the top columns on the basis of overall frequency. Window Size is the radius of the window around the target word within which neighbors are counted. Window Shape indicates whether the window is ramped, with a higher weighting for closer neighbors, or flat. Closed Class indicates whether closed class columns are included among the 14,000. Vector Norm. is the procedure used to normalize the matrix. *Length* is the vector length normalization used in HAL, while *correl.* is the square root positive correlation used in COALS. Finally, the Distance Func. is

the function used to compute vector similarity. *Euclid.* is the inverse squared Euclidean distance and *correl.* is the correlation distance defined in Table 3.

The typical HAL model (A) has distinct left- and right-neighbor columns, a size 10 window, uses closed class columns, length normalization, and some form of Euclidean distance or similarity. In contrast, COALS does not have distinct left- and right-neighbor columns, uses a size 4 window, ignores closed class columns, and uses correlation normalization and similarity functions. We will explore the importance of each of these differences.

Simply switching to a size 4 window under the HAL method (B) is counter-productive. Not enough information is being separated from the noise in the co-occurrence data for this difference to matter. A major problem with the HAL length normalization procedure is that the vectors are dominated by high-frequency columns. Thus, eliminating the closed class columns (C), which are high frequency but carry relatively little semantic information, results in a significant improvement in the model. Model D is identical to B except that the correlation distance function is used, which results in no change in performance. Simply using correlation similarity without the right normalization is not helpful.

Model E is like B but uses correlation normalization, resulting in a significant improvement. Adding the correlation similarity measure on top of this (F) now leads to further improvement. Ignoring closed class columns as well (H) is better still. In this case, using a size 10 window (G) is now somewhat less effective than the size 4 window. Model I is similar to F but it does not make the distinction between left and right neighbors. This improves the overall performance by several points. Now,

Table 16

Some variations that explore differences between the COALS (A) and HAL (L) models.

Model	Distinct L/R	Window Size	Window Shape	Closed Class	Vector Norm.	Distance Func.	WS-RG	WS-MC	WS-353	WS-400	WS-400-PT	VT-TOEFL	VT-ESL	VT-RDWP	VT-ALL	Overall
A. (HAL)	✓	10	ramp	✓	length	euclid.	14.6	25.6	28.2	13.6	24.8	56.2	26.0	37.9	39.7	27.8
B.	✓	4	ramp	✓	length	euclid.	14.2	18.7	28.6	11.4	17.9	56.2	32.0	36.2	39.2	26.2
C.	✓	4	ramp		length	euclid.	43.7	52.3	34.3	36.6	67.9	66.2	36.0	53.8	54.1	48.9
D.	✓	4	ramp	✓	length	correl.	14.3	19.3	29.2	11.4	14.9	56.2	32.0	35.8	39.0	26.1
E.	✓	4	ramp	✓	correl.	euclid.	63.3	63.6	48.9	50.0	75.8	80.0	46.0	60.8	62.4	61.1
F.	✓	4	ramp	✓	correl.	correl.	66.3	65.7	58.6	54.1	78.7	82.5	52.0	59.5	62.9	64.5
G.	✓	10	ramp		correl.	correl.	65.3	65.1	61.9	58.7	87.8	82.5	44.0	56.8	60.1	65.4
H.	✓	4	ramp		correl.	correl.	68.1	68.0	61.1	58.5	83.8	83.8	54.0	62.5	65.5	67.2
I.		4	ramp	✓	correl.	correl.	67.7	65.7	61.1	60.5	87.2	85.0	50.0	64.2	66.4	68.0
J.		10	flat		correl.	correl.	66.0	62.0	64.1	60.4	86.6	77.5	44.0	58.5	60.4	65.0
K.		10	ramp		correl.	correl.	65.4	63.6	62.6	59.7	87.8	80.0	46.0	58.5	61.0	65.7
L.		4	flat		correl.	correl.	69.2	64.6	66.3	62.5	87.8	83.8	58.0	64.8	67.3	69.4
M. (COALS-14K)		4	ramp		correl.	correl.	68.2	67.1	62.6	62.4	88.0	86.2	52.0	65.5	67.8	69.2

ignoring the closed class columns, to produce model M, results in a small improvement. Therefore, due to its improved normalization, COALS is not hurt significantly by the closed class columns as is the HAL model. But removing them is still somewhat helpful, if only to reduce the influence of syntax on the similarity structure.

Model L is like M, but uses a flat window, in which all eight neighbors are given the same weighting. This actually improves the performance slightly, but perhaps not significantly so. The ramped size 10 window (K) is worse than the size 4 windows. But it is a bit better than the flat size 10 window. We will explore the role of window size further in Section 4.4.

In short, the changes that lead to the improved performance of COALS over HAL are, in order of importance, the normalization of columns through the conversion to word-pair correlations, ignoring negative values and taking the square roots of correlations to reduce the influence of large values, using the correlation similarity measure, using a narrower window, not distinguishing between left and right neighbors, and ignoring closed class columns.

4.3 Dimensionality

All the previous results for COALS used the 14,000 most common open-class columns from the co-occurrence matrix. In Table 17 we explore models with dimensionality ranging from 1,000 to 100,000. Above 14,000, there is very little change in performance. The best overall model actually uses 80,000 columns, but it is only 0.7 points ahead of the COALS-14K model. Even with just 1,000 columns, the overall score is still over 60%. Most of the word similarity tests are performed best by the 30,000 to

60,000 dimension models, although the small models do better on WS-MC and the very large models tend to be better at the vocabulary tests.

Table 18 shows the effect of dimensionality on the real-valued COALS-SVD model. Using the SVD not only reduces the required dimensionality of the models' vectors, but can also improve its performance by several points, due to the helpful generalization SVD can provide on well-structured matrices. In this case, there is a disadvantage to using too many dimensions. The best performance is achieved with 500–1000 dimensions. Even with as few as 200, COALS-SVD can perform as well as the plain COALS model. The 500-dimension model does best on the WS-353 and WS-400 tasks, while the 1000-dimension model does best on the vocabulary tests. Using 600 or 800 dimensions is a nice compromise between them.

Table 19 shows the effect of dimensionality on the binary-valued COALS-SVDB model, which is identical to COALS-SVD except that the vectors have been discretized, with negative components set to 0 and positive to 1. Impressively, the use of binary vectors results in little degradation in performance, usually just 2.5 to 3 percentage points. The 400 to 800 dimension binary models are all roughly equivalent, with performance dropping off with fewer dimensions. But for small modeling projects, binary COALS-SVDB vectors with just 25 or 50 dimensions can be used while still retaining nearly as much of the useful similarity structure as is captured by the WordNet-based approaches.

Table 17

Performance of COALS using real-valued vectors with dimensionality ranging from 1000 to 100,000.

Model	WS-RG	WS-RG-ND	WS-MC	WS-MC-ND	WS-353	WS-400	WS-400-NV	WS-400-ND	WS-400-PT	VT-TOEFL	VT-TOEFL-NV	VT-ESL	VT-ESL-NV	VT-RDWP	VT-RDWP-NV	VT-ALL	VT-ALL-NV	Overall
COALS-500	54.1	60.2	72.1	82.6	51.6	44.7	43.5	52.8	74.0	78.8	73.7	40.0	42.5	50.8	50.5	54.8	52.6	57.1
COALS-1K	60.6	68.9	70.8	86.4	54.6	49.7	47.5	60.9	78.8	81.2	76.3	42.0	47.5	53.8	52.4	57.6	55.0	60.9
COALS-2K	64.2	74.3	67.2	84.1	57.9	54.6	52.8	65.0	82.1	83.8	76.3	42.0	45.0	58.8	59.4	61.5	59.8	63.6
COALS-3K	64.7	74.7	66.0	83.4	59.2	57.0	55.1	66.9	82.7	81.2	76.3	46.0	42.5	57.8	58.5	60.8	58.8	64.1
COALS-4K	66.8	76.5	68.6	83.8	60.9	58.3	56.5	67.7	84.4	82.5	79.0	44.0	42.5	60.5	59.4	62.7	59.8	65.5
COALS-5K	67.5	77.9	68.7	85.4	61.4	59.8	58.0	68.7	85.6	86.2	84.2	46.0	45.0	59.5	58.5	62.9	60.2	66.6
COALS-6K	68.7	79.3	69.4	86.5	62.2	60.2	58.6	69.3	86.1	85.0	84.2	46.0	45.0	60.2	59.4	63.1	60.9	67.1
COALS-8K	68.0	78.8	67.2	84.8	62.3	61.5	60.0	70.5	87.2	86.2	84.2	52.0	50.0	62.2	62.7	65.5	64.0	68.2
COALS-10K	68.3	78.9	67.0	84.3	62.4	61.4	59.8	70.1	87.2	85.0	81.6	50.0	50.0	63.5	63.1	65.9	64.0	68.1
COALS-12K	68.0	78.8	66.1	83.9	62.5	62.0	60.5	70.9	87.6	85.0	81.6	50.0	52.5	63.8	65.0	66.2	65.7	68.4
COALS-14K	68.2	79.1	67.1	85.2	62.6	62.4	60.9	71.2	88.0	86.2	81.6	52.0	52.5	65.5	67.4	67.8	67.2	69.2
COALS-16K	68.1	78.7	66.9	84.9	62.5	62.6	61.2	71.2	88.0	85.0	84.2	50.0	50.0	65.5	66.9	67.3	67.1	68.9
COALS-20K	68.7	79.2	66.5	84.8	62.8	62.2	60.9	71.1	88.0	83.8	84.2	50.0	50.0	66.2	67.8	67.6	67.8	69.0
COALS-30K	68.9	79.5	66.3	84.4	63.1	63.0	62.0	71.7	88.7	85.0	84.2	50.0	50.0	67.8	69.7	68.7	68.8	69.6
COALS-40K	68.5	79.0	65.8	84.0	62.9	63.1	62.3	71.7	89.3	86.2	86.8	48.0	50.0	66.8	68.8	68.0	68.5	69.5
COALS-50K	68.9	78.6	66.1	84.3	62.7	63.3	62.7	71.4	89.5	87.5	86.8	50.0	50.0	67.5	68.8	69.0	68.5	69.8
COALS-60K	69.3	78.8	66.2	84.2	62.6	63.3	62.7	71.3	89.7	87.5	86.8	48.0	50.0	67.5	68.3	68.7	68.1	69.7
COALS-70K	69.3	78.7	66.3	84.0	62.3	63.1	62.5	71.2	89.3	87.5	86.8	50.0	52.5	67.8	68.8	69.2	68.8	69.9
COALS-80K	69.2	78.2	66.4	83.8	62.1	63.0	62.4	71.2	89.3	87.5	86.8	50.0	52.5	68.2	69.7	69.4	69.5	69.9
COALS-90K	68.9	78.1	65.8	83.9	61.8	62.7	62.1	71.0	89.1	86.2	86.8	48.0	50.0	68.5	69.7	69.2	69.1	69.4
COALS-100K	68.8	77.8	65.5	84.0	61.6	62.3	61.7	70.6	88.9	86.2	86.8	48.0	50.0	68.2	69.2	69.0	68.8	69.2

Table 18

Performance of COALS-SVD using real-valued vectors with dimensionality ranging from 50 to 2,000.

Model	WS-RG	WS-RG-ND	WS-MC	WS-MC-ND	WS-353	WS-400	WS-400-NV	WS-400-ND	WS-400-PT	VT-TOEFL	VT-TOEFL-NV	VT-ESL	VT-ESL-NV	VT-RDWP	VT-RDWP-NV	VT-ALL	VT-ALL-NV	Overall
COALS-SVD-25	54.3	60.1	50.7	65.3	55.8	47.1	44.5	59.2	73.6	62.5	65.8	40.0	42.5	47.8	50.0	49.4	50.9	54.4
COALS-SVD-50	57.3	63.9	60.3	72.6	61.8	55.0	52.8	65.8	79.7	76.2	79.0	46.0	45.0	50.2	48.1	54.5	51.9	60.5
COALS-SVD-100	64.5	74.2	68.6	82.4	67.1	59.4	59.5	68.9	82.0	80.0	79.0	52.0	52.5	52.8	50.0	57.6	54.0	65.4
COALS-SVD-150	60.1	71.2	65.9	80.9	66.3	63.3	63.4	72.1	85.7	87.5	84.2	48.0	45.0	57.5	57.5	61.7	59.1	66.8
COALS-SVD-200	64.7	76.6	68.0	82.7	65.3	64.9	65.0	73.8	88.4	86.2	84.2	58.0	55.0	60.8	60.3	65.0	62.6	69.4
COALS-SVD-300	66.4	77.8	69.8	81.9	65.3	66.6	66.9	75.1	88.9	86.2	84.2	62.0	60.0	63.2	64.1	67.1	66.0	71.0
COALS-SVD-400	67.0	78.5	71.9	82.0	66.1	67.5	67.1	76.2	90.4	87.5	86.8	62.0	60.0	66.2	66.4	69.4	68.1	72.2
COALS-SVD-500	66.2	78.0	71.4	83.1	67.1	69.3	68.2	76.4	92.0	86.2	84.2	62.0	62.5	67.2	67.8	69.9	69.1	72.8
COALS-SVD-600	64.9	77.5	73.0	84.7	65.7	68.9	67.8	76.0	91.6	88.8	86.8	64.0	60.0	66.8	68.3	70.3	69.5	72.8
COALS-SVD-800	67.3	77.9	72.7	82.5	65.7	68.4	67.5	74.2	91.6	88.8	86.8	68.0	67.5	66.8	69.2	70.8	71.3	73.4
COALS-SVD-1000	68.0	77.3	67.7	79.3	62.4	67.8	67.6	74.0	91.3	90.0	86.8	72.0	72.5	66.8	70.2	71.5	72.6	73.0
COALS-SVD-1200	68.3	76.8	67.0	79.1	60.3	66.8	66.1	73.1	90.0	90.0	86.8	66.0	65.0	66.5	68.8	70.6	70.5	71.6
COALS-SVD-1600	68.4	75.9	67.2	76.9	58.1	65.9	64.4	71.6	89.9	88.8	84.2	68.0	65.0	66.8	68.3	70.8	69.8	70.9
COALS-SVD-2000	67.8	76.3	65.8	75.9	54.2	65.6	64.6	71.2	89.0	90.0	86.8	68.0	67.5	66.8	68.8	71.0	70.9	70.5

Table 19

Performance of COALS-SVDB using binary vectors with dimensionality ranging from 50 to 2,000.

Model	WS-RG	WS-RG-ND	WS-MC	WS-MC-ND	WS-353	WS-400	WS-400-NV	WS-400-ND	WS-400-PT	VT-TOEFL	VT-TOEFL-NV	VT-ESL	VT-ESL-NV	VT-RDWP	VT-RDWP-NV	VT-ALL	VT-ALL-NV	Overall
COALS-SVDB-25	62.4	67.0	50.9	63.7	48.1	42.2	39.6	52.6	71.0	58.8	56.6	28.0	30.0	40.8	40.3	42.4	40.8	49.9
COALS-SVDB-50	57.9	63.9	63.2	72.9	53.2	49.4	46.4	60.0	78.3	60.0	57.9	38.0	40.0	46.7	46.5	48.1	47.2	55.5
COALS-SVDB-100	63.9	70.1	68.2	79.1	61.6	56.4	55.5	65.0	82.0	78.1	72.4	38.0	40.0	52.3	50.7	55.5	52.2	62.0
COALS-SVDB-150	59.8	68.5	67.5	78.7	61.0	58.9	59.0	69.0	85.3	85.0	84.2	50.0	50.0	54.0	53.0	59.3	56.9	64.8
COALS-SVDB-200	62.1	72.0	66.6	79.7	59.7	60.4	59.8	69.7	87.0	85.0	84.2	54.0	55.0	56.7	55.6	61.4	59.1	66.1
COALS-SVDB-300	63.7	75.0	68.4	78.8	61.0	63.3	63.6	73.0	88.1	86.2	86.8	58.0	60.0	61.8	61.3	65.7	64.3	68.8
COALS-SVDB-400	66.6	78.2	75.1	84.2	60.9	64.4	64.4	73.2	89.4	88.8	89.5	60.0	60.0	62.3	63.9	66.7	66.5	70.8
COALS-SVDB-500	66.7	76.8	74.6	82.3	61.4	65.2	64.8	73.5	89.6	91.2	92.1	56.0	52.5	64.2	66.0	68.0	67.4	70.8
COALS-SVDB-600	64.9	76.4	74.9	82.6	60.0	64.6	65.1	73.3	87.8	88.8	86.8	62.0	57.5	64.2	66.4	68.3	67.8	70.5
COALS-SVDB-800	67.2	77.0	73.9	80.2	59.3	64.4	64.6	71.5	87.2	86.2	86.8	70.0	67.5	63.2	64.1	68.0	67.4	70.8
COALS-SVDB-1000	66.6	76.6	68.5	79.7	57.7	63.9	64.2	70.5	86.0	86.2	89.5	62.0	62.5	63.2	64.5	67.3	67.8	69.3
COALS-SVDB-1200	68.2	76.5	71.9	80.6	54.8	63.4	64.0	69.5	86.7	86.2	89.5	70.0	70.0	65.2	66.4	69.7	70.2	70.4
COALS-SVDB-1600	65.0	72.8	71.9	77.7	51.9	61.0	61.1	66.2	84.6	87.5	89.5	68.0	65.0	64.5	65.5	69.2	68.8	68.5
COALS-SVDB-2000	65.8	73.0	71.1	77.6	47.4	59.3	59.0	64.7	83.0	88.8	92.1	68.0	70.0	63.5	64.5	68.5	68.8	67.7

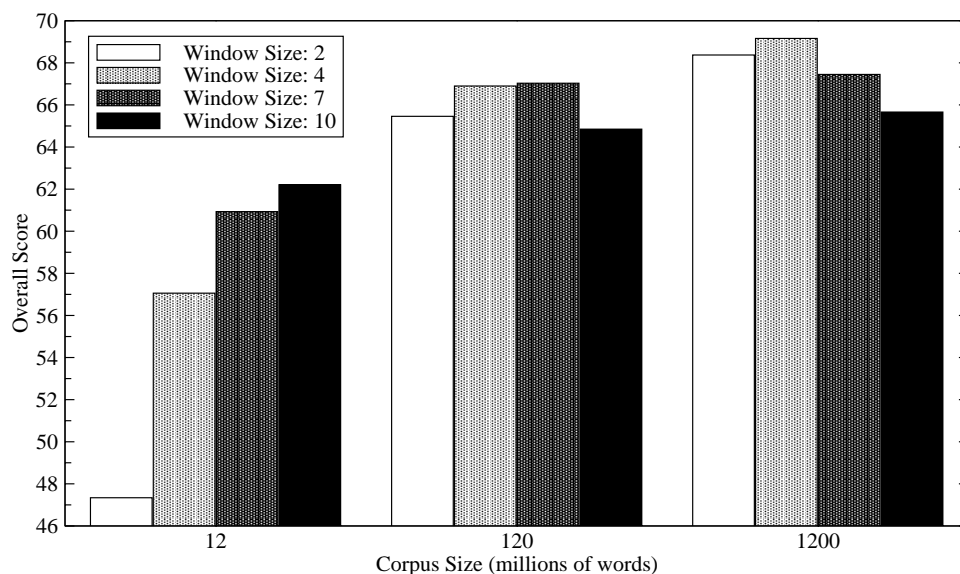


Figure 14: Effects of corpus size and ramped window size on the overall weighted score of the COALS-14K model.

4.4 Corpus and window size

HAL has typically been trained on smaller corpora but with larger windows than we have used here. In this section we examine the interaction between corpus size and window size on the performance of COALS-14K. Figure 14 shows the overall performance of the model with ramped windows ranging from radius 2 to radius 10. The models were trained on either our full Usenet corpus or on a random sampling of 10% or 1% of the articles, which have approximately 120 million and 12 million words, respectively.

Larger windows capture more meaningful co-occurrences, but they also capture more noise. On the smallest corpus, the best performance is achieved by the size 10 window because it is able to collect more of the precious data. However, with the 120 million-word corpus, the signal-to-noise ratio of the largest window declines and the size 7 and size 4 windows perform better. On the full corpus, the size 4 window is the best, followed by the size 2 window and the largest window is actually the worst. Therefore, there is a clear tradeoff to consider in choosing an appropriate window for use with the HAL or COALS methods. Smaller corpora demand larger windows and larger corpora demand smaller ones.

5 Discussion

We have introduced a new method for deriving word meanings, in the form of high-dimensional vectors, from large, unlabeled text corpora. Our method, COALS, and especially its lower-dimensional variant, COALS-SVD, performed well in comparison to 11 other models on a variety of empirical tests. It was particularly effective in predicting human similarity judgments on our WS-400 task, which included a greater diversity of word-pair types than did previous experiments.

The only other model to perform as well as COALS was that based on Roget's Thesaurus (Jarmasz & Szpakowicz, 2003). However, ROGET, like the WordNet models, suffers from several drawbacks. Principal among these is the fact that these models rely on structured lexicons resulting from years of expert human labor, and cannot therefore easily be extended to larger vocabularies or other languages. The HAL, LSA, and COALS models, on the other hand, can be applied to any language for which sufficient text is available. Also, the WordNet and Roget models do not produce semantic representations that can easily be used in a cognitive model (although see Powell, Zajicek, and Duce (2000) for one attempt at this). A principal advantage of the WordNet and Roget's models is that they distinguish word senses. More work needs to be done in automatic word-sense clustering and disambiguation to allow the vector-based methods to do this.

Our model originated as a replication of HAL (Lund & Burgess, 1996), and it continues to be similar to HAL both in its applicability and in the basic premise that human knowledge of lexical semantics can, to a certain extent, be represented as high-dimensional vectors learned through exposure to language alone. Although COALS-SVD achieves better results using fewer dimensions, we have reported several analyses of the plain COALS model because of its analogy to HAL. As we have shown, the improved normalization function used in our method, along with the other less significant changes, results in a much more effective methodology.

In our tests, LSA also performed much better than HAL, although not quite as well as COALS, particularly on the vocabulary tests. COALS-SVD and LSA both make use of the singular value decomposition. One difference between the two models is their normalization procedures, but a more fundamental difference is the form of the co-occurrence matrix they construct. LSA assumes that the input is a discrete set of (usually small) documents. COALS, on the other hand, like HAL, makes use of an undifferentiated text corpus, using a moving window to define the collocation of words. As a result, the models will scale up differently with more data. The number of columns in the LSA matrix is proportional to the number of documents. Although the SVD can be computed on matrices with millions of rows and columns, that is only possible with special machinery and with very sparse matrices. Therefore, it may not be possible to run LSA on very large document collections without discarding all but a small excerpt from each document.

On the other hand, the size of the matrix used in COALS-SVD is essentially fixed. In this work, we used a matrix with 15,000 rows and 14,000 columns. If the size of the training corpus were increased by several orders of magnitude, the size of this matrix need not change. Its density would increase and a slightly larger matrix might be preferable with additional data, but it should still be possible to implement the COALS-SVD method on a single personal computer. Therefore, COALS-SVD should scale up more easily than LSA to corpora much larger than even our 1.2 billion-word text.

Real-valued and binary COALS-SVD vectors can be accessed on the web at:

<http://tedlab.mit.edu/~dr/COALS/>

References

- Banerjee, S., & Pedersen, T. (2002). An adapted lesk algorithm for word sense disambiguation using WordNet. In *Proceedings of the third international conference on intelligent text processing and computational linguistics*. Mexico City.
- Berry, M., Do, T., O'Brien, G., Krishna, V., & Varadhan, S.

- (1996). *Svdpackc (version 1.0) user's guide* (Tech. Rep. No. Tech. Report CS-93-194). Univ. of Tennessee.
- Budanitsky, A., & Hirst, G. (2001). Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Workshop on wordnet and other lexical resources, in the north american chapter of the association for computational linguistics*. Pittsburgh, PA: NAACL-2000.
- Burgess, C. (1998). From simple associations to the building block of language: Modeling meaning in memory with the HAL model. *Behavior Research Methods, Instruments, and Computers*, 30, 188-198.
- Burgess, C., & Lund, K. (1997a). Modelling parsing constraints with high-dimensional context space. *Language and Cognitive Processes*, 12(2/3), 177-210.
- Burgess, C., & Lund, K. (1997b). Representing abstract words and emotional connotation in a high-dimensional memory space. In *Proceedings of the 19th annual conference of the Cognitive Science Society* (pp. 61-66). Mahwah, NJ: Lawrence Erlbaum Associates.
- Deerwester, S. C., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41, 391-407.
- Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., & Ruppin, E. (2002). Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1), 116-131.
- Foltz, P. W., Kintsch, W., & Landauer, T. K. (1998). The measurement of textual coherence with Latent Semantic Analysis. *Discourse Processes*, 25, 285-307.
- Hinton, G. E., & Shallice, T. (1991). Lesioning an attractor network: Investigations of acquired dyslexia. *Psychological Review*, 98, 74-95.
- Hirst, G., & St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. In C. Fellbaum (Ed.), *Wordnet: An electronic lexical database* (pp. 305-332). Cambridge, MA: The MIT Press.
- Jarmasz, M., & Szpakowicz, S. (2003). Roget's Thesaurus and semantic similarity. In *Proceedings of the conference on recent advances in natural language processing (ranlp 2003)* (pp. 212-219). Borovets, Bulgaria.
- Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th international conference on research in computational linguistics (rocling x)* (pp. 19-33). Taiwan: Academica Sinica.
- Johnson, S. (1967). Hierarchical clustering schemes. *Psychometrika*, 2, 241-254.
- Kawamoto, A. H. (1993). Nonlinear dynamics in the resolution of lexical ambiguity: A parallel distributed processing account. *Journal of Memory and Language*, 32, 474-516.
- Kintsch, W. (2000). Metaphor comprehension: A computational theory. *Psychonomic Bulletin and Review*, 7, 257-266.
- Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29, 1-27.
- Kucera, H., & Francis, W. N. (1967). *Computational analysis of present-day American English*. Providence, RI: Brown University Press.
- Laham, D. (1971). Latent semantic analysis approaches to categorization. In *Proceedings of the 19th annual conference of the Cognitive Science Society* (p. 979). Mahwah, NJ: Lawrence Erlbaum Associates.
- 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.
- Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to Latent Semantic Analysis. *Discourse Processes*, 25, 259-284.
- Leacock, C., & Chodorow, M. (1998). Combining local context and wordnet similarity for word sense identification. In C. Fellbaum (Ed.), *Wordnet: An electronic lexical database* (pp. 265-283). Cambridge, MA: The MIT Press.
- Lesk, M. (1986). Automatic sense disambiguation using machine-readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of sigdoc '86* (pp. 24-26). Toronto.
- Levin, B. (1993). *English verb classes and alternations: A preliminary investigation*. Chicago, IL: The University of Chicago Press.
- Lin, D. (1997). Using syntactic dependency as a local context to resolve word sense ambiguity. In *Proceedings of the 35th annual meeting of the association for computational linguistics* (pp. 64-71). Madrid.
- Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, and Computers*, 28, 203-208.
- Lund, K., Burgess, C., & Atchley, R. A. (1995). Semantic and associative priming in high-dimensional semantic space. In *Proceedings of the 17th annual conference of the Cognitive Science Society* (pp. 660-665). Mahwah, NJ: Lawrence Erlbaum Associates.
- Lund, K., Burgess, C., & Audet, C. (1996). Dissociating semantic and associative word relationships using high-dimensional semantic space. In *Proceedings of the 18th annual conference of the Cognitive Science Society* (pp. 603-608). Mahwah, NJ: Lawrence Erlbaum Associates.
- McDonald, S., & Lowe, W. (1998). Modelling functional priming and the associative boost. In *Proceedings of the 20th annual conference of the Cognitive Science Society* (pp. 675-680). Mahwah, NJ: Lawrence Erlbaum Associates.
- Miller, G., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1990). Introduction to wordnet: an online lexical database. *International Journal of Lexicography*, 3(4), 235-244.

- Miller, G. A., & Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1), 1–28.
- Patwardhan, S., Banerjee, S., & Pedersen, T. (2003). Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the fourth international conference on intelligent text processing and computational linguistics, february 17–21*. Mexico City.
- Plaut, D. C. (1997). Structure and function in the lexical system: Insights from distributed models of word reading and lexical decision. *Language and Cognitive Processes*, 12, 767–808.
- Plaut, D. C., & Booth, J. R. (2000). Individual and developmental differences in semantic priming: Empirical and computational support for a single-mechanism account of lexical processing. *Psychological Review*, 107, 786–823.
- Plaut, D. C., Seidenberg, M. S., McClelland, J. L., & Patterson, K. (1996). Understanding normal and impaired word reading: computational principles in quasi-regular domains. *Psychological Review*, 103, 56–115.
- Powell, C., Zajicek, M., & Duce, D. (2000). The generation of word meanings from dictionaries. In *Icslp 2000, vol. iii* (pp. 482–485). Beijing.
- Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19(1), 17–30.
- Ramscar, M. (2001). The influence of semantics on past-tense inflection. In *Proceedings of the 23rd annual conference of the Cognitive Science Society* (pp. 809–814). Mahwah, NJ: Lawrence Erlbaum Associates.
- Resnick, P. (1995). Using information content to evaluate semantic similarity. In *Proceedings of the 14th international joint conference on artificial intelligence* (pp. 448–453). Montreal.
- Rohde, D. L. T. (2002a). *A connectionist model of sentence comprehension and production*. Unpublished doctoral dissertation, Carnegie Mellon University, Department of Computer Science, Pittsburgh, PA.
- Rohde, D. L. T. (2002b). Methods for binary multidimensional scaling. *Neural Computation*, 14, 1195–1232.
- Rubenstein, H., & Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, 8(10), 627–633.
- Shepard, R. N. (1962). The analysis of proximities: Multidimensional scaling with an unknown distance function. *Psychometrika*, 27, 125–139, 219–246.
- Turney, P. D. (2001). Mining the web for synonyms: PMI-IR vs LSA on TOEFL. In *Proceedings of the twelfth european conference on machine learning (ecml-2001)* (pp. 491–502). Freiburg, Germany.
- Wu, Z., & Palmer, M. (1994). Verb semantics and lexical selection. In *Proceedings of the 32nd annual meeting of the association for computational linguistics*. Las Cruces, Mew Mexico.