Organizing the space and behavior of semantic models

Timothy N. Rubin (timrubin@indiana.edu)
Jon A. Willits (jwillits@indiana.edu)
Brent Kievit-Kylar (bkievitk@indiana.edu)
Michael N. Jones (jonesmn@indiana.edu)

Department of Brain and Psychological Sciences, 1101 E. 10th Street
Bloomington, IN 47405

Abstract

Semantic models play an important role in cognitive science. These models use statistical learning to model word meanings from co-occurrences in text corpora. A wide variety of semantic models have been proposed, and the literature has typically emphasized situations in which one model outperforms another. However, because these models often vary with respect to multiple sub-processes (e.g., their normalization or dimensionality-reduction methods), it can be difficult to delineate which of these processes are responsible for observed performance differences. Furthermore, the fact that any two models may vary along multiple dimensions makes it difficult to understand where these models fall within the space of possible psychological theories. In this paper, we propose a general framework for organizing the space of semantic models. We then illustrate how this framework can be used to understand model comparisons in terms of individual manipulations along sub-processes. Using several artificial datasets we show how both representational structure and dimensionality-reduction influence a model's ability to pick up on different types of word relationships.

Keywords: Semantic Modeling. Language Models. Computational Models. Model Comparison.

1. Introduction

Consider the words *robin*, *sparrow* and *wings*. It is clear to any reader that there exists a semantic relationship among all three of these words. However, the *types* of relationships between the pairs are different; a *robin* is a similar animal to a *sparrow*, whereas *wings* are a feature of both a *sparrow* and a *robin*. In many instances, the usage of the word *robin* is indistinguishable from the usage of *sparrow*; that is, the two words could be exchanged and no one would be the wiser. However, replacing either word with *wings* would typically produce an incoherent sentence. One might be able to replace the word *wings* with *arms* while retaining the basic meaning of a sentence, but this would feel like an incorrect usage of the word. This example illustrates the range of ways in which words can be semantically related: two words might be largely substitutable for one another (e.g., *sparrow* and *robin*), two words might be associated with one another (e.g., *sparrow* and *wings*), and two words might belong to the same class of words while not being highly substitutable (e.g., *wings* and *arms*). A central aim of computational models of semantics is to learn about these types of word relationships using linguistic data as input (Jones, Kintsch, & Mewhort, 2006).

A variety of semantic models have been proposed in the psychological literature (see McRae & Jones, 2013 for a review). The relative ability of different models to capture human behavior is evaluated using tasks such as synonym tests (Landauer & Dumais, 1997), predicting human-generated word-associations (Griffiths, Steyvers, & Tenenbaum, 2007) or semantic priming (Jones et al., 2006). The high variability in both the types of semantic models and tasks on which they have been evaluated makes it difficult to compare results across publications. In particular, any two semantic models typically vary with respect to several sub-processes (such as the type of structure in which they encode data, or the type of dimensionality-reduction method they employ). This makes it difficult to identify which modeling choices are responsible for the observed differences in model behavior. Furthermore, comparisons are often made on tasks capturing only a subset of the possible types of word relationships, making it difficult to know in what aspects one model outperforms another.

The goal of the current paper is two-fold. First, we present a framework to organize the space of computational models of semantics. This framework is useful for understanding the various dimensions along which semantic models differ. Furthermore, by identifying existing models—such as LSA (Landauer & Dumais, 1997) or HAL (Lund & Burgess, 1996)—within this framework, it provides a clearer picture of nature of the relationships between these models. Second, we illustrate the usefulness of such a framework for understanding how different modeling choices influence a models' ability to pick up on different aspects of word similarity. To this end, we present experimental results on a number of artificial datasets, using a set of models that vary along two different dimensions within our framework. These results illustrate how both a model's representational structure and use of dimensionality-reduction interacts with the ability to pick up on different aspects of word-similarity.

Components of Semantic Models (Semantic Modeling pipeline): Most semantic models largely consist of the same basic components/sub-processes, where several choices exist for each of these steps. To understand the relationship between different semantic models, it is important to first explicitly define what each of these components is. The relationships between any two models can then be well described by the individual choices they employ for each modeling component. To give an overview of our framework, we first summarize the basic steps in constructing a semantic model from a tokenized corpus.

Model Components and Sub-processes:
1. Encoding Region: The "window" over which text is encoded within a representational structure.
2. Representational Structure: The form of the matrix in which words within encoding regions are stored.

3. Representational Transformation: Matrix normalization and dimensionality-reduction method
4. Similarity Metric / Decision Process: Process by which information is retrieved from the semantic structure

2. Organizing the space of semantic models

In this section, we focus in detail on the steps highlighted above. After describing each step we discuss the space of modeling options that are available. We then locate a number of previously described models within this overall framework.

2.1. Encoding Region

The encoding region defines the span of the individual observations of text that are encoded by a model. In the context of corpus-based semantic models, the encoding region corresponds to the "window" over which a sample of text is encoded into the matrix structure. For example, many semantic models employ a sliding window of 10 words, wherein the co-occurrence of words within a 10-word window is encoded for all unique 10-word windows in a corpus.

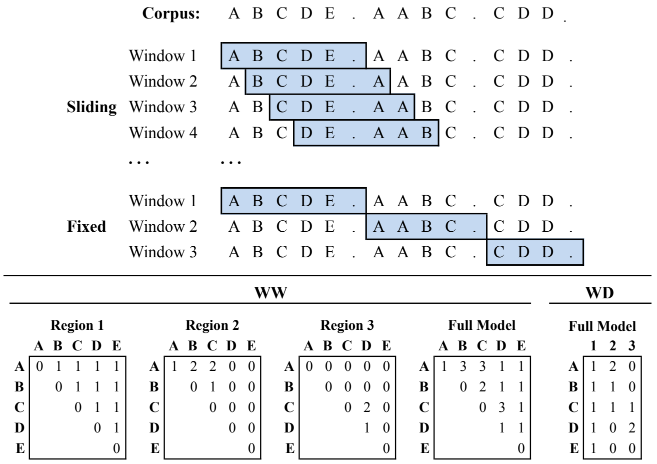
Defining a model's encoding region consists of two distinct options, which correspond to different theoretical stances regarding the process by which semantic knowledge is encoded. The first option determines whether regions employ a "fixed" or "sliding" window. A fixed window method utilizes a set of rules that govern region boundaries within a text. Typically, these regions are defined such that they capture semantically coherent regions of text, such as sentences, paragraphs, or documents. In contrast, a sliding window method utilizes all possible N-length sequences of word-tokens as encoding regions. A model employing a fixed window therefore posits that co-occurrences are tracked primarily within linguistic boundaries (such as sentences), rather than over arbitrary distances within a text.

The second option when defining encoding regions corresponds to the *window-size*. For sliding windows, the region size can be any positive integer between 2 (which encodes the minimal possible information—the co-occurrence of a single pair of words) and the length of the longest document in the corpus. For a fixed window method, a small, medium and large window size might correspond to sentences, paragraphs, and documents. A sliding-window method will have always have the number of encoding regions equal to the number of tokens in a corpus, whereas for a fixed-window method the window-size will be inversely proportional to the total number of encoding regions.

At the top of Figure 1 we illustrate the differences between fixed and sliding windows using a toy corpus consisting of three sentences. For the illustration, we use a 6 token sliding window (including punctuation), and a fixed-window method that utilizes sentences as boundaries around regions of interest.

2.2 Representational Structure

Once the encoding regions have been defined, these regions are mapped to a representational structure for storing the data in a corpus. The two choices for this representational structure we will refer to as Word-by-Word (WW) and Word-by-Document (WD) representations. Within both structures, each unique word is represented via a row in a matrix. In a WW representation, each column also corresponds to a unique word in the corpus. In a WD representation, each column corresponds to a unique encoding region. To be consistent with the psychological and NLP literature, we refer to this as a WD structure, despite the fact that the columns could correspond to any type of encoding region (e.g. sentences, paragraphs, or all regions defined by a sliding window method).


Corpus: A B C D E . A A B C . C D D .
Sliding
Window 1 A B C D E . A A B C . C D D .
Window 2 A B C D E . A A B C . C D D .
Window 3 A B C D E . A A B C . C D D .
Window 4 A B C D E . A A B C . C D D .
… …
Fixed
Window 1 A B C D E . A A B C . C D D .
Window 2 A B C D E . A A B C . C D D .
Window 3 A B C D E . A A B C . C D D .
WW
WD

Region 1

	A	B	C	D	E
A	0	1	1	1	1
B		0	1	1	1
C			0	1	1
D				0	1
E					0

Region 2

	A	B	C	D	E
A	1	2	2	0	0
B		0	1	0	0
C			0	0	0
D				0	0
E					0

Region 3

	A	B	C	D	E
A	0	0	0	0	0
B		0	0	0	0
C			0	2	0
D				1	0
E					0

Full Model

	A	B	C	D	E
A	1	3	3	1	1
B		0	2	1	1
C			0	3	1
D				1	1
E					0

Full Model

	1	2	3
A	1	2	0
B	1	1	0
C	1	1	1
D	1	0	2
E	1	0	0

Figure 1 **Top:** Comparison of encoding regions for a simple corpus using a "sliding" vs. a "fixed" window. **Bottom**: Using the "fixed" encoding regions defined above, a comparison of the WW vs. WD representational structures.

In a WW structure, a word's presence within an encoding region is encoded entirely via its co-occurrence with other words. For all pairs of words within a region, w_1 and w_2, a count is added to the WW matrix at element (w_1,w_2) and (w_2,w_1). This is illustrated at the bottom of Figure 1 using the fixed-window encoding regions defined above. Since the WW matrix is symmetric, we only show the upper-triangular region of the matrix for clarity.[1] As shown in Figure 1, each of the three encoding regions can be mapped to a unique WW matrix of a fixed size. Summing across all of the regions then creates single WW representation of the full corpus. In a WD structure, each encoding region is mapped to a unique column. Each word's frequency within the region is encoded within the row for of that column. For example, the word "D" occurs twice in the third encoding region, so the row corresponding to "D" is assigned a value of two in the third column of the WD matrix.

A key theoretical distinction between the WW and WD representational structures is that, in a WW structure, words are represented strictly by their co-occurrence with other words, making WW structures akin to other psychological theories that stress the associations between individual items, stimuli, or responses. In contrast, WD structures posit

[1] Due to size restrictions we limit the discussion to models that do not account for word-order, although the current framework can be generalized to account for such models as well.

an association between an item and its encoding region (such as a sentence or document). A second, more practical difference between WW and WD representational structures is that since WW "contexts" are fixed across the corpus, this allows the row-representations to be collapsed across all encoding regions. In contrast, the row-representations in a WD encoding structure cannot be collapsed, resulting in a representation who's size scales as a function of the number of encoding regions in the corpus.

2.3. Representational Transformation

The representational transformation corresponds to how the information encoded within the WW or WD matrix is manipulated after it has been stored. This process can (optionally) involve a number of difference procedures, including normalization, abstraction, and dimensionality-reduction. In the semantic modeling literature, a variety of these methods have been proposed. For example, LSA (Landauer and Dumais, 1997) employs log-entropy normalization followed by Singular-Value-Decomposition (SVD) for abstraction and dimensionality reduction. Since the number of possible transformations is potentially infinite, we identify a number of published models with respect to their encoding regions, representational structures and representational transformations, comprising the space we have described thus far in the paper (Table 1).

2.4. Similarity Metric / Decision Process

To compute a similarity between two words w_1 and w_2, semantic models apply some function to the transformed representational structure. Typically this consists of a vector operation across the row representations for each word. For example, in LSA the cosine similarity is computed between words' singular vectors across a subset of dimensions. Although there are alternative approaches that could be employed here (e.g., within a WW matrix a model could directly utilize the value of the matrix element $WW_{w1,w2}$ as a measure of the semantic relationship between words w_1 and w_2), such alternatives have rarely been used in the literature.

2.5. Final Considerations

While it is useful (and computationally equivalent) to define the steps in our framework independently, it is not necessary that a model perform them in a strictly sequential fashion. For the purposes of psychological theories, it is valid to posit that two (or more) steps actually occur in parallel. For example, it may be more psychologically plausible for a model such as LSA to perform dimensionality-reduction *during* the encoding process, such that it does not asymptotically require infinite storage (as more and more regions are encoded).

3. Experiments using artificial datasets

Due to the fact that semantic modeling entails choices along a number of dimensions, it is difficult to know which of these dimensions is responsible for the differences observed when comparing any pair of semantic models. For example, HAL and LSA employ different encoding regions (*sliding windows* over small regions vs. *fixed windows* over large regions), different representational structures (WW vs. WD), different normalization (conditional probability vs. log entropy) and different dimensionality-reduction methods (no abstraction vs. SVD). In this section we illustrate that by isolating individual modeling components, we can identify precisely how the components influence a model's ability to capture different types of word relationships. We employ artificially constructed datasets designed to capture different types of inter-word relationships, while minimizing the number of confounding variables between models.

We designed datasets that captured three distinct types of word relationships, while also limiting the number of possible variables that can contribute to observed differences in model performance. In particular, all datasets were constructed such that they consisted of sets of documents, each of which contained only a single word-pair. By limiting each document to a single word-pair, we eliminated any potential effects caused by the definition of encoding-region; for a 2-word document, a single word-pair will be encoded for each document, independent of both the encoding region type (sliding vs. fixed) and size. Within the previously defined modeling framework, this limits two key modeling choices to (1) whether to use a WW or WD representational structure, and (2) whether or not to use an abstraction algorithm such as SVD.

In designing our toy datasets, we wished to explore which *types* of semantic relationships between words were captured by different manipulations in terms of the semantic models. In particular, we designed each dataset such that it captured (1) associativity: words with which a target word directly co-occurs, (2) substitutability: words that have similar co-occurrence patterns to a target word, and (3) categorical-relationships: words which co-occur with similar *types* of words to the target word.

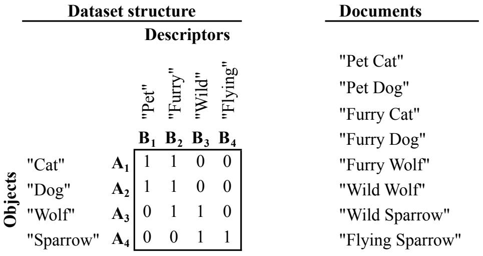
To make this more concrete, consider the example dataset represented in Figure 2. Words in this dataset belong to one of two syntactic categories: objects or descriptors. We limit the existing word pairs in the dataset such that objects only co-occur with descriptors (as in, e.g., the sentences "*pet cat*", "*pet dog*" and "*wild wolf*"). Of all 16 possible object-descriptor pairs, only the pairs with an indicator value of 1 in fact co-occur in the dataset. By doing so, we build two

	Encoding Region		**Representational**	**Representational Transformation**		
Model	Type	Size	**Structure**	Normalization	Dimensionality-Reduction	**Reference**
HAL	Sliding	10 Words	WW	Row-sum	None	Lund & Burgess, 1996
COALS	Sliding	10 Words	WW	Correlational	Singular-Value Decomposition	Rohde, Gonnerman, & Plaut, 2009
BEAGLE	Fixed	Sentence	WW	None	Random Vector Accumulation	Jones, Kintsch, & Mewhort, 2006
LSA	Fixed	Document	WD	Log-Entropy	Singular-Value Decomposition	Landauer & Dumais, 1997
Topic Model	Fixed	Document	WD	None	Latent Dirichlet Allocation	Griffiths, Steyvers & Tenenbaum, 2007

Table 1: Situating several semantic models within the organizational framework

types of semantic information into the data: associativity and substitutability. Words with associative relationships in the dataset are word-pairs with values of 1 (e.g. *dog* and *pet* are associated, whereas *sparrow* and *furry* are not). Words with substitutable relationships in the dataset are word-pairs that have similar sets of associative relationships (e.g. *cat* and *dog* are perfectly substitutable in this dataset since they both only co-occur with *pet* and *furry*, whereas *dog* and *wolf* are partially substitutable). Words with a categorical relationship are words that co-occur with the same type of word, regardless of substitutability (e.g. s*parrow* belongs to the same category as *dog* and *cat* despite it not sharing a single associate, because it co-occurs with other descriptors and not with other objects).

In Figure 3, we show all dataset structures used in generating our artificial datasets. These corpora were designed such that they captured a range of associative, substitutability, and categorical relationships, across a range of category-sizes. The question of interest here is: what modeling manipulations allow a model to pick up on the three different relationships captured by the structure of these datasets.



$$\begin{array}{llcccc} & & B_1 & B_2 & B_3 & B_4 \\ \text{"Cat"} & A_1 & 1 & 1 & 0 & 0 \\ \text{"Dog"} & A_2 & 1 & 1 & 0 & 0 \\ \text{"Wolf"} & A_3 & 0 & 1 & 1 & 0 \\ \text{"Sparrow"} & A_4 & 0 & 0 & 1 & 1 \end{array}$$

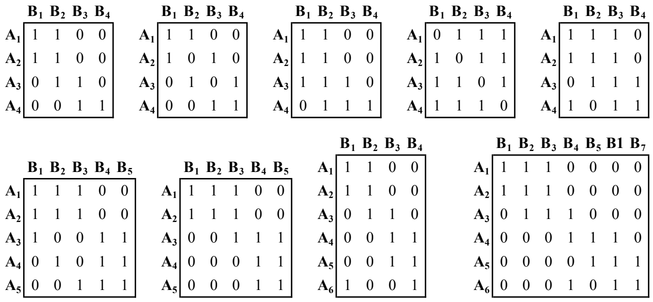
"Pet Cat"
"Pet Dog"
"Furry Cat"
"Furry Dog"
"Furry Wolf"
"Wild Wolf"
"Wild Sparrow"
"Flying Sparrow"

Figure 2: Example of design and construction of artificial datasets.

3.1. Methods

3.1.1. Dataset generation: Corpora were generated using the associative structures illustrated in Figure 3. Each dataset (which we will refer to as a corpus) consisted of a set of documents, each of which contained a single pair of words. Within each corpus, only the word-pairs indicated in the figure were represented. Frequencies of each pair of words were adapted such that all words from category A had equal frequencies (words in category B had equal frequencies in about half of the corpora). The nine corpora were designed such they each capture a range of patterns of the three distinct types of word relationships described above, while additionally varying in factors such as category size. This was done to ensure that our findings were consistent across a variety of data.

3.1.2. Models: Since each document within our corpora consisted of only a single word-pair, this eliminated the need to define or manipulate the encoding regions in our models. This left us with two primary factors along which models could vary: the type of encoding structure used and whether or not they employed an abstraction algorithm. As previously discussed, there are two types of encoding structures used in semantic modeling (WW and WD), and a wide variety of abstraction algorithms. We limit our exploration of abstraction here to the use of Singular Value Decomposition, because of its rich history in the semantic model literature (e.g., Landauer & Dumais, 1997; Rohde et al., 2009). As shown in Table 2, this two-by-two space of semantic models under consideration thus encapsulates three models that have been employed in the psychological literature, as well as the Vector-Space Model (VSM) from information retrieval (Salton, Wong & Yang, 1975). To control for other possible ways in which the models could vary, we employ the cosine-similarity metric and row-normalization for all models[2].



$$\begin{array}{lcccc} & B_1 & B_2 & B_3 & B_4 \\ A_1 & 1 & 1 & 0 & 0 \\ A_2 & 1 & 1 & 0 & 0 \\ A_3 & 0 & 1 & 1 & 0 \\ A_4 & 0 & 0 & 1 & 1 \end{array}
\quad
\begin{array}{lcccc} & B_1 & B_2 & B_3 & B_4 \\ A_1 & 1 & 1 & 0 & 0 \\ A_2 & 1 & 0 & 1 & 0 \\ A_3 & 0 & 1 & 0 & 1 \\ A_4 & 0 & 0 & 1 & 1 \end{array}
\quad
\begin{array}{lcccc} & B_1 & B_2 & B_3 & B_4 \\ A_1 & 1 & 1 & 0 & 0 \\ A_2 & 1 & 1 & 0 & 0 \\ A_3 & 1 & 1 & 1 & 0 \\ A_4 & 0 & 1 & 1 & 1 \end{array}
\quad
\begin{array}{lcccc} & B_1 & B_2 & B_3 & B_4 \\ A_1 & 0 & 1 & 1 & 1 \\ A_2 & 1 & 0 & 1 & 1 \\ A_3 & 1 & 1 & 0 & 1 \\ A_4 & 1 & 1 & 1 & 0 \end{array}
\quad
\begin{array}{lcccc} & B_1 & B_2 & B_3 & B_4 \\ A_1 & 1 & 1 & 1 & 0 \\ A_2 & 1 & 1 & 1 & 0 \\ A_3 & 0 & 1 & 1 & 1 \\ A_4 & 1 & 0 & 1 & 1 \end{array}$$

$$\begin{array}{lccccc} & B_1 & B_2 & B_3 & B_4 & B_5 \\ A_1 & 1 & 1 & 1 & 0 & 0 \\ A_2 & 1 & 1 & 1 & 0 & 0 \\ A_3 & 1 & 0 & 0 & 1 & 1 \\ A_4 & 0 & 1 & 0 & 1 & 1 \\ A_5 & 0 & 0 & 1 & 1 & 1 \end{array}
\quad
\begin{array}{lccccc} & B_1 & B_2 & B_3 & B_4 & B_5 \\ A_1 & 1 & 1 & 1 & 0 & 0 \\ A_2 & 1 & 1 & 1 & 0 & 0 \\ A_3 & 0 & 0 & 1 & 1 & 1 \\ A_4 & 0 & 0 & 0 & 1 & 1 \\ A_5 & 0 & 0 & 0 & 1 & 1 \end{array}
\quad
\begin{array}{lcccc} & B_1 & B_2 & B_3 & B_4 \\ A_1 & 1 & 1 & 0 & 0 \\ A_2 & 1 & 1 & 0 & 0 \\ A_3 & 0 & 1 & 1 & 0 \\ A_4 & 0 & 0 & 1 & 1 \\ A_5 & 0 & 0 & 1 & 1 \\ A_6 & 1 & 0 & 0 & 1 \end{array}
\quad
\begin{array}{lccccccc} & B_1 & B_2 & B_3 & B_4 & B_5 & B1 & B_7 \\ A_1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ A_2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ A_3 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ A_4 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ A_5 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ A_6 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}$$

Figure 3: Illustration of structure for all artificial datasets

Table 2: The experimental models employed, and approximate corresponding models from literature

		Structure	
		WW	WD
Abstraction	No-SVD	HAL	VSM
	SVD	COALS	LSA

3.1.3. Model Evaluations: To evaluate which models captured the three previously described relationships, we provide formal definitions of each[3]:

Associativity: The extent to which a pair of words locally co-occurs. As a measure of a pair of word's "true" associativity, we use the Pairwise Mutual Information measure. This measure has been employed previously in both the psychological and machine-learning literature. Intuitively, it corresponds to the observed probability with which two words co-occur relative to their expected co-occurrence probability: $PMI = log \frac{p(w_1,w_2)}{p(w_1)*p(w_2)}$

Substitutability: The extent to which two words have similar co-occurrence patterns. We measure this using the Jensen-Shannon divergence (a measure of the similarity of two probability distributions) between each word's probability of co-occurrence across all other words. To give some examples, consider the words from category "A" in Figure 2. The probability distribution of co-occurrences for "dog" is equivalent to that for "cat" (with *p=.5* for both *pet* and *furry*); these words' have a JS-divergence of 0. The JS-Divergence for *dog* and *wolf* (which share one associate) equals .5, and for *dog* and *sparrow* (which share no associates) equals 1. We transform this value into a similarity using:

JS-Similarity = 1 - JS-Divergence.

[2] The broad trends presented in our results are consistent across cosine, city-block, and correlational similarity metrics.

[3] We do not wish to argue the case for whether these are the "proper" or "true" definitions of these different types of relationships. However, the types of relationships we describe have a basis in both statistical measures of text and the psychological literature e.g., see (Jones et al., 2006), and furthermore capture intuitive psychological aspects of semantics.

Categorical: A binary measure of whether the two words belong to the same category. Using the example given in Figure 2, *dog* has a categorical similarity of 1 to all object-words in category A, and a similarity of 0 to all descriptor-words in category B.

Each model generates only a single set of predictions, and these predictions may conflate the different relationships (e.g., a model's similarity metric might pick up on both substitutability and associativity). However, the design of our datasets is such that we are able to evaluate each model's ability to pick up on different relationships independently.[4] In particular, each word only associates with words from opposite categories, but will only *share* associates with words from within its own category.

The emphasis of the present experiments is theoretical (i.e., to determine which models are *capable* of capturing which aspects of similarity), rather than practical. In light of this, we make two choices with respect to model-evaluation that emphasize ceiling-performance rather than performance that might be expected in real-world conditions. In particular: (1) during evaluation, we only evaluate a model's ability to pick up on substitutability within categories, and to pick up on associativity between categories (i.e., the relationships between model-predictions of PMI and JS-Similarity are only evaluated for word-pairs relevant to the task)[5], and (2) we evaluate models that employ SVD with respect to their *best* performance on a given task, across all dimensionalities for which the singular values is greater than zero (that is, for an SVD model with seven dimensions that account for variance in the dataset, we compute the best performance among the six sets of predictions generated by the model using between two and seven dimensions). For evaluating the category-based relationships, we compare the model's similarity score and a binary variable—indicating if two words belong to the same category—across all words.

For each of the prediction tasks (predicting associativity, substitutability, and categories), we evaluate a model's ability to pick up on each word's pattern of relationships using Spearman's rank correlation. For example, to evaluate whether a model picks up on the pattern of associativity for the word *dog* in the dataset shown in Figure 2, we compute the rank correlation between a model's predicted similarities and the PMI in the dataset between *dog* and all words in category B. These rank-correlations are then averaged across all words within a dataset. For models employing SVD, the best-performing model on this task is taken from among all dimensionalities.

3.2. Results

For all three types of word-similarities we defined, the average rank-correlation between model predictions and true word-similarities (across all corpora) is shown in Table 3. These results indicate clear main effects as well as interactions between modeling manipulations and the types of word relationships that a model captures.[6]

Table 3 Average rank-correlation between all model similarities and the three word relationships across all corpora

	Associativity		**Substitutability**		**Categorical**	
	WW	**WD**	**WW**	**WD**	**WW**	**WD**
No-SVD	.00	1.00	1.00	.15	.81	-.22
SVD	.17	.94	.92	.88	1.00	.15

First, these results illustrate that similarities computed from raw WD matrices perfectly capture the associativity between two words. This is because the word-vector within a WD matrix simply encodes the instances in which the word has occurred, and the extent to which this vector is aligned between a pair of words captures the relative frequency with which they co-occur. Furthermore, the raw WW matrix perfectly captures the extent to which words are substitutable. This is because the rows within the WW matrix capture each word's patterns of co-occurrence. Additionally, since the category-membership was defined by the set of valid words with which a word could co-occur with in each dataset, the raw WW-matrix picks up on category membership to the extent that category-membership is correlated with substitutability.

Employing SVD as an abstraction method significantly affects model performance for both the substitutability and category-membership measures. The ability of the WD matrix to capture substitutability dramatically improves when SVD is employed, and achieves near perfect performance. To give a concrete example of how this is achieved, refer back to the design shown in Figure 2. In the raw-document space, the cosine-similarities between words within a category are always equal to zero except when comparing a word to itself (due to the fact that this matrix picks up *only* on word-associativity). However, the similarities in the first two dimensions of the word-space after performing SVD perfectly capture the relative substitutability of all words within their categories except for A_3 and B_1. This is due to the fact that the SVD process uses its first dimensions to encode as much variance in the dataset as possible. In this case the most variance can be accounted for by collapsing across documents with partially overlapping *object* or *descriptor* words. It is important to note that within a single choice of dimensionality, the model ends up conflating substitutability and associativity; e.g., if one were to use just the first two dimensions of the SVD to predict word-associativity, the average rank correlation between model-similarities and word-associativity on dataset 2 would be just .59 (but using either 4 or 5 dimensions gives the observed performance of .94). This result is consistent across the different datasets; the SVD of the WD matrix picks up primarily on substitutability using the first few dimensions, and picks up on associativity in higher dimensions (as it more closely approximates the

[4] Although the extent to which the models may weight different aspects of similarity is of both theoretical and practical interest, it is not the focus of the current experiments

[5] For JS-Similarity we furthermore do not include the item's self-similarity, as this is greatly over-estimates model-performance, since both values will always equal one.

[6] We note here that the results were highly consistent across all nine corpora, and did not interact with corpus features such as category-size.

original space, which captures associativity). This is why the associativity score does not dramatically worsen when moving from a raw to SVD representation.

Within the WW matrix, employing an SVD allows the model to perfectly capture the category-membership of all words. This is an interesting result, since it indicates that this model has the ability to generalize across category members, despite the fact that in some cases they have orthogonal patterns of associativity; e.g., in Figure 2 the pattern of *sparrow* is orthogonal to both *dog* and *cat*, but the model nonetheless picks up on the fact that this word associates with only members of category B and is therefore a member of category A. As with the WD matrix capturing substitutability, category membership is entirely captured within the first few dimensions of the reduced matrix (typically the first 2 dimensions). Since perfectly capturing the rank-ordering of category members necessitates that all within-category members have equal similarity, the SVD-reduced matrix does not pick up on substitutability at these lower-dimensionalities (for a single set of predicted similarities, if performance on the category task is 1, performance on the substitutability task is zero). However, just as the WD matrix picks up again on associativity as more dimensions are included, the WW matrix picks up on again on substitutability as more dimensions are included.

Since the best performing dimensionality is used separately for each task, performance for the SVD-reduced WW matrix significantly improves on the category task while hardly being impacted on the substitutability task. It is important to note, however, that at any individual dimensionality, the SVD-reduced WW matrix could not perform as well as is shown in Table 3 on both the substitutability and category tasks. Similarly, the SVD-reduced WD matrix could not perform as well on both associativity and substitutability tasks using a single dimensionality.

Lastly, we get striking failures for both the WW and WD representational structures in their ability to capture specific types of relationships. The WD matrix—using either a raw representation or an optimally reduced dimensionality—fails to pick up on category-membership. The WW matrix likewise fails to ever pick up on associativity.

4. Discussion

In this paper, we presented a general framework for organizing the space of semantic models, and identified a number of existing models within this space. We then demonstrated how this framework is useful for guiding experimental work into modeling semantic structure. In particular, we showed that by isolating and comparing individual components within the framework, we can identify how specific manipulations influence a model's ability to capture different aspects of semantic structure.

Using artificial data generated using a known structure, we showed that both a model's representational structure (WW vs. WD) and its use of dimensionality reduction have specific consequences in terms of a model's ability to capture types of different kinds of relationships between words. In particular, without dimensionality-reduction, a WW representation captures the substitutability between two words, whereas a WD representation captures the associativity between the words. Employing an abstraction process like SVD on a WW matrix allows it to induce category-level relationships, even when the two words' patterns of associativity are orthogonal. Employing an SVD on the WD matrix allows it to capture the substitutability of words in addition to their associativity. However, both structures have their own unique limitations: the similarity between words composed of WW structures can not pick up on associativity, and the WD matrix can not pick up on categorical-similarities, whether or not an SVD is used.

Our results indicate that a single semantic model's predictions may be insufficient to capture the full range of semantic relationships that people are able to represent. This suggests that a valuable direction for future research may be in embedding multiple representational structures and abstractions within a larger model.

It remains to be seen how these findings will interact with manipulations along other dimensions in our framework. For example, while we have shown that a WW structure is unable to capture associativity when the model's encoding regions are restricted to individual word-pairs, this should change as encoding regions increase in size. For example, using a larger encoding region should allow a WW model's row representations to indirectly capture word-associativity via second order co-occurrences; when encoding the phrase "pet dog chased", the word *chased* would be encoded within the rows for both *dog* and *pet*. Since WW matrices pick up on co-occurrence patterns, they could indirectly capture the associativity between *pet* and *dog* via their mutual co-occurrence with *chased*. But while a larger encoding region may increase performance with respect to associativity, performance with respect to other types of word relationships may suffer. This leaves many open questions regarding how other manipulations within the space of models we described will qualitatively affect performance. Additional results such as those presented within this paper should serve to constrain the types of psychological processes one might posit for *how* a model captures particular aspects of human behavior.

Acknowledgements

This work was supported by NSF BCS-1056744 and NIMH-096906.

References

Griffiths, T. L., Steyvers, M., & Tenenbaum, J. B. (2007). Topics in semantic representation. *Psychological Review*, *114*, 211-244.

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

Jones, M. N., Kintsch, W., & Mewhort, D. J. K. (2006). High-dimensional semantic space accounts of priming. *Journal of Memory and Language*, *55*, 534-552.

Landauer, T., & Dumais, S. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, *104*, 211-240.

McRae, K., & Jones, M. (2013). 14 Semantic Memory. *The Oxford Handbook of Cognitive Psychology*, 206.

Rohde, D. L. T., Gonnerman, L., & Plaut, D. C. (2009). An improved model of semantic similarity based on lexical co-occurrence. *Cognitive Science*.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11), 613-620.