COGNITIVE SCIENCE
A Multidisciplinary Journal

Cognitive Science 43 (2019) e12730

ISSN: 1551-6709 online
DOI: 10.1111/cogs.12730

The Role of Negative Information in Distributional Semantic Learning

Brendan T. Johns,[a] Douglas J. K. Mewhort,[b] Michael N. Jones[c]

[a]*Department of Communicative Disorders and Sciences, University at Buffalo*
[b]*Department of Psychology, Queen's University*
[c]*Department of Psychological and Brain Sciences, Indiana University*

Received 14 February 2018; received in revised form 18 January 2019; accepted 25 March 2019

Abstract

Distributional models of semantics learn word meanings from contextual co-occurrence patterns across a large sample of natural language. Early models, such as LSA and HAL (Landauer & Dumais, 1997; Lund & Burgess, 1996), counted co-occurrence events; later models, such as BEAGLE (Jones & Mewhort, 2007), replaced counting co-occurrences with vector accumulation. All of these models learned from positive information only: Words that occur together within a context become related to each other. A recent class of distributional models, referred to as neural embedding models, are based on a prediction process embedded in the functioning of a neural network: Such models predict words that should surround a target word in a given context (e.g., *word2vec*; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013). An error signal derived from the prediction is used to update each word's representation via backpropagation. However, another key difference in predictive models is their use of negative information in addition to positive information to develop a semantic representation. The models use negative examples to predict words that should *not* surround a word in a given context. As before, an error signal derived from the prediction prompts an update of the word's representation, a procedure referred to as negative sampling. Standard uses of *word2vec* recommend a greater or equal ratio of negative to positive sampling. The use of negative information in developing a representation of semantic information is often thought to be intimately associated with *word2vec*'s prediction process. We assess the role of negative information in developing a semantic representation and show that its power does not reflect the use of a prediction mechanism. Finally, we show how negative information can be efficiently integrated into classic count-based semantic models using parameter-free analytical transformations.

Keywords: Distributional semantics; Cognitive modeling; Natural language processing; Big data; Machine learning

Correspondence should be sent to Brendan Johns, Department of Communicative Disorders and Sciences, University at Buffalo, 122 Carey St., Buffalo, NY 14214. E-mail: btjohns@buffalo.edu

1. Introduction

Landauer and Dumais's classic Latent Semantic Analysis (LSA; 1997) model challenged the field of lexical semantics to re-think what knowledge can be extracted from linguistic experience and what has to be built into a cognitive system—"Plato's problem" in their provocative title. The introduction of LSA led to a new a class of models of semantic memory being developed, entitled distributional models, which have focused on how much knowledge can be extracted from large text corpora.

Distributional models of semantics learn the meanings of words based on patterns of word co-occurrence within a large sample of language. The underlying idea is that words that occur in similar contexts have similar meanings, an idea based in both the philosophy of language and in linguistics (Harris, 1954; Wittgenstein, 1953). Current distributional models use different mechanisms to acquire semantic knowledge, including count-based methods (e.g., Bullinaria & Levy, 2007, 2012; Lund & Burgess, 1996; Recchia & Jones, 2009), matrix decomposition techniques (e.g., Landauer & Dumais, 1997), probabilistic inference (Griffiths, Steyvers, & Tenenbaum, 2007), vector-accumulation/noise-cancellation methods (e.g., Jones & Mewhort, 2007; Recchia, Sahlgren, Kanerva, & Jones, 2015), and retrieval-based mechanisms (Jamieson, Johns, Avery, & Jones, 2018; Johns & Jones, 2015; Kwantes, 2005). All models make different assumptions about how co-occurrence information is used, but all use co-occurrence as the fundamental building block when learning semantic representations. The idea is that word co-occurrence in the natural language environment provides sufficient information to support the development of a representation for the meaning of words.

A new class of distributional models—referred to as neural embedding models—are based upon established connectionist principles and use standard backpropagation methods (Rumelhart, Hinton, & Williams, 1986) but add active sampling to train the network to predict the words that should occur with a target word in its context[1] (e.g., *word2vec*; Mikolov, Chen, Corrado, & Dean, 2013; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013). The occurrence of a target word within a window—its size is defined by a free parameter—prompts the model to predict accompanying context words that should occur within that window. Incorrect predictions provide an error signal used to bring the hidden weights of the network closer to the base rate of environmental co-occurrence of the training corpus.

Unlike LSA and other distributional models, *word2vec* uses negative information to refine the word's semantic representation, a technique called negative sampling. Negative sampling operates by randomly sampling a number of unrelated words based on word frequency and training the network to suppress those words in the predicted output layer. The idea is that the network should be able to predict the words with which a target word should occur and to inhibit unrelated words.

Mandera, Keuleers, and Brysbaert (2017) have argued that *word2vec* is a realistic cognitive model, and more plausible than earlier models. They show, using multiple corpora across *word2vec*'s parameter space, that *word2vec* outperforms count-based models (cf. Baroni, Dinu, & Kruszewski, 2014). They also proposed that *word2vec* is more

cognitively plausible, as it solves both a computational-level and algorithmic-level problem within Marr's (1982) hierarchy. The latter point follows from *word2vec*'s use of a neural network as its underlying representation.

However, negative sampling seems quite implausible from a cognitive perspective, as it requires a person to generate a number of unrelated words each time a word is encountered in the linguistic environment. Additionally, although Mandera et al. (2017) found a sizeable advantage for *word2vec* over count-based models, other studies have found smaller differences (e.g., Asr, Willits, & Jones, 2016; De Deyne, Perfors, & Navarro, 2016; Levy, Goldberg, & Dagan, 2015; Recchia & Nulty, 2017), while Demski, Ustun, Rosenbloom, and Kommers (2014) report that a modified BEAGLE model (Jones & Mewhort, 2007) algorithm outperforms *word2vec* on analogy tasks.

This is not to say that negative information is not used in language learning and organization. Using a framework co-opted from animal learning models, Ramscar, Hendrix, Shaoul, Milin, and Baayen (2014) and Ramscar, Sun, Hendrix, and Baayen (2017) have recently demonstrated that as positive associations are formed between words, there are corresponding negative associations being formed among words with which the target word does not co-occur. Ramscar et al. (2017) demonstrated empirically that the buildup of the negative associations has important consequences for performance on a paired-associate learning task. Thus, even though negative sampling may seem questionable from a mechanistic point of view, the integration of negative information into a model of distributional semantics has empirical support.

There are two conflated differences between *word2vec* and classic models of semantics. The first is *word2vec*'s architecture, a predictive neural network with error correction. The second is the use of negative information in developing semantic representations. Because classic count-based models have not explicitly integrated negative information into their representations, it is difficult to determine how much of *word2vec*'s success is attributable to its predictive connectionist architecture and how much is attributable to the use of negative information.

Negative information is a natural way to hone a prediction mechanism: A negative error signal could potentially be as informative as a positive one.

The goal of the current work is to clarify the contribution of negative information in distributional semantics and to show how negative information can be integrated into classic model representations. To do so, we will start with a very simple count-based distributional model and add negative information to its representation. The goal is to assess how much power comes from negative information and, indirectly, to assess the power contributed by the connectionist architecture. To anticipate the results, we will show that negative information can be a powerful factor in distributional modeling. Additionally, we will describe analytical solutions designed to integrate negative information into a word's semantic representation.

There are multiple ways to interpret negative information in distributional semantics. For example, Landauer, Foltz, and Laham (1998) noted that LSA implicitly uses the lack of connection between two words to infer that those words are unrelated: "... the aggregate of all the word contexts in which a given word does and does not appear provides a

set of mutual constraints that largely determines the similarity of meaning of words . . ." (p. 259). The goal of the first part of this article is to understand the impact of explicit negative sampling, which we define as the generation of unrelated words that are used to update a target word's representation.

Given the success of *word2vec*, it is important to understand how negative information works. Curiously, it has mostly defied traditional mathematical analysis. For example, Goldberg and Levy (2014) conducted a formal analysis of the role of negative sampling and concluded by asking "Why does this produce good word representations?" Their answer was unusually candid: "Good question. We don't really know" (p. 5). Although subsequent work has elucidated the role of negative information somewhat (see Levy et al., 2015), the role of negative information in distributional semantics is still an open question. The first section of this article will answer this challenge.

The second part of the article will focus on using the information gleaned from the first analysis to allow standard models of distributional semantics to integrate negative information without the need for an explicit sampling mechanism, simplifying the approach.

The overall goal of this article is to provide an understanding of how both positive and negative information combine to account for semantic behavior. The hope is that by gaining an explicit understanding of how different information sources interact in forming semantic representations, better models can be developed, both in terms of the model's power and in terms of conceptual clarity, providing new pathways for theoretical and empirical work in lexical semantics to understand how humans learn and represent distributional semantics.

2. Modeling framework

To assess the power of integrating negative information into a distributional model, we need a simple model in which we can manipulate the use of this information source. Accordingly, we used a very simple model, namely a word-by-word frequency matrix (referred to as the WW model in the below simulations). In the model, each row is a word's semantic representation, which encodes the frequency distribution of co-occurrences with other words in context. Counting is restricted to a moving window of n-words (n being a free parameter) within a sentence in a corpus. In a word-by-word representation, the semantic similarity of two words is their overlapping co-occurrence patterns with other words. Hence, similarity between two words can be assessed using a vector cosine (normalized dot product) between their respective rows. The word-by-word frequency model shares assumptions with the classic HAL model (Lund & Burgess, 1996; see also Hills, Maouene, Riordan, & Smith, 2010). The COALS model of Rohde, Gonnerman, and Plaut (2006; see also, Chang, Furber, & Welbourne, 2012) offers a similar view of lexical semantics.

The standard count-based approach contains pointwise mutual information (PMI; Church & Hanks, 1990). PMI is a direct measure of the probability of two words

occurring in the same context, and it has been shown to provide an excellent account of lexical behaviors, especially word similarity measures (Bullinaria & Levy, 2007, 2012; Levy & Goldberg, 2014a,b; Levy et al., 2015; Recchia & Jones, 2009; Recchia & Nulty, 2017). Hence, we elected a word-by-word representation because it is concise and will allow any role of negative information to be well understood.

Levy and Goldberg (2014b) describe how to integrate negative information into PMI by taking out a constant amount from the resulting metric (see equations 5 and 6 below). Their transformation increases performance but does not explain how negative information helps to construct accurate semantic representations of words. Because it is based upon probabilities of occurrence, negative sampling cannot be built into a PMI measure directly; obviously, probability of occurrence must be nonnegative. In the following sections, we will contrast our proposals with that of PMI.

Within the word-by-word frequency matrix framework, positive information will be learned by increasing the strength of two words that occur with each other in the same context (defined as a window size within a sentence) by adding 1 to both words entry in the matrix upon each co-occurrence. Negative information will also be considered. For each context studied, a set number (represented with the parameter k) of words will be generated randomly for each word in that context; the randomly generated words are, in effect, negative samples. The negative samples will be unique for each target word. The strength between a studied word and a negative sample will be decreased by 1. Thus, for every sentence in a corpus, each studied word will have positive and negative information integrated into its resulting representation.

A unique aspect of *word2vec* is its use of subsampling, a process designed to limit the impact of very high–frequency words, similar to the use of stop lists in other distributional approaches (e.g., Landauer & Dumais, 1997). Subsampling works by probabilistically skipping words relative to their frequency: High–frequency words are assigned a greater probability of being skipped. The expression used to determine the probabilities differs between publications and the publicly available code. We took the function from the C code of *word2vec* to define the probability of a word being included (Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013):

$$P(w_i) = \left(\sqrt{z(w_i)/t} + 1\right) \times \left(\frac{t}{z(w_i)}\right), \tag{1}$$

where $z(w_i)$ is the probability of a word occurring across the whole corpus (frequency of a word divided by the total number of word occurrences) and t is a free parameter.[2] The use of subsampling is an elegant way of removing the use of stop lists that were used in previous models (e.g., Landauer & Dumais, 1997).

Fig. 1 contains sampling probability of the most frequent 6,000 words (after approximately this point no words are subsampled) in the corpus described below, organized by rank word frequency. This figure shows that the subsampling routine used here has a roughly linear increase in a word's probability of being sampled as a function of word

frequency, with the most frequent words being the least likely to be sampled. The use of a subsampling routine allows the impact of very high–frequency words to be mitigated, while allowing lower frequency words to affect a word's representation. By manipulating the parameter *t*, the subsampling distribution is changed. The correct setting for this parameter is likely corpus dependent, as there are significant deviations in frequency distributions by type of corpus used (see Johns & Jamieson, 2018 and Johns, Jones, & Mewhort, 2019, for examples).

In *word2vec*, negative samples are selected based on the frequency distribution of the corpus (Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013). Typically, the distributions are smoothed in some way to reduce the influence of high–frequency words (Levy et al., 2015). Here, negative samples will be selected based on a word's frequency, equivalent to past work. The probability distribution will be smoothed by multiplying a word's frequency value by its subsampling probability before calculating a word's probability of being sampled, reducing the overall impact of high–frequency words. Consistent with past results (e.g., Levy et al., 2015), the smoothing process produces increased performance.

We used a corpus of 20 million sentences, derived by combining Wikipedia articles and non-fiction books (Johns, Jones, & Mewhort, 2016; Johns et al., in press). The corpus consists of approximately 120 million words. The word list for the model will be the 50,000 highest frequency words from the corpus. Increasing the size of the word list had a negligible effect on performance.

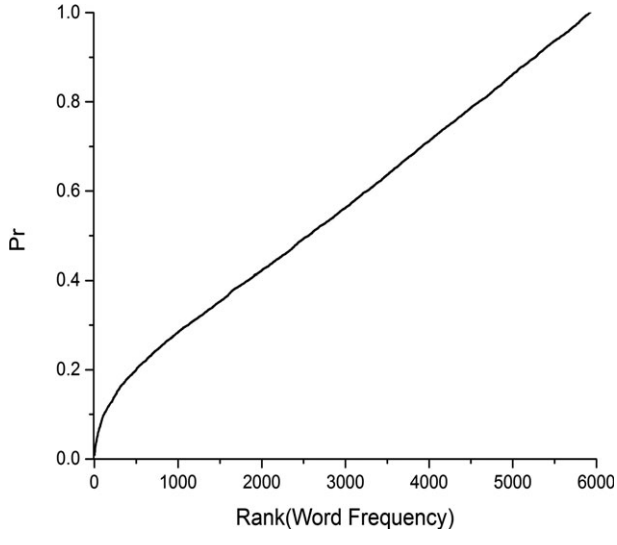

Fig. 1. Sampling probability as a function of rank word frequency. Frequency was assessed with a 20 million sentence corpus consisting of Wikipedia articles and non-fiction books. This figure shows that the subsampling procedure used here results in a roughly linear increase in the probability of a word being sampled. Of note, this sampling procedure only impacts approximately the most frequent 6,000 words.

We used six word relatedness and similarity measures to assess the model's performance. Using labels taken from De Deyne et al. (2016), the measures included (a) the WordSim data (n = 353; Finkelstein et al., 2002), (b) RG1965 data (n = 65; Rubenstein & Goodenough, 1965), (c) the MTURK-771 data (n = 771; Halawi, Dror, Gabrilovich, & Koren, 2012), (d) the MEN data (n = 3,000; Bruni, Boleda, Baroni, & Tran, 2012), (e) the SimLex-999 data (n = 999; Hill, Reichart, & Korhonen, 2016), and (f) the Radinsky-2011 dataset (n = 287; Radinsky, Agichtein, Gabrilovich, & Markovitch, 2011).

Our goal is to evaluate the power of integrating negative information into a count-based co-occurrence representation. Because *word2vec* is based on prediction (Mandera et al., 2017)—the model predicts words that it should and should not occur with a target word—it is unclear whether negative sampling should work in a model based on a word-by-word frequency representation. Active prediction is not used in such a model. If the model does see an increase in performance, it would signal that the role of negative information in distributional semantics does not reflect a prediction mechanism, or underlying learning framework, but instead results from some other aspect of the statistical structure of the language environment.

3. Results

The word-by-word frequency matrix model has two free parameters: window size and the number of negative samples. The first simulation reports the fit across the parameter space for both parameters using the complete WordSim data set (Finkelstein et al., 2002).

Fig. 2 shows variance accounted for as a function of the number of negative samples and the window size. When negative sampling was used (i.e., when the negative-sampling parameter is > 0), there was a rapid increase in variance accounted for. The best performance at each number of negative samples was not overly impacted by window size, and for each window size, the optimal number of negative samples was one less than the window size. Because some sentences are shorter than the window size, this pattern means that optimal performance is given with a roughly equal sampling of positive and negative information. When the amount of negative information exceeded the positive information, there was a rapid reduction in performance. The best combination had a window size of 4, with 3 negative samples per word, $r = 0.7$, $p < .001$, an impressive correlation for the data, given the simplicity of the underlying representation.

To confirm that an advantage with negative sampling extends to other datasets, Fig. 3 shows performance for seven datasets noted earlier.[3] In each case, we used an optimal window size and number of negative samples. As is shown in Fig. 3, the addition of negative sampling into a very simple co-occurrence framework produced a substantial increase in the variance explained by the model. The increase in fit as a function of negative samples is coherent with a variety of studies of *word2vec*'s performance (e.g. Levy & Goldberg, 2014a,b; Levy et al., 2015); they have shown that increasing (and optimizing) the network's number of negative samples significantly increases the model's performance.

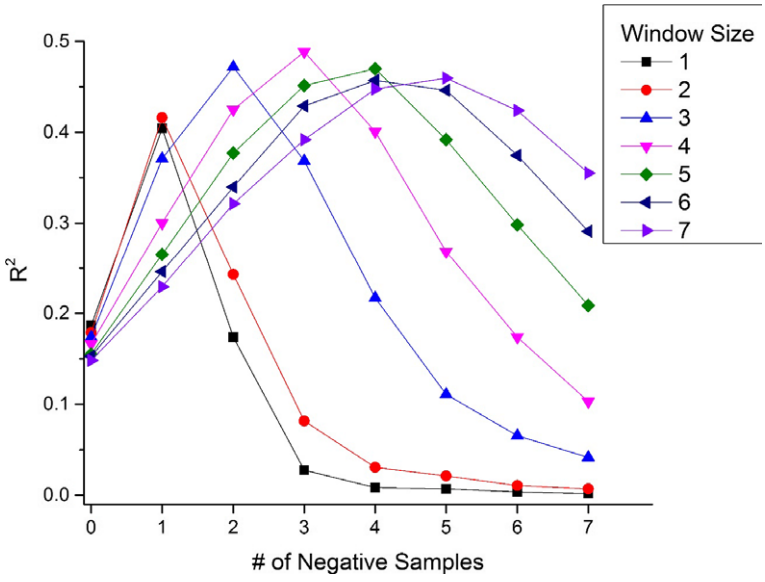

Fig. 2. Variance accounted for by the WW model with negative sampling for the Finkelstein et al. (2002) data. This figure shows a massive increase in the capabilities of the model when negative sampling is introduced. However, there is massive drop in model performance when the number of negative samples exceeds the window size (amount of positive information).

Why does negative sampling add so much power? The standard explanation is that negative information helps *word2vec* to hone its prediction mechanism via error correction. However, that explanation could not hold for the representation used here.

To consider why negative information has such a powerful impact, we constructed two sampling procedures to assess the standard frequency-based sampling. The first used a frequency distribution from a different corpus. Using a distribution from a different corpus checks whether the success of negative sampling depends on the actual construction of the training corpus. Two corpora were used: (a) a related corpus consisting of 20 million sentences from different non-fiction books and Wikipedia articles; (b) a corpus consisting of 20 million sentences from books of fiction, an unrelated corpus (corpus described in Johns et al., 2019). The second sampling procedure used uniform sampling, in which each word in the vocabulary has an equal probability of being selected. Both manipulations were implemented using a window size of 4, with the number of negative samples being manipulated.

Fig. 4 shows the variance accounted for as a function of the number of negative samples and the sampling algorithm. Note that uniform sampling produced very little change in the model's fit (a point also noted by Mikolov, Sutskever, et al., 2013). The null effect suggests that negative sampling requires information about the distributional structure of the language from which the model is learning in order to aid in semantic learning. A direct test of this idea is to use a frequency distribution from an unrelated corpus. When

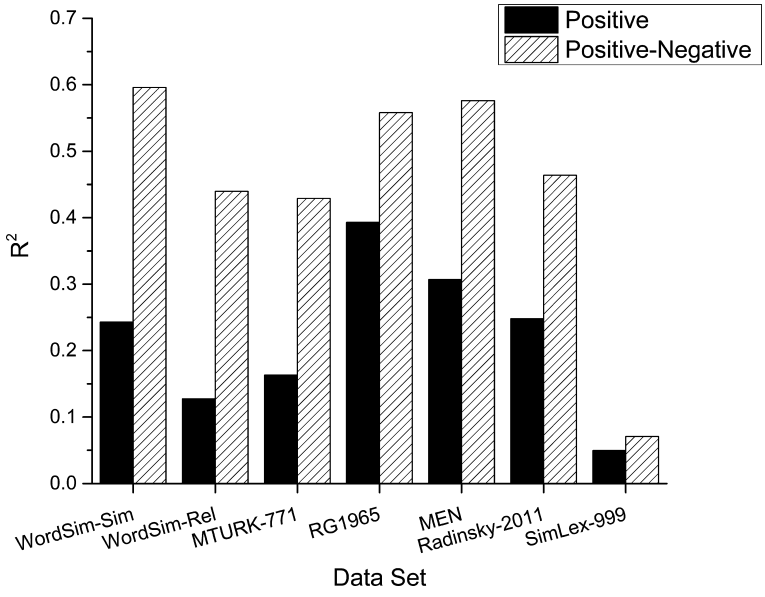

Fig. 3. Variance accounted for across seven word similarity/relatedness datasets for the word-by-word frequency-matrix model as a function of information used (only positive only or positive and negative information).

using an unrelated corpus of fiction books to drive sampling, there was a very small improvement in performance. The result suggests that, unless the frequency distribution maps onto the training corpus, negative sampling offers very little improvement. The results using a related corpus tap into the question explicitly. As Fig. 4 illustrates, there was an improvement when using a corpus that had a similar construction to the training corpus but that had different underlying materials. That said, the improvement was small compared to using the actual training corpus to drive the sampling procedure.

The results shown in Fig. 4 strongly suggest that the negative sampling success depends on the underlying distributional structure of language; it does not rely upon a prediction mechanism.

3.1. Discussion

The data in Figs. 2 and 3 demonstrate that building negative sampling into a very simple word-by-word frequency model produces a substantial increase in performance. The results in Fig. 4 demonstrate that negative sampling depends on the distributional structure of the corpus, not on prediction mechanisms.

The corpus is key. When negative sampling maps onto training materials, there is a balance between positive and negative information: the positive information increases the relationship between correct associates, and the negative information decreases the connection to unrelated words. Both happen systematically for each word. When the sampling procedure is altered, the balance between positive and negative information falls

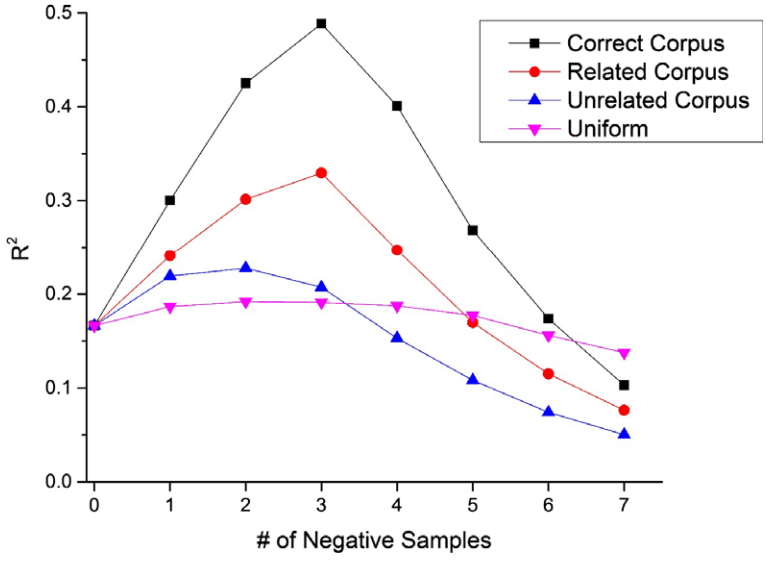

Fig. 4. Variance accounted for as a function of the number of negative samples and the sampling regime. Uniform sampling had very little impact on performance. Sampling using the frequency distribution of an unrelated corpus only caused marginal improvements in fit. There was a moderate increase when words are sampled from a related corpus, but the increase was slight compared to sampling from the training corpus.

apart: Negative information no longer compliments the positive information. As a result, the negative information no longer indicates the structure of the language that is being encoded. When negative information no longer refers to the structure of the language—either because sampling is guided by a uniform distribution or is taken from the wrong corpus—there is very little improvement in performance.

Negative sampling works because it allows the base rate of occurrence to be included in the association between two words. If the amount of positive information between two words vastly exceeds the base rate of occurrence of those words, it signifies that those two words have a strong association. If the positive information does not exceed the base rate, the two words are not associated.

Base-rate information depends on corpus size. Because a large number of samples are needed to establish a base rate, if integration of base rate occurrence drives the success of negative sampling, corpus size should be critical. To test the prediction, we trained a word-by-word frequency matrix model on 2,000,000 sentences, in steps of 200,000, using positive information only and using both positive and negative information. The fit was calculated to the combined data of Finkelstein et al. (2002).

Fig. 5 shows the variance accounted for as a function of corpus size and the kind of information used in learning (positive only vs. positive and negative information). Negative sampling increased performance as a function of corpus size, signaling that, as a base rate of occurrence is formed, negative sampling has an increasingly large effect on the semantic representation. The advantage corresponds to the general findings that *word2vec* models are

more successful with larger corpora (e.g., Levy et al., 2015). Increasing corpus size not only provides more positive samples from which to learn but also allows negative sampling to help form a more accurate base rate of occurrence across all words.

Of course, the base rates are contained within the word-by-word matrix. Hence, a sampling procedure may not even be necessary. If so, a free parameter can be removed.

4. Analytical solutions to negative sampling

The argument of the previous section is that negative sampling helps establish the base rate against which positive occurrence is assessed. Given that base rate information of word occurrence is contained in a word-by-word matrix, it should be possible to exploit base-rate information without relying on a sampling procedure.

An analytical solution would have multiple advantages. First, and most important, it would simplify the model by removing a free parameter, making it a more parsimonious account. Secondly, negative sampling is dubious from a cognitive standpoint, as it seems unlikely that when a person studies a word, she would have to generate a number of unrelated words in order to learn the word under study. Instead, a mechanism that can establish the base rate without relying upon a sampling procedure would simplify the model and would aid in understanding how negative information might be used in cognition.

We developed two analytic solutions. The first is an analog to negative sampling: We added the base rate of occurrence of every word to a word's entry as negative information. We call the model the global negative (GN) approach.

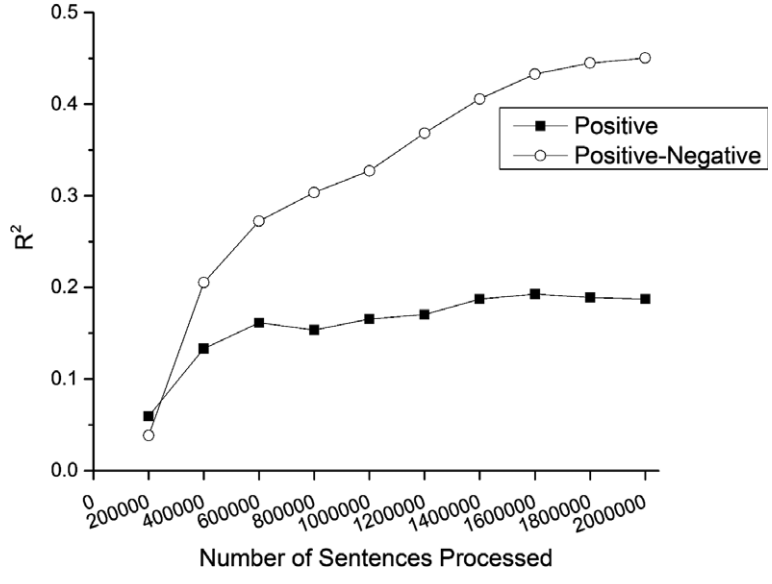

Fig. 5. Variance accounted for as a function of corpus size and the kind of information used in learning (positive only vs. positive and negative information).

The second approach used the distribution of word co-occurrences to infer the relative weight of an association relative to a base rate. We dubbed the method the distribution of associations (DOA) approach. Finally, we combined the two approaches to determine if there is an advantage to using both transformations concurrently.

4.1. Global negative (GN) model

The first step is to construct a global occurrence rate by summing across all columns of the Word-by-Word matrix:

$$\mathbf{GN}_j = \sum_{i=1}^{n} \mathbf{M}_{i,j} \tag{2}$$

where **GN** is the global negative vector, **M** is the word-by-word matrix, j is the column being calculated, and i increments through all n rows in the matrix. It is computed for each column in the matrix. The entries in the **GN** vector are directly related to a word's overall frequency but deviate slightly due to window size. The vector is then unit normalized by dividing each column in the **GN** vector by the total magnitude of the global vector so that the **GN** vector has a total magnitude of 1:

$$\mathbf{GN}_j = \frac{\mathbf{GN}_j}{\sum_{k=1}^{n} \mathbf{GN}_k}, \tag{3}$$

where k increments through each index in the **GN** vector.

Each index in the vector represents the probability of a word occurring in the window of another word during training. The normalized vector will be used to add a base rate of occurrence into a word's entry in the matrix. Given that previous simulations have demonstrated best performance with a roughly equal mix of positive and negative information, we added the **GN** vector into a word's matrix entry proportional to the number of positive occurrences a word has had using the following equation:

$$\mathbf{M}_i = \mathbf{M}_i - \left(\mathbf{GN} * \sum_{j=1}^{n} \mathbf{M}_{i,j} \right), \tag{4}$$

where $\mathbf{M}_i$ is a word's row in the matrix, and j goes through each column in the matrix.

Equation 4 states that for each positive occurrence of a word, add in an equal amount of (global) negative information. Although this could be done continuously, it would be computationally burdensome to do so. Hence, in the data reported here, the transformation will be applied after a word-by-word matrix has been formed. The GN approach does what negative sampling strives to do, that is, to provide the base rate of all associations into a word's representation.

To illustrate the GN transformation, Fig. 6 displays a numerical example of the GN transformation applied to a hypothesized WW matrix of four words {*dog*, *cat*, *car*, *door*}. In step 1, the columns are summed and unit normalized to form the GN vector, and the sum of the rows is taken to assess the amount of positive information each word has accumulated. In step 2, the normalized GN vector is added into each word's row, proportional to the amount of positive information each word had. The result in step 3 is a transformed matrix that has both positive and negative associations. Words that had few co-occurrences (e.g., *dog-car*, *cat-car*, *cat-door*) are negatively associated, while words that had strong contextual overlap (e.g., *dog-cat*, *car-door*) have positive associations. Words that had middling co-occurrences (e.g., *dog-door*) have associations around zero. As the demonstration shows, the GN transformation offers a simple mechanism by which negative information can be incorporated in a word's representation.

4.2. Distribution of association (DOA) model

Instead of adding negative information directly into a word's representation, the DOA transformation uses the distribution of occurrences across the entire matrix. Negative information is contained in the latent structure of the matrix: It is the relative uniqueness of two words occurrence rate, above a base rate, that matters. This is similar to many collocation methods, such as pointwise mutual information (PMI), which weight the association between two words based upon the co-occurrence and overall frequency

STEP 1

WW =	Dog	Cat	Car	Door	Sum
Dog	0	7	1	4	12
Cat	7	0	1	2	10
Car	1	1	0	10	12
Door	4	2	10	0	16
GN =	12/50	10/50	12/50	16/50	50

STEP 2

WW =	Dog	Cat	Car	Door	Sum
Dog	0	7-(.2*12)	1-(.24*12)	4-(.32*12)	12
Cat	7-(.24*10)	0	1-(.24*10)	2-(.32*10)	10
Car	1-(.24*12)	1-(.2*12)	0	10-(.32*12)	12
Door	4-(.24*16)	2-(.2*16)	10-(.24*16)	0	16
GN =	0.24	0.2	0.24	0.32	

STEP 3

WW =	Dog	Cat	Car	Door
Dog	0	4.6	-1.88	0.16
Cat	4.6	0	-1.4	-1.2
Car	-1.88	-1.4	0	6.16
Door	0.16	-1.2	6.16	0

Fig. 6. A numerical example of the GN transformation.

of those words. Specifically, the formula for PMI between words i and j is classically defined as follows:

$$PMI(i,j) = log_2 \frac{P(i,j)}{P(i)P(j)}, \tag{5}$$

where *P(i,j)* is a count of the number of times that the words i and j occur together, *P(i)* is the number of times the word i occurs in the corpus, and *P(j)* is the number of times the word j occurs in the corpus. Bullinaria and Levy (2007) demonstrated that positive PMI (PPMI) provides a better fit than the standard formulation. PPMI simply sets any negative PMI values to 0.

PPMI provides an excellent fit to word similarity data (Bullinaria & Levy, 2007, 2012; Recchia & Jones, 2009), and Levy and Goldberg (2014a,b) offer a method by which negative sampling can be integrated into a PPMI measure (they entitled this measure shifted PPMI, or SPPMI), by taking the log of the negative sampling parameter k from the PMI value of two words:

$$SSPMI(i,j) = \max(PMI(i,j) - \log(k), 0), \tag{6}$$

where $k > 0$. The addition of the parameter k means that for a two words to have a positive PMI value, it must exceed some negative baseline. This transformation has been shown to improve the fit of the method to word similarity data (Levy & Goldberg, 2014a,b; Levy et al., 2015). The SSPMI will be used as a comparison for the performance of the GN and DOA methods.

For the DOA transformation, a count-based transformation similar to PMI will be used, with the modification that the DOA method will consider the distribution of counts within a Word x Word matrix, similar to past suggestions in computational linguistics (e.g., Gries, 2013). Specifically, instead of a direct measure of the co-occurrence overlap between two words, as PMI uses, the DOA measure weighs the connection between two words relative to the connection that the words have to other words contained in memory. Thus, the transformation described below is not unique in the study of distributional semantics, but, consistent with the goal of this paper, conceptualizes the contribution in terms of the interaction of positive and negative information.

The simplest method to weight relative uniqueness is to simply transform each column into a standard (z) score:

$$\mathbf{M}_{i,j} = \frac{\mathbf{M}_{i,j} - \mu_j}{\sigma_j}, \tag{7}$$

where i represents a row in the matrix, j represents a column, μ_j represents the mean of the column, and σ_j represents the standard deviation of the column. The resulting score indicates how many standard deviations the co-occurrence of two words is, over and above the other co-occurrence values, providing a relative weighting for that association.

However, high–frequency words may have generally higher than average co-occurrence values with other words, meaning that their relative weights will also be higher. To normalize for frequency effects, the weights in each row will also be transformed into z-scores, allowing for the relative importance of associations to be sharpened for an individual word:

$$\mathbf{M}_{i,j} = \frac{\mathbf{M}_{i,j} - \mu_i}{\sigma_i}, \tag{8}$$

where μ_i is the mean of a word's row, and σ_i is the standard deviation of that row.

The result is that each word's entry in the matrix is a set of standard scores indicating how strong the relative association is between two words. Higher values indicate a more unique association. The transformation provides a direct measure of how strongly two words are associated over and above the co-occurrence rates of other words. Both the GN and DOA transformations can be thought of as a sharpening process where the important associations of a word are highlighted, by including the co-occurrence relationships of other words into a word's representation. PMI utilizes a similar operation by normalizing a word's count by the total number of occurrences the two words have. Thus, the GN and DOA transformations can be operationalized as more fine-grained approaches to quantifying the unique associations between words, conceptualized as balancing the contribution of positive and negative information in forming semantic representations.

To demonstrate the DOA transformation, Fig. 7 displays the same numerical example as used to demonstrate the GN transformation in Fig. 6. The first step in the DOA transformation is to calculate the mean and standard deviation of each column. In step 2, these are used to transform each count into a z-score, with the score indicating how unique a count is, compared to all other counts in that column. In step 3, the matrix now contains both positive and negative associations, similar to the GN transformation. Step 4 further smooths the matrix by calculating the mean and standard deviation of the z-scores in each row, and transforming the z-scores into z-scores that reflect the other associations that the word has. The resulting matrix in step 5 is the final matrix, and again contains both positive and negative values. In contrast to the GN transformation, the DOA matrix has positive associations only to words that have a strong co-occurrence connection (e.g. *dog-cat*, *car-door*), and all other associations are negative. The simulations contained below will evaluate which transformation best accounts for word similarity data.

4.3. Combination models

The GN and DOA models are obviously connected, but there are some slight differences. The GN approach adds the base rate of word occurrence directly into a word's entry in the matrix. The DOA approach determines the relative weight of an association by transforming co-occurrence values into standard scores, determining the uniqueness of the occurrence of two words. By combining these two transformations, it may be possible to build a more refined measure of a word's occurrence patterns. We will first add in the

base rate negative information using the GN transformation, and then normalize the matrix using the DOA standardization.

Both approaches integrate the information supplied by negative sampling into a word's semantic representation using analytical transformations of a word-by-word co-occurrence matrix. Importantly, the transformations are parameter-free; they rely only on the co-occurrence values contained in the matrix. The three models, described above, are simple two-parameter models (window size and the subsampling parameter), a contrast to the highly parameterized *word2vec*.

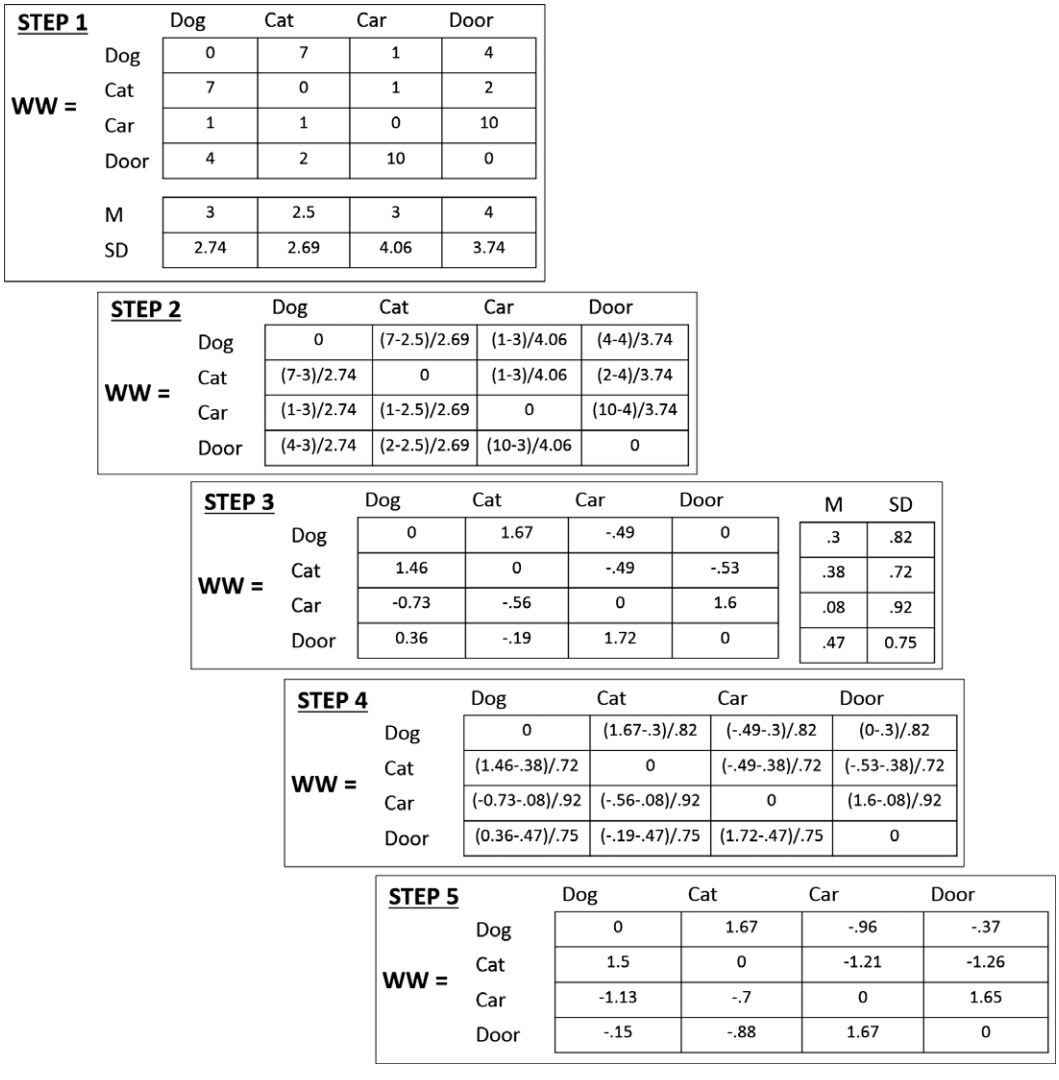

Fig. 7. A numerical example of the DOA transformation.

5. Results

Table 1 shows the results for the seven previously described word relatedness and similarity data sets for the GN, DOA, and combined models, with the results for the positive-only and positive with negative sampling from Fig. 3 for comparison, as well as the fit of the SPPMI metric described above.[4] As shown in Table 1, the GN transformation provides a negligible benefit over the negative sampling model and the SPPMI model. Again, an optimal window size was used in each case. The DOA model, by contrast, provided a substantial benefit. When the two approaches were combined, there was a further benefit, suggesting that the GN and DOA transformations provide complimentary forms of negative information. Put together, they provide a very powerful, yet simple, model of lexical semantics.

To understand the information contained in the different representations, we took the similarity to the combined word pair sets from all seven data sets (n = 5,565) and calculated the inter-correlation of all of the model's similarity values. The results are presented in Table 2.

As shown in Table 2, the most important finding is that the similarity for the negative sampling model and GN model are virtually identical, suggesting that the GN approach is a direct analogue to negative sampling (but with one fewer free parameter). Negative sampling is a mechanism to enhance the magnitude of a positive association above a baseline of occurrence.

So far, we have examined only word relatedness and similarity data. To show that the advantage seen in Table 1 extends to a different semantic task, we applied the models to the classic TOEFL test, first used by Landauer and Dumais (1997). The TOEFL is a synonym test in which a person is presented with a target word and a set of four alternatives. The task is to find alternative closest to the target. The test consists of 80 questions and performance is determined by counting the number of correct synonyms.

Fig. 8 shows performance of all five models on the TOEFL. The performance of the different models replicates the word relatedness and similarity measures, demonstrating that the advantages of the different models extend to a different semantic task. Overall,

Table 1
Fits of the WW model with the five transformations to word relatedness and similarity data

Data	SPPMI	Positive	Neg Samp	GN	DOA	Combined
WordSim-Sim	0.749	0.483	0.762	0.768	0.775	0.811
WordSim-Rel	0.679	0.347	0.653	0.672	0.669	0.696
MTURK-771	0.618	0.394	0.645	0.634	0.658	0.681
RG1965	0.8	0.617	0.737	0.733	0.772	0.791
MEN	0.741	0.544	0.749	0.752	0.768	0.774
Radinsky-2011	0.623	0.488	0.671	0.678	0.684	0.714
SimLex-999	0.266	0.213	0.256	0.286	0.376	0.389
Average	0.639	0.441	0.639	0.646	0.672	0.694

Table 2
Intercorrelations of similarity values for all models

Model	1	2	3	4	5
1. Positive	–				
2. Neg Samp	0.834	–			
3. GN	0.829	0.993	–		
4. DOA	0.773	0.935	0.938	–	
5. Combined	0.72	0.933	0.939	0.986	–

the negative sampling and GN model outperform the positive-only model, the DOA model outperforms the GN model, and the combined model provides the best fit.

To help understand what the GN and DOA transformations do to the co-occurrence values in the word-by-word matrix, Fig. 9 shows histograms of matrix values for the non-transformed, GN, DOA, and combined models. The values in Fig. 9 were attained from 200 randomly selected words and for a model with a vocabulary of 10,000 words. For the non-transformed values, the most common co-occurrence value is zero. All transformations change this distribution into a roughly negatively skewed Gaussian distribution. All three transformations produce large positive tails. The words contained in the tail are those words with which a word has the most unique co-occurrence values.

Importantly, both the GN, DOA, and combined distributions are centered on negative values—indicating that after the transformations most words are not related to most other words. The reasons underlying the shift are slightly different for the two transformations. For the GN transformation, the zero or small co-occurrence values are shifted downwards because the base rate occurrence of a word has been subtracted from the association value. The downward shift is more pronounced for higher frequency words that have few connections to other words—if word *x* occurs frequently with other words, but not with word *y*, the *x*-*y* association is shifted more negatively than when word *x* did not occur with many other words. An example of this can be seen in the DOA transformation applied to the example WW matrix in Fig. 7. Even though the word *dog* had a moderate co-occurrence strength with the word *door*, after the DOA transformation this association was changed to a negative value. This occurred because of the strength of the *car-door* associations caused all other associations to be negatively weighted. A similar shift occurs for zero values with the DOA transformation where they are shifted negatively, as the mean of a column will be positive. The magnitude of the shift depends on the mean and standard deviation of the column. For both transformations, strong associations remain positive because they co-occur at a greater frequency than base rate co-occurrence.

In the DOA transformation both the columns (connection to other words) and rows (a word's lexical semantic entry) are normalized by standardizing their values. The standardization of a row is more important for low–frequency words, as it is likely to have mostly negative values after both transformations. After the row has been standardized, however, many values will be transformed into positive associations, reflecting the distribution of values within the row. Standardization produces the advantage for the combined GN and

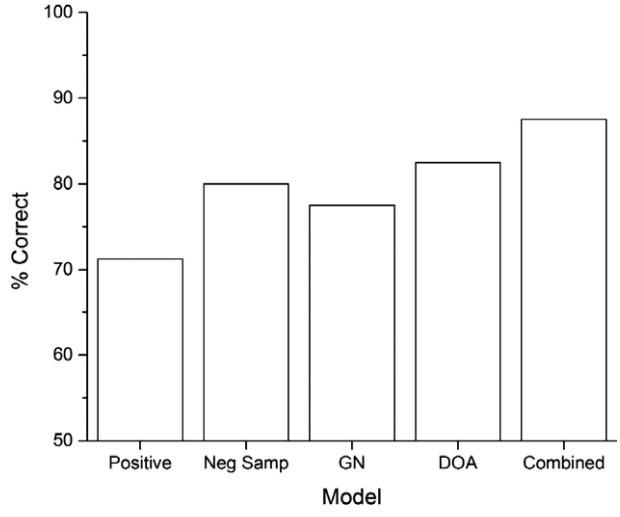

Fig. 8. Results of the word-to-word matrix model across the various transformations on the TOEFL test.

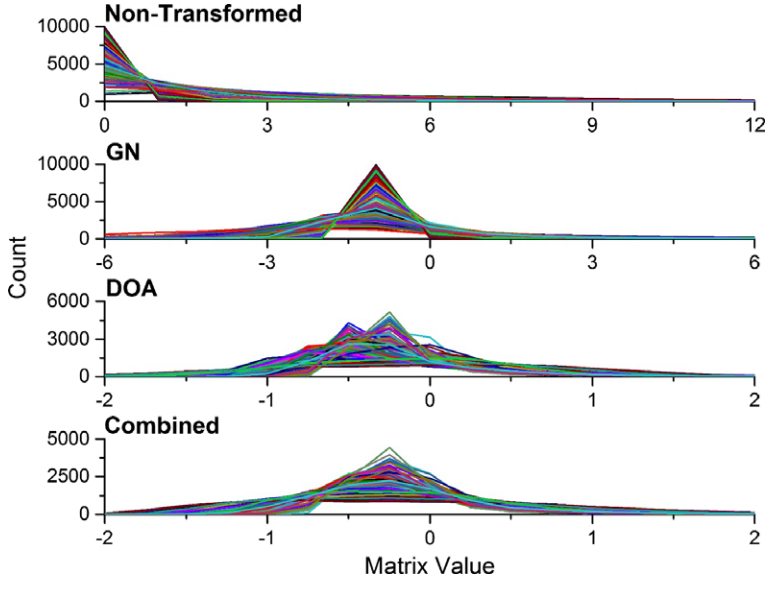

Fig. 9. Histograms of feature values for the non-transformed matrix values and the three different transformations for 200 randomly selected words. Each line represents a different word.

DOA transformation, as the DOA transformation provides a mechanism to smooth out the integration of positive and negative information for an individual word. When a low-frequency word's association values do not exceed the base rate, it does not mean that those associations are unimportant for that word—it simply signals that there were

insufficient samples to exceed the base rate. By normalizing by row, low–frequency words also have positive associations to other words.

5.1. Application to an alternative framework

The foregoing results make it clear that the integration of negative distributional information into a *direct count-based* model's semantic representation provides a substantial increase in the model's predicative power. We kept the underlying representation intentionally simple to explicate the role of negative information. Given that a word-by-word matrix is not currently a mainstream method of constructing semantic representations, it is an open question whether the GN and DOA transformations, described previously, can be applied to an alternative framework. To test the question, the transformations were added to representations derived from the BEAGLE model of semantics (Jones & Mewhort, 2007).

BEAGLE is a vector-accumulation model in which distributed representations are built by learning two types of statistical information: context (the words that co-occur with a given word in language, e.g., *cat-mouse*) and order (the shared temporal roles of words with respect to other words, e.g., both *cat* and *panther* pounce on prey). The information is continuously accumulated in a large, distributed vector as language is encountered.

BEAGLE has been shown to be highly successful at accounting for a wide variety of semantic behaviors, including semantic priming (Hare, Jones, Thomson, Kelly, & McRae, 2009; Jones, Kintsch, & Mewhort, 2006), memory search (Hills, Jones, & Todd, 2012), measuring degradations in semantic memory performance in clinical populations (Johns et al., 2018), the impact of aging on verbal fluency performance (Taler, Johns, & Jones, in press), individual differences in language usage (Johns & Jamieson, 2018), and release from proactive interference in memory (Mewhort, Shabahang, & Franklin, 2018). The standard model uses Gaussian vectors, however the current work will use a sparse representation approximation (see Recchia et al., 2015).[5,6]

The values in BEAGLE vectors are not direct co-occurrence weights but instead represent latent patterns of both context and order learning, unlike a word-by-word representation where each value maps onto the specific association level between two words. As a result, the association level is distributed across patterns contained in the representation. Applying the GN and DOA transformations to this representation, should test whether they can also enhance the representation of models with differing representational assumptions.

We tested BEAGLE with five different transformations: (a) positive-only, (b) negative sampling (using only context information as order would require sampling both words and locations), (c) GN, (d) DOA, and (e) GN-DOA combined. We applied the models to the word similarity data sets and the TOEFL

Table 3 shows that the simulations mimic the previous results using a word-by-word representation. The biggest difference is that the GN model exceeds the fits of the DOA model, although both offer significant improvements over the positive-only model. As

before, the combined model provided by far the best performance. The results in Table 3 demonstrate that negative distributional information can be integrated into different representations with the GN and DOA transformations, allowing for remarkable increases in performance. The BEAGLE model does not achieve the same level of overall performance as the WW model does, but they are close. Obviously, one test does not mean that the transformations can be applied to every representation type (see note 6), but it does demonstrate that it can be applied to some.

5.2. Transformation of similarity

So far, the transformations that have been applied to distributional models has been done at the feature level (either a co-occurrence count in a word-by-word matrix or a strength value in BEAGLE), in which the value in a word's semantic vector was changed to represent how strong that feature is for that word, relative to the representation of the feature value in other word's representations. The GN and DOA transformations demonstrate that by shifting the features of words to reflect how unique a feature value is for a word (compared to the feature value of all other words), large increases in fits to semantic behavior were observed.

There is another level at which the transformations can be applied: word similarity. In standard vector-based models (such as HAL, LSA, BEAGLE, and *word2vec*), similarity is assessed using the vector cosine between two word's vectors, as was done in the foregoing simulations. Given the evidence accumulated so far about the importance of base-rate information in semantic representation, it is fair to ask whether the transformations can also be applied to the similarity between words. Similarity can also be changed to reflect the relative strength of two words, compared to the similarity value that those words share to all other words in the lexicon.

To do so, the entire similarity space of the words in the lexicon must be computed. Then the GN and DOA transformations can be applied to the resulting word-by-word similarity matrix. The GN transformation will directly add in the global similarity levels

Table 3
Fits of BEAGLE models with the five transformations to word relatedness and similarity data

Data	Positive	Neg Samp	GN	DOA	Combined
WordSim-Sim	0.463	0.753	0.741	0.744	0.784
WordSim-Rel	0.389	0.665	0.647	0.58	0.664
MTURK-771	0.383	0.59	0.603	0.575	0.65
RG1965	0.606	0.686	0.718	0.603	0.786
MEN	0.562	0.712	0.726	0.633	0.755
Radinsky-2011	0.482	0.634	0.653	0.707	0.756
SimLex-999	0.186	0.244	0.268	0.295	0.338
Average	0.438	0.612	0.622	0.591	0.676
TOEFL	68.75%	75%	81.25%	70.0%	87.5%

Note. The values for the word similarity data are Pearson correlation coefficients, while TOEFL values are percent correct.

into a word's similarity values, proportional to the overall similarity that a word has to all other words. The DOA transformation will change the similarity value into a z-score reflective of where that similarity value lies in the distribution of all similarity values for those words.

Table 4 shows the results for the word-by-word matrix model. Table 4 also contains the fit from the best *word2vec* model from De Deyne et al. (2016). Table 5 shows the corresponding results for the BEAGLE simulation. They compared their model fits to the vectors used in alternative studies using *word2vec*, including the best model from Mandera et al. (2017), and alternative neural embedding models (e.g., GloVe; Pennington, Socher, & Manning, 2014), and found that their fits were equivalent to other modeling efforts, and so serve as a strong comparison for the model performance reported here.

The first important trend in the results is that the GN, DOA, and combined transformations applied to the positive-only model's similarity space yielded results roughly equivalent fits to previous findings in which the transformations had been applied to the feature values within a word's representation. The positive-only model in this refers to the WW matrix that has not had its values transformed. This table shows that relative weighting is important not just for semantics representation but also at the similarity level. This important finding indicates that it is the relative weighting of a word's similarity or semantic features to other words that matters and not necessarily the direct co-occurrence between those two words.

The combined transformation was the best fitting model for both representation types applied to the similarity values from the transformed representations. The result confirms the importance of relative weighting in distributional semantics; the similarity between words should reflect how similar the two are in the distribution of all similarity values. The transformations hold at both the feature and similarity level, at least for these two models. Additionally, note that performance of the word-by-word matrix model was roughly equivalent to the *word2vec* model used in De Deyne et al. (2016), even though it was much simpler than a neural embedding model, demonstrating the power of this approach.

Table 4
The fits of WW with the GN, DOA, and combined transformations applied to the similarity matrix for both the positive-only and transformed representation

	Positive-Only				Transformed				
Data	None	GN	DOA	Combined	None	GN	DOA	Combined	DDPN
WordSim-Sim	0.347	0.721	0.727	0.739	0.696	0.756	0.74	0.753	0.7
WordSim-Rel	0.483	0.768	0.776	0.791	0.811	0.812	0.8	0.822	0.79
MTURK-771	0.394	0.623	0.609	0.633	0.681	0.661	0.675	0.683	0.71
RG1965	0.617	0.744	0.768	0.786	0.791	0.809	0.825	0.854	0.83
MEN	0.544	0.756	0.754	0.776	0.774	0.789	0.765	0.8	0.85
Radinsky-2011	0.488	0.725	0.728	0.711	0.714	0.721	0.744	0.744	0.711
SimLex-999	0.213	0.288	0.331	0.347	0.389	0.391	0.395	0.4	0.43
Average	0.441	0.667	0.67	0.683	0.694	0.705	0.706	0.722	0.728

Note. DDPN refers to the fits of the best *word2vec* model from De Deyne et al. (2016).

Table 5
The fits of BEAGLE with the GN, DOA, and combined transformations applied to the similarity matrix for both the positive-only and transformed representation

	Positive-Only				Transformed			
Data	None	GN	DOA	Combined	None	GN	DOA	Combined
WordSim-Sim	0.463	0.766	0.764	0.77	0.784	0.791	0.793	0.826
WordSim-Rel	0.389	0.717	0.718	0.727	0.664	0.716	0.723	0.729
MTURK-771	0.383	0.625	0.624	0.627	0.65	0.66	0.652	0.664
RG1965	0.606	0.713	0.726	0.739	0.786	0.798	0.803	0.825
MEN	0.562	0.765	0.756	0.757	0.755	0.774	0.789	0.802
Radinsky-2011	0.482	0.721	0.714	0.724	0.756	0.744	0.743	0.735
SimLex-999	0.186	0.277	0.283	0.285	0.338	0.341	0.358	0.364
Average	0.438	0.655	0.655	0.661	0.676	0.689	0.694	0.706

5.3. Simplifying the framework

The word-by-word matrix model used here, has three parameters: (a) the subsampling parameter, (b) window size, and (c) vocabulary size. For the BEAGLE model, there are two additional parameters (vector size and vector sparsity), although these are fixed hyperparameters that have a relatively small impact on model performance (Jones & Mewhort, 2007; Recchia et al., 2015). The vocabulary size parameter is a common parameter for all distributional models and one that is not easily removed. The subsampling and window size could both be removed with minimal impact on performance. The window size parameter can be removed by using a weighted window (similar to the operation of the skipgram implementation in *word2vec*). As Levy et al. (2015) specify, words further apart in a sentence are given a lesser learning weight using a harmonic function:

$$\Delta M_{w_i,w_j} = \frac{1}{|i-j|} \tag{9}$$

where i and j are locations within a sentence, and w_i and w_j are the corresponding words in those locations.

Although this weighting scheme can be used within a window, it also provides a mechanism by which the window parameter can be removed, by applying the weights to all words within a sentence. Words that are far away within a sentence will have a minimal increase in strength.

Using the GN and DOA transformations may make the subsampling parameter unnecessary, as the base rate occurrence of very high frequency would be high for every word, resulting in these words having a negligible impact on model performance once the normalization procedures have been applied. That is, adding very high-frequency words into a model's representation will produce a constant increase in similarity for all words, but the normalization procedures neutralizes this artifact.

To test the impact of having a weighted window and removing the subsampling parameter, four simulations were conducted crossing the two parameters: (a) with and without a weighted window, and (b) with and without subsampling. We tested both the word-by-word matrix model and the BEAGLE model. When there is no weighted window, all words in the sentence will be updated with a weighted window across the entire sentence (i.e., there is no set window size).

Table 6 shows the results for the word-by-word matrix model; Table 7 shows the corresponding results the BEAGLE. In both cases, removing subsampling had a minimal impact on performance. A weighted window had a positive impact for the WW model, but a negative impact on the BEAGLE model, demonstrating, again, that some transformations are representation dependent. Thus, although performance drops slightly from having an optimized window size (as would be expected), it is possible to reduce the model to a simple one-parameter (vocabulary size) model, while still retaining excellent fits to lexical semantic behaviors (0.722 average correlation with the optimized word-by-word matrix model vs. 0.704 for the simplified word-by-word matrix model; 0.706 average correlation for the optimized BEAGLE model vs. 0.683 for the simplified BEAGLE model.

As in all formal models, adding a parameter allows greater flexibility in accounting for data. However, there needs to be a consideration for parsimony. The simulations contained in Tables 6 and 7 demonstrate that with the GN and DOA, it is possible to construct very simple but powerful models of lexical semantics which offer comparable performance to much more complex neural embedding models.

5.4. Discussion

This section tested two analytical matrix transformation techniques designed to integrate negative information into a word's semantic representation, but without having to use a free parameter or a sampling methodology. The transformations were designed with both parsimony and cognitive plausibility in mind. The first transformation, the global negative technique, adds in negative global co-occurrence rates directly proportional to

Table 6
Effects of using a weighted window and removing the use of subsampling on the WW model.

	No Weighted Window		Weighted Window	
Data	No Subsampling	Subsampling	No Subsampling	Subsampling
WordSim-Sim	0.784	0.801	0.806	0.811
WordSim-Rel	0.764	0.763	0.742	0.753
MTURK-771	0.638	0.642	0.684	0.678
RG1965	0.803	0.812	0.821	0.854
MEN	0.781	0.782	0.793	0.801
Radinsky-2011	0.694	0.684	0.698	0.697
SimLex-999	0.293	0.313	0.382	0.365
Average	0.68	0.685	0.704	0.708

Table 7
Effects of using a weighted window and removing the use of subsampling on the BEAGLE model

	No Weighted Window		Weighted Window	
Data	No Subsampling	Subsampling	No Subsampling	Subsampling
WordSim-Sim	0.792	0.806	0.781	0.81
WordSim-Rel	0.717	0.701	0.654	0.692
MTURK-771	0.656	0.654	0.664	0.661
RG1965	0.802	0.828	0.77	0.804
MEN	0.794	0.796	0.769	0.792
Radinsky-2011	0.688	0.697	0.664	0.699
SimLex-999	0.334	0.334	0.362	0.351
Average	0.683	0.688	0.666	0.687

the positive associations that a word has received; it turned out to be a direct analog to negative sampling.

The second technique, which we referred to as the distribution of associations (DOA) transformation, infers the strength of the association between two words by taking into account the co-occurrence patterns of all other words in the matrix. The result is a matrix of standardized associations, signaling how unique the co-occurrence of two words are, over and above all other word co-occurrences. By combining these two techniques, better fits to word pair similarity and synonym tests were attained. Additionally, it was shown that these transformations could be applied to an alternative framework, the BEAGLE model of semantics, and that the transformations can be also applied at the word similarity level. Finally, we demonstrated that the transformations are powerful enough that the two main free parameters, window size and the subsampling parameter, can be removed with only a small drop in performance for the models.

6. General discussion

The goal of this article was to evaluate the role of negative sampling in training distributional models of semantics. The simulations contained in this article show that negative sampling serves to integrate an alternative type of distributional information into a word's semantic representation. Specifically, we showed that negative sampling includes base rate occurrence in a word's representation. The model used to explore this effect was based on a word-by-word matrix model, and when negative sampling is used in this framework the resulting values in the matrix represent positive co-occurrence over a negatively sampled base rate. It was further shown that this process may be integrated into a word's representation using multiple parameter-free analytical techniques, resulting in both a more parsimonious and more powerful model.

The results demonstrate that negative information plays an important role in distributional semantics. The GN and DOA transformations calculate a semantic feature's uniqueness to that individual word, when compared against the feature value for other words. The

transformations do so in slightly different ways, and, when combined, they provide superior results. Given that the transformations work at both the semantic feature and word similarity levels, relative comparison of word properties is a general component of lexical semantic memory.

The GN and DOA transformations were also successfully applied to multiple distributional models. Given the diversity of types of distributional models, it is not necessarily the case that these techniques will work on all representation types. However, the underlying idea of the operations can be used where the strength of other word's values is used to determine the relative strength of a word's feature or similarity. The proper transformation may depend on the mathematical properties of the underlying representation of a model, an important topic for future research.

The goal of this article was not to propose a new type of distributional model. Indeed, as described, the DOA transformation has much in common with past proposals, such as shifted PMI. Rather, the goal was to understand the interaction between positive and negative information in forming semantic representations of words. Through the simulations in this article we have demonstrated that negative information plays an important role in building distributed semantic representations, and we have conceptualized its impact. Our hope is that this greater level of understanding about the role of negative information in semantic behavior leads to better, simpler, and conceptually clear models of semantic behavior, with the GN and DOA transformations initial guidelines as to how this can be done.

The trajectory of this research serves as a reminder of the importance of understanding the theoretical basis of the computational models that are used in the cognitive sciences. As Mandera et al. (2017) note, the fits provided by neural embedding models are undeniably impressive, but these models have a much greater level of parameterization than standard models (Levy et al., 2015; Asr & Jones, 2017). We should not be surprised that extra parameters provide significant power to the model. The number of parameters matters relatively little in applied research where the models are engineering solutions designed to solve a specific problem. In contrast, the number of free parameters is an important consideration in cognitive modeling because the parameters often serve important functions in explaining human behavior. Further, methods of model comparison penalize models that have a large number of parameters (e.g., Akaike, 1974; Schwarz, 1978).

To continue developing cognitive theory, distributional models will need a better grasp of model complexity. With certain semantic datatypes (e.g., verbal fluency; Hills et al., 2012; Johns et al., 2018; Taler, Johns, Young, Sheppard, & Jones, 2013) it is possible to use standard model comparison techniques like AIC (Akaike, 1974) to compare the complexity of process models using distributional semantic representations, but this still does not get to the problem of representational complexity (for a more in depth discussion of this problem, see Jones, Hills, & Todd, 2015). Outside of number of parameters, there is also complexity in training materials. Some materials are more informative depending on task (e.g., Johns et al., 2019), while some models benefit from increased amount of training materials (e.g, Recchia & Jones, 2009). Quantifying model complexity is an important challenge in constructing cognitively plausible models of semantic memory, and one that the field needs to devote attention to.

We do not intend to imply that *word2vec* is a poor model; indeed, as previous research has shown, it is a very powerful model. But its success does not appear to reflect the use of a connectionist architecture or of active prediction. Instead, its success reflects how it interprets linguistic information.

Subsampling of words and negative sampling are clearly important components of distributional modeling, as previous research and this article have shown convincingly. The role of negative sampling can be readily accounted for with simple transformations of semantic vector representations, either at the feature or similarity level. The power of the transformation methods described in this paper is not from their mathematical sophistication, but instead the ideas that underlie them. They offer an insight into how semantic representations are organized and compared in the human mind. The fact that they offer improved fits to data is tangential—it is the understanding that they provide that is important.

A coherent science requires coherent theory. Even though a field like distributional semantics and language comprehension lends itself to advanced machine learning techniques, it does not mean that we should not try to understand human cognition. Instead, as more advanced techniques are being used in the cognitive sciences, it will become even more necessary to understand the reason why different computational techniques are successful in accounting for human behavior.

As computational cognitive science moves forward and machine learning begins to play a larger role both in theoretical and empirical pursuits, it will still be necessary to ground results in cognitive theory. The classic latent semantic analysis model (Landauer & Dumais, 1997) is a case study in how to do so successfully. Although Landauer and Dumais introduced advanced computational techniques to the field, they also put serious effort into attaching the techniques to theories of cognition. The results of this article demonstrate the continued importance of this practice.

Notes

1. The *word2vec* architecture has two possible model directions: The context may be used to predict the word (referred to as a CBOW), or the word may be used to predict the context (a skipgram). The theoretical claims in this paper apply to either training scheme, but we will focus on the skipgram in all examples as it has been the most prominently used in the literature.
2. A value of 6×10^{-6} for t was found to perform best for this modeling framework. This parameter was held constant across all simulations. This may lead to poorer performance depending on corpus size and construction, but for the sake of simplicity, the parameter was not manipulated.
3. Here the Finkelstein et al. (2001) was split into a similarity and relatedness dataset, following the suggestions of Agirre et al. (2009). This split is common practice in analyzing distributional models (e.g., De Deyne et al., 2016).
4. The performance of the SSPMI model was optimal at k=5.

5. Similar to the work in Recchia et al. (2015), environmental vectors will have a dimensionality of 20,000 with 6 non-zero values.
6. The transformations described here work much better with the sparse implementation of the model, rather than the Gaussian implementation. This is due to feature values in these vectors already having a Gaussian distribution post-training, and so the transformations have a relatively small impact on the resulting representations. Different representation schemes will likely need unique transformations in order to integrate negative information optimally.

References

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., & Soroa, A. (2009). A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 19–27). Stroudsburg, PA: Association for Computational Linguistics.

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

Asr, F. T., & Jones, M. N. (2017). An artificial language evaluation of distributional semantic models. In *Proceedings of the ACL Conference on Natural Language Learning (CoNLL)* (pp. 134–142). Stroudsburg, PA: Association for Computational Linguistics.

Asr, F. T., Willits, J. A., & Jones, M. N. (2016). Comparing predictive and co-occurrence based models of lexical semantics trained on child-directed speech. In A. Papafragou, D. Grodner, D. Mirman & J. C. Trueswell (Eds.), *Proceedings of the 37th Meeting of the Cognitive Science Society* (pp. 1092–1097). Austin, TX: Cognitive Science Society

Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics-Volume 1* (pp. 238–247). Stroudsburg, PA: Association for Computational Linguistics.

Bruni, E., Boleda, G., Baroni, M., & Tran, N. K. (2012). Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1* (pp. 136–145). Stroudsburg, PA: Association for Computational Linguistics.

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

Bullinaria, J. A., & Levy, J. P. (2012). Extracting semantic representations from word co-occurrence statistics: Stop-lists, stemming, and SVD. *Behavior Research Methods*, *44*, 890–907.

Chang, Y. N., Furber, S., & Welbourne, S. (2012). Generating realistic semantic codes for use in neural network models. In N. Miyake, D. Peebles & R. P. Cooper (Eds.), *Proceedings of the Annual Meeting of the Cognitive Science Society* (pp. 198–203). Austin, TX: Cognitive Science Society.

Church, K. W., & Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, *16*, 22–29.

De Deyne, S., Perfors, A., & Navarro, D. J. (2016). Predicting human similarity judgments with distributional models: The value of word associations. In *COLING* (pp. 1861–1870). Stroudsburg, PA: Association for Computational Linguistics.

Demski, A., Ustun, V., Rosenbloom, P. S., & Kommers, C. (2014). Outperforming word2vec on analogy tasks with random projections. *arXiv preprint arXiv*:1412.6616.

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*, 116–131.

Goldberg, Y., & Levy, O. (2014). word2vec Explained: Deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:*1402.3722.

Gries, S. T. (2013). 50-something years of work on collocations. *International Journal of Corpus Linguistics*, *18*, 137–166.

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

Halawi, G., Dror, G., Gabrilovich, E., & Koren, Y. (2012). Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1406–1414). New York: ACM.

Hare, M., Jones, M., Thomson, C., Kelly, S., & McRae, K. (2009). Activating event knowledge. *Cognition*, *111*, 151–167.

Harris, Z. S. (1954). Distributional structure. *Word*, *10*, 146–162.

Hill, F., Reichart, R., & Korhonen, A. (2016). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, *4*, 665–695.

Hills, T. T., Jones, M. N., & Todd, P. M. (2012). Optimal foraging in semantic memory. *Psychological Review*, *119*, 431–438.

Hills, T. T., Maouene, J., Riordan, B., & Smith, L. (2010). The associative structure of language and contextual diversity in early language acquisition. *Journal of Memory and Language*, *63*, 259–273.

Jamieson, R. K., Johns, B. T., Avery, J. E., & Jones, M. N. (2018). An instance theory of semantic memory. *Computational Brain & Behavior*. *1*, 119–136.

Johns, B. T., & Jamieson, R. K. (2018). A large-scale analysis of variance in written language. *Cognitive Science*, *42*, 1360–1374.

Johns, B. T., & Jones, M. N. (2015). Generating structure from experience: A retrieval-based model of language processing. *Canadian Journal of Experimental Psychology*, *69*, 233–251.

Johns, B. T., Jones, M. N., & Mewhort, D. J. K. (2016). Experience as a free parameter in the cognitive modeling of language. In A. Papafragou, D. Grodner, D. Mirman & J. C. Trueswell (Eds.), *Proceedings of the 37th Annual Conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society.

Johns, B. T., Jones, M. N., & Mewhort, D. J. K. (2019). Using experiential optimization to build lexical representations. *Psychonomic Bulletin & Review*. *26*, 103–126

Johns, B. T., Taler, V., Pisoni, D. B., Farlow, M. R., Hake, A. M., Kareken, D. A., Unverzagt, F. W., & Jones, M. N. (2018). Cognitive modeling as an interface between brain and behavior: Measuring the semantic decline in mild cognitive impairment. *Canadian Journal of Experimental Psychology*, *72*, 117–126.

Jones, M. N., Hills, T. T., & Todd, P. M. (2015). Hidden processes in structural representations: A reply to Abbott, Austerweil, & Griffiths. *Psychological Review*, *122*, 570–574.

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

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

Kwantes, P. J. (2005). Using context to build semantics. *Psychonomic Bulletin & Review*, *12*, 703–710.

Landauer, T. K., & Dumais, S. T. (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.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, *25*, 259–284.

Levy, O., & Goldberg, Y. (2014a). Dependency-based word embeddings. In R. Morante & S. W. Yih (Eds.), *Proceedings of ACL* (pp. 171–180). Stroudsburg, PA: Association for Computational Linguistics.

Levy, O., & Goldberg, Y. (2014b). Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 27* (pp. 2177–2185). Cambridge, MA: MIT Press.

Levy, O., Goldberg, Y., & Dagan, I. (2015). Improving distributional similarity with lessons learned from word embedding. *Transactions of the Association for Computational Linguistics*, *3*, 211–225.

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

Mandera, P., Keuleers, E., & Brysbaert, M. (2017). Explaining human performance in psycholinguistic tasks with models of semantic similarity based on prediction and counting: A review and empirical validation. *Journal of Memory and Language*, *92*, 57–78.

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. San Francisco: W. H. Freeman.

Mewhort, D. J. K., Shabahang, K. D., & Franklin, D. R. J. (2018). Release from PI: An analysis and a model. *Psychonomic Bulletin & Review*, *25*, 932–950.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv Preprint* arXiv:1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing Systems* (pp. 3111–3119). Cambridge, MA: MIT Press.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. (1532–1543). Stroudsburg, PA: Association for Computational Linguistics.

Radinsky, K., Agichtein, E., Gabrilovich, E., & Markovitch, S. (2011). A word at a time: Computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th International Conference on World Wide Web* (pp. 337–346). New York: ACM.

Ramscar, M., Hendrix, P., Shaoul, C., Milin, P., & Baayen, H. (2014). The myth of cognitive decline: Non-linear dynamics of lifelong learning. *Topics in Cognitive Science*, *6*, 5–42.

Ramscar, M., Sun, C. C., Hendrix, P., & Baayen, H. (2017). The Mismeasurement of mind: Life-span changes in paired-associate-learning scores reflect the "cost" of learning, not cognitive decline. *Psychological Science*, *28*, 1171–1179.

Recchia, G. L., & Jones, M. N. (2009). More data trumps smarter algorithms: Comparing pointwise mutual information to latent semantic analysis. *Behavior Research Methods*, *41*, 657–663.

Recchia, G., & Nulty, P. (2017). Improving a fundamental measure of lexical association. In G. Gunzelmann, A. Howes, T. Tenbrink, & E. Davelaar (Eds.), *Proceedings of the 39th Annual Conference of the Cognitive Science Society* (pp. 2963–2968). Austin, TX: Cognitive Science Society.

Recchia, G., Sahlgren, M., Kanerva, P., & Jones, M. N. (2015). Encoding sequential information in semantic space models: Comparing holographic reduced representation and random permutation. *Computational Intelligence and Neuroscience*, *58*.

Rohde, D. L., Gonnerman, L. M., & Plaut, D. C. (2006). An improved model of semantic similarity based on lexical co-occurrence. *Communications of the ACM*, *8*, 627–633.

Rubenstein, H., & Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, *8*, 627–633.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by backpropagating errors. *Nature*, *323*, 533–536.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*, 461–464.

Taler, V., Johns, B. T., & Jones, M. N. (in press). A large scale semantic analysis of verbal fluency across the aging spectrum: Data from the Canadian longitudinal study on aging. *Journal of Gerontology: Psychological Sciences*.

Taler, V., Johns, B. T., Young, K., Sheppard, C., & Jones, M. N. (2013). A computational analysis of semantic structure in bilingual fluency. *Journal of Memory and Language*, *69*, 607–618.

Wittgenstein, L. (1953). *Philosophical investigations*. Hoboken, NJ: John Wiley & Sons.