Psychological Review
2007, Vol. 114, No. 1, 1–37


0033-295X/07/$12.00 DOI: 10.1037/0033-295X.114.1.1

# Representing Word Meaning and Order Information in a Composite Holographic Lexicon

Michael N. Jones
University of Colorado at Boulder

Douglas J. K. Mewhort
Queen's University

The authors present a computational model that builds a holographic lexicon representing both word meaning and word order from unsupervised experience with natural language. The model uses simple convolution and superposition mechanisms (cf. B. B. Murdock, 1982) to learn distributed holographic representations for words. The structure of the resulting lexicon can account for empirical data from classic experiments studying semantic typicality, categorization, priming, and semantic constraint in sentence completions. Furthermore, order information can be retrieved from the holographic representations, allowing the model to account for limited word transitions without the need for built-in transition rules. The model demonstrates that a broad range of psychological data can be accounted for directly from the structure of lexical representations learned in this way, without the need for complexity to be built into either the processing mechanisms or the representations. The holographic representations are an appropriate knowledge representation to be used by higher order models of language comprehension, relieving the complexity required at the higher level.

*Keywords:* semantic memory, mental lexicon, latent semantic analysis, statistical learning, holographic models

Language is an immensely complex behavior. At minimum, it requires knowledge of the words of that language, typically thought to be stored in a *mental lexicon,* and knowledge of the grammatical application of those words in sentences. Higher order models of language comprehension (e.g., Kintsch, 1988, 1998, 2001) require a realistic representation of word meaning and order information to be successful, and it would be particularly appealing if these representations could be learned from the statistical redundancies present in natural language.

The mental lexicon has a rich history in psychology. Traditionally, the lexicon has been viewed as a dictionary database, each entry containing a word's meaning and, in some cases, even its syntactic rules and phonological characteristics (see Elman, 2004, and Pustejovsky, 1996, for reviews). The precise representation of this information, however, has been greatly debated. This article presents a model based on signal processing and associative memory theory that builds distributed representations for words containing both semantic and order information from unsupervised learning of natural language. Word meaning and order information are stored together as a single pattern of elements in a distributed holographic lexicon. A word's representation gradually stabilizes from statistical sampling across experience. Furthermore, transition information can be retrieved from the lexicon using holographic decoding and resonance.

In the 1950s, Charles Osgood and George Miller worked independently on seemingly unrelated problems. Osgood (1952, e.g.) tackled the problem of representing word meaning. It appeared at the outset that representing word meaning would always be abstract and unobtainable, and the best approach possible was to approximate semantic relationships with subjective ratings. Miller (1951, e.g.), on the other hand, was interested in the problem of sequential dependencies between words. Because sequential dependencies involve direct statistical transitions between words, it appeared at the outset that a comprehensive model of sequential dependencies was within reach, whereas a comprehensive model of abstract semantics was likely unobtainable. The sequential dependency problem, however, turned out to be considerably more complicated than was originally anticipated.

By the 1990s, there were several good models that could learn semantic representations for words automatically from experience with text (e.g., Landauer & Dumais, 1997; Lund & Burgess, 1996). However, these models neglect information inherent in word transitions. Theoretical work on word meaning has largely been developed independently from theoretical work on word transitions, but a complete model of the lexicon needs to incorporate both sources of information to account for the broad range of semantic categorization, typicality, and priming data, as well as data relating to lexical class, word usage, and word transitions in sentences.

Michael N. Jones, Institute of Cognitive Science, University of Colorado at Boulder; Douglas J. K. Mewhort, Department of Psychology, Queen's University, Kingston, Ontario, Canada.

This article is based in part on a doctoral dissertation submitted to Queen's University by Michael N. Jones. The research was supported by grants to Douglas J. K. Mewhort from the Natural Sciences and Engineering Council of Canada (NSERC) and Sun Microsystems. Michael N. Jones was supported by both a graduate scholarship and a postdoctoral fellowship from NSERC. Simulations were conducted on the supercomputers at the High Performance Computing Virtual Laboratory. We are grateful to Walter Kintsch, Tom Landauer, and Ben Murdock for comments on a draft of the article.

Correspondence concerning this article should be addressed to Michael N. Jones, who is now at the Department of Psychological and Brain Sciences, Indiana University, Bloomington, IN 47405. E-mail: jonesmn@indiana.edu

## Representing Word Meaning

Pioneering work on the representation of knowledge focused on feature lists and semantic networks. Feature lists represent words using lists of descriptive binary features (e.g., E. E. Smith, Shoben, & Rips, 1974). For example, birds have wings, and dogs do not. The feature lists are typically hand coded by the theorist on the basis of intuition. Some recent models of word recognition bypass the problem of determining what semantic features to use by employing random binary vectors (e.g., Masson, 1995; Plaut, 1995), whereas others build the lists from empirical feature reports generated by subjects (e.g., McRae, de Sa, & Seidenberg, 1997). However, all feature list models share the common theme that the semantic representation for a word is a list of binary features that describe it.

By contrast, semantic network theories (e.g., Collins & Quillian, 1972; Collins & Loftus, 1975) assume that words are represented by localist nodes of interconnected concepts. Words that are connected to one another with many (or with direct) pathways are more similar in meaning. More recent semantic networks learn new concepts and connections unsupervised and have large-scale learning structure that parallels human learning in many ways (Steyvers & Tenenbaum, 2005).

A third method of semantic representation has built on Osgood's (1941, 1952, 1971) multidimensional representation scheme (see also Salton, 1973; Salton, Wong, & Yang, 1975). Modern hyperspace models build semantic representations directly from statistical co-occurrences of words in text, typically representing them in a high-dimensional semantic space (e.g., Landauer & Dumais, 1997; Lund & Burgess, 1996). The semantic space approach minimizes representation and processing assumptions because much of the model's complexity is learned from the environment, not hardwired into the model itself.

Semantic space models differ from semantic networks in that they generate distributed representations for words rather than the localist representations used by semantic nets. Semantic space models and feature lists are both distributed word representations; a principled difference between a semantic space model and a feature list model is the nature of word features. In a semantic space model, the features that represent a word are abstract values that have no identifiable meaning in isolation from the other features. Although a particular feature of *bird* in a feature list might be "has wings," the presence of which has birdlike meaning on its own, the meaning of *bird* in a semantic space model is the aggregate distributed pattern of all the abstract dimensions, none of which has interpretable meaning on its own.

Latent semantic analysis (LSA; Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990; Landauer & Dumais, 1997) has received the most attention of all the semantic space models. LSA begins by computing a Word × Document frequency matrix from a large corpus of text, using about 90,000 words and about 37,000 documents; obviously, this matrix is quite sparse. The raw entries represent the frequency with which each word appears in a particular document. The entries are then converted to log-frequency values and are divided by the word's entropy, $-\Sigma\, p \log p$, over all its documents. Next, the dimensionality of the Word × Document matrix is optimized using singular value decomposition (SVD) so that each word is represented by a vector of approximately 300 dimensions; however, the dimensions have no particular meaning or direct correspondence to the text. SVD has the effect of bringing out latent semantic relationships among words even if they have never co-occurred in the same document. The basic premise in LSA is that the aggregate contexts in which a word does and does not appear provide a set of mutual constraints to induce the word's meaning (Landauer, Foltz, & Laham, 1998), or, as Firth (1957) has put it, "you shall know a word by the company it keeps" (p. 11).

LSA has been very successful at simulating a wide range of psycholinguistic phenomena, from judgments of semantic similarity (Landauer & Dumais, 1997) to word categorization (Laham, 2000), discourse comprehension (Kintsch, 1998), and judgments of essay quality (Landauer, Laham, Rehder, & Schreiner, 1997). LSA has even earned college entrance-level grades on the Test of English as a Foreign Language (TOEFL) and has been shown to acquire vocabulary at a rate that is comparable to standard developmental trends (Landauer & Dumais, 1997).

In spite of their successes, co-occurrence models have been criticized on a number of grounds. Most importantly, LSA is often criticized as a bag-of-words model in that it ignores the statistical information inherent in word transitions within documents (e.g., Perfetti, 1998; Serafin & Di Eugenio, 2004; Wiemer-Hastings, 2000, 2001). A word's meaning is defined not only by its context but also by its temporal position in that context relative to other words. Perfetti (1998) has noted that the "lack of syntax is an architectural failure [of LSA]" (p. 367). Although LSA can learn much about document meanings without the benefit of word order (Landauer et al., 1997), order information is certainly necessary for a comprehensive model of word meaning and usage.

## Constraints From Word Order

Solving the bag-of-words problem is the next major step in the development of semantic space models. As well as providing additional information about meaning, transition information defines a word's lexical class and grammatical behavior. So, to amend Firth's (1957) comment, you shall know a word by both the company it keeps and how it keeps it.

Many models of sequential dependencies in language are based on Miller's (1951) notion of an *n*-gram (see also Miller & Selfridge, 1950).[1] A classic *n*-gram model records the frequency of occurrence of every possible word sequence chunk that it encounters in a textbase (there is usually a window of about seven words around the target that are considered *n*-grams). To predict a word in a new sentence position, the model looks up the frequency with which the target word has been encountered as a bigram, trigram, and so on during training. To be useful, *n*-gram models usually need to be trained on massive amounts of text and require extensive storage space for relatively little information.

More recently, the focus has been to identify and simulate the generative rules of word transitions (inspired by the work of Chomsky, 1965, 1980). Probabilistic methods (e.g., hidden Markov models) infer the generative rules that could produce a text corpus by observing transition statistics (for a modern example, see Solan, Horn, Ruppin, & Edelman, 2005). Similarly, sto-

[1] To be clear, *n*-gram in this article refers specifically to sequence chunks of *n* words encountered during learning (the term is also often used to refer to sequences of letters within a word).

chastic context-free grammars (Booth, 1969) learn probabilities for production rules from training data (e.g., Jelinek & Lafferty, 1991). Such models typically require supervised training on tagged corpora (i.e., the part of speech or syntactic role of a word in a sentence is hand coded by a human rater; e.g., Marcus, Santorini, & Marcinkiewicz, 1993; Palmer, Gildea, & Kingsbury, 2005).

Where there are open questions regarding the exact nature of representation for a word's meaning, there is consensus that the representation for a word's grammatical usage is in the form of rules or production systems. It follows that knowledge of word meaning and knowledge of word usage are represented in different forms and are stored separately from one another.

In the early 1990s, simple recurrent networks (SRNs; Elman, 1990, 1991, 1993, 1995; see also Servan-Schreiber, Cleeremans, & McClelland, 1991) became promising at learning sequential dependency information from artificial languages. An SRN is a connectionist network that learns to predict temporal sequences with a recurrent context layer that represents the network's previous initial state as the hidden layer is given input for the present state. Because the hidden layer's state is affected by the hidden layer state from the previous time step, which is in turn affected by the state before it, SRNs can retain sequential dependency information over several time steps (Elman, 2004).

In an SRN, the representation of a word's meaning is the pattern of activation across the hidden layer. However, this same information is used when predicting transitions using the word. Thus, SRNs afford the possibility that word meaning and word usage are the same type of knowledge and may be stored together.

SRNs can predict the next word in a sequence (Elman, 1990), can identify embedded structure (Elman, 1991) and recursion (Christiansen & Chater, 1999), and can track limited long-range dependencies (Elman, 1995). Unfortunately, SRNs have questionable scaling properties. Although they function quite well on small finite-state grammars, to date, none has scaled up to unconstrained natural language (but see Burgess & Lund, 2000, and Howard & Kahana, 2002, for similar large-scale models). In addition, SRNs have difficulty retaining information over many time steps, and it is questionable whether feedback is even necessary in an SRN (Plate, 2003). Nonetheless, if SRNs can be scaled up to real-word language problems, they provide a very appealing architecture.

With a random distributed representation for words, a simple composite memory system can learn that particular words have appeared together in a sentence or document with a composite vector, formed by superposition (e.g., adding or averaging the vectors for words in the sentence or document). Although a lexicon built with one composite vector per word can theoretically learn co-occurrence information, it cannot learn sequential information. In a composite vector, one can no longer determine which features (elements) go with which objects (original vectors)—this is referred to as the *binding problem*. Furthermore, there is no way to determine the original ordering of the vectors from a composite representation. The next section presents a binding operation, adapted from signal processing and associative memory theory, that preserves order and is a potential solution to the binding problem.

A comprehensive model of the lexicon requires information about both word meaning and word order, as well as an account of how the sources of information are learned, represented, and retrieved from memory. Merging a semantic space model with a sequential dependency model is not a trivial task. However, precedents exist for fusing the two types of information in other domains of memory theory.

Assuming that language is similar to other types of associative stimuli, is it possible, as small-scale work with SRNs has suggested, that knowledge about word transition is represented and stored in the same manner as knowledge about word meaning? If so, how much of the human data can be accounted for directly from the structure of knowledge represented in this manner? Operationalizing language as merely a complex type of sequential dependency stimulus allows ideas to be borrowed from well-established theory in associative memory.

## Lessons From Associative Memory Theory

To model paired-associate learning, Murdock (1982, 1992, 1993) has successfully used *convolution* as a mechanism to form associations between pairs of random vectors representing words or objects. Convolution is basically a method of compressing the outer-product matrix of two vectors and avoids the binding problem. *Correlation* (also called *deconvolution*) is an approximate inverse that can decode convolution from the association back into either of the parent representations when presented with one parent representation in the environment.

For example, to associate item vectors $\mathbf{x}$ and $\mathbf{y}$, one could convolve them into a third vector, $\mathbf{z}$, for storage: $\mathbf{z} = \mathbf{x} * \mathbf{y}$. The association vector may contain the superposition of several pairwise associations with minimal interference. At retrieval, the memory vector $\mathbf{z}$ is probed with one item of a pair, and the other item can be successfully reconstructed (in a noisy form) with correlation: $\mathbf{y} \# \mathbf{z} \approx \mathbf{x}$. Convolution is associative and commutative, and it distributes over addition. Such convolution–correlation memory models are often referred to as *holographic* models because they are based on the same mathematical principles as light holography (see Plate, 2003, for a review).

Because convolution can code associative information between vectors without losing track of which features belong to which parent vector and the associative information can be stored in the same composite memory representation as the item vectors themselves (e.g., Murdock, 1982), convolution appears particularly appealing for encoding sequential information in language. In the next section, we describe a model that uses convolution to encode word order in language and apply the model to the large-scale statistical structure in a text corpus. First, however, convolution must be adapted into an encoding mechanism that is appropriate for the structure of language.

Memory models predominantly use aperiodic (linear) convolution to bind together pairs of vectors. In aperiodic convolution, the diagonals of the outer-product matrix are summed; this procedure is illustrated in Figure 1. The convolution of $\mathbf{x}$ and $\mathbf{y}$ (both $n$-dimensional vectors) produces a vector containing their association, $\mathbf{z}$, with $2n - 1$ dimensions. The association cannot be directly compared or added to the item vectors because it has a larger dimensionality. To finesse the problem, many models pad the item vectors with zeros to balance dimensionality (e.g., Murdock, 1982) or simply truncate the association vector by trimming the outside elements ($z_{-2}$ and $z_2$ in Figure 1) to match the dimensionality of the item vectors (e.g., Metcalfe-Eich, 1982).

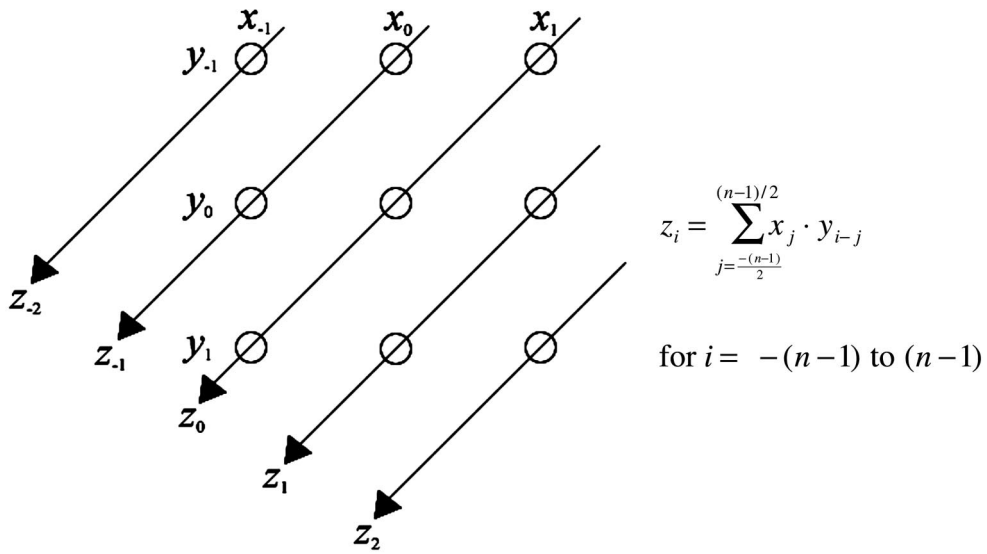


*Figure 1.* Collapsing an outer-product matrix with aperiodic convolution, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the argument vectors and $\boldsymbol{z}$ represents the resulting compressed vector from collapsing the outer-product matrix. The values $i$ and $j$ represent the row and column indices, respectively, for an element in the outer-product matrix.

Expanding dimensionality via convolution is a serious problem for a sequential dependency model to be applied to language that does not stop with binding successive pairs but proceeds recursively to triplets, quadruplets, and so on. Both the padding and the truncating solutions would still limit such a model to learning only pairwise associations around the target word. To associate words in a sentence, one must recursively bind together successive *n*-grams, which produces a serious problem of expanding dimensionality with successive bindings. Generally, convolving $k$ vectors, each having $n$ dimensions, produces an associative vector with $kn - k - 1$ dimensions. Hence, dimensionality of convolved vectors rapidly expands with recursive aperiodic convolution.

Many systems in machine learning use tensor-product algebra to bind together multiple vectors (e.g., Smolensky, 1990); however, in addition to expanding dimensionality with higher order *n*-grams, they also have an expanding rank problem. Thus, the association of *dog bit* is represented as a second-order tensor product (a square matrix), whereas the association of *dog bit mailman* is represented as a third-order tensor product (a cube matrix). Because different-order *n*-grams are represented by higher dimensional or rank structures, they cannot be directly compared with either aperiodic convolution or tensor algebra. Expanding dimensionality makes it difficult to compare different-order *n*-grams or words with sentence fragments. In addition, it cannot be known a priori how large a dimensional structure will be required to encode a sentence.

To avoid the expanding dimensionality problem with recursive convolution, we employ *circular convolution,* a technique used extensively in image and signal processing (Gabel & Roberts, 1973; see also Plate, 1995, 2003, for examples in cognitive modeling). The circular convolution of two *n*-dimensional vectors produces an associative vector, also with dimensionality *n,* using modulo subscripts.[2] The operation of circular convolution is illustrated in Figure 2. The recursive circular convolution of several vectors always has the same dimensionality as the original item vectors and does not waste information by truncating elements.

Like aperiodic convolution, circular convolution is commutative and associative and distributes over addition. It also produces unique associations: The circular convolution of a vector, **a**, with each of two unique vectors, **b** and **c**, produces two unique association vectors. Thus, **a** ⊛ **b** and **a** ⊛ **c** are unique associations[3] even though both contain **a** (e.g., *kick ball* and *kick table* are unique, assuming that *ball* and *table* are unique). It follows from this that circular convolution has a uniqueness of *n*-grams property as well. Thus, the bigram **a** ⊛ **b** is unique from the trigram **a** ⊛ **b** ⊛ **c**. Both associations can be directly compared because they have the same dimensionality, but the higher order *n*-grams are different sources of information than the lower order ones.

The commutative property of circular convolution does, however, create a problem for coding transitions in language because a change in the order of two successively bound items cannot be distinguished. For example, *dog* ⊛ *bite* ⊛ *mailman* and *mailman* ⊛ *bite* ⊛ *dog* are very different ideas, but both produce the same associative vector with convolution. To take advantage of the asymmetric temporal structure of language, a binding operation that is noncommutative is required (i.e., **a** ⊛ **b** is unique from **b** ⊛ **a**). For computational efficiency, we use the noncommutative permutation convolution proposed by Plate (1995). Plate suggested that circular convolution could be made noncommutative by permuting the elements of the two argument vectors differently prior to convolving them. The details of the directional operation are outlined in Appendix A. The result is circular convolution that is neither commutative nor associative but that still distributes over

[2] In practice, a fast Fourier transform (FFT) convolution is computationally more efficient. Convolution via FFT takes $O(n \log n)$ time to compute, whereas the technique with modulo subscripts takes $O(n^2)$ time to compute—this may result in a run time difference of days when trained on a large corpus. Also, FFT lends itself to parallelization more easily.

[3] ⊛ denotes the operation of circular convolution.

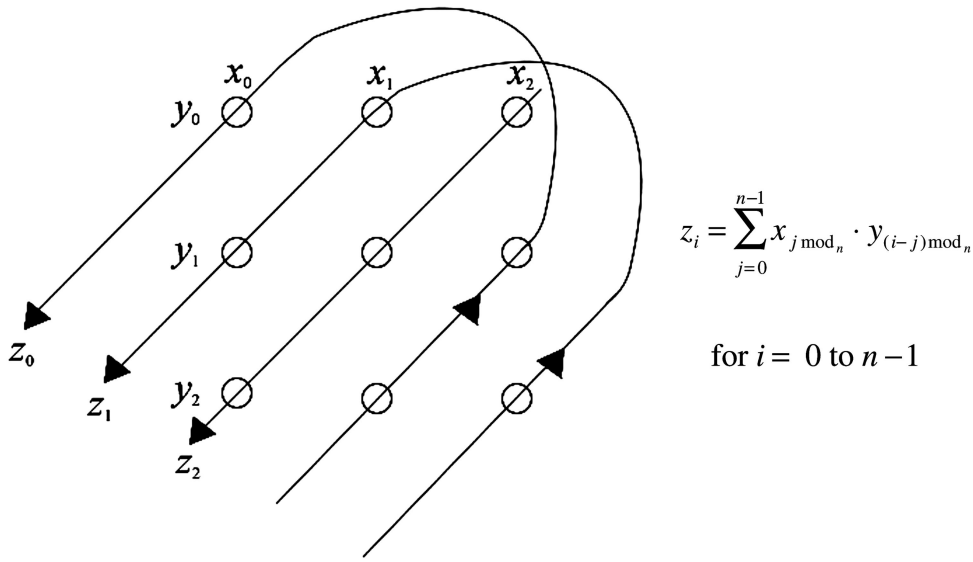


*Figure 2.* Collapsing an outer-product matrix with circular convolution, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the argument vectors and $\boldsymbol{z}$ represents the resulting compressed vector from collapsing the outer-product matrix. The values $i$ and $j$ represent the row and column indices, respectively, for an element in the outer-product matrix.

addition and preserves similarity. Thus, **a** ⊛ **b** and **b** ⊛ **a** are unique associations.

## Statistical Structure in Language

In a sequential data stream, such as a sample of text, there are four nonindependent sources of statistical information that can be used to learn about words. *Co-occurrence* information about a word is information about the word's context, that is, the other words that tend to appear with it in a context unit (phrase, sentence, or document). *Transition* information, on the other hand, is information about the position of a word relative to other words in the context unit.

Both co-occurrence and transition information can come from either *direct* or *indirect* statistical sources. Direct statistical information comes from other words that have directly appeared in the same context unit as a particular word. For example, in the sentence "The robin flew down from the tree and ate the worm," there is a direct co-occurrence relationship formed between *robin, flew,* and *worm* because they have all appeared together. However, there is also an indirect (latent) co-occurrence relationship formed between *robin, feathers, bird,* and *hawk,* even though they may not have directly co-occurred. This is because *robin* and *feathers* have both appeared with *flew* even though *robin* and *feathers* have not appeared together. Furthermore, *robin, bird,* and *hawk* frequently appear around the same words, such as *fly* and *feathers*; this indirect information forms a relationship between the three words even if they have never directly co-occurred.

In the same sample sentence, direct transition relationships are formed between *robin* and its surrounding words—*the* precedes *robin, flew* succeeds it, and *robin* also appears in a trigram flanked by *the* and *flew.* However, indirect transition information is also being learned: *Robin* is similar to *hawk* and *bee* in that all three words may have been found flanked by *the* and *flew*. Early in learning, direct sources of information (both co-occurrence and transition) are more important to learning word meanings; however, as more words are learned, progressively more information can be learned from indirect sources (both co-occurrence and transition).

## The BEAGLE Model

In this section, we describe *bound encoding of the aggregate language environment* (BEAGLE), a computational model that builds a semantic space representation of meaning and word order directly from statistical redundancies in language. We first document how the model develops a distributed representation of meaning for words from contextual similarity and then how the model can develop a representation of word order by forming associations between words with vector convolution (binding). Finally, we demonstrate how these two types of information (word meaning and word order) can be learned together as a single pattern of vector elements in a composite lexical representation and how sequential dependencies can be retrieved from the lexicon.

We assume that a word's meaning and usage are a single pattern of vector elements. This lexical representation is initially a random pattern, and gradually, representation noise is reduced across experience as the word is sampled in different contexts and positions within contexts. With sufficient samples, a stable central limit for each vector element emerges, stabilizing the lexical pattern as a whole. Depending on variability of a word's meaning and usage in language, more or fewer observations of the word are required to converge on a stable lexical representation. Statistical noise reduction is a simple yet powerful mechanism for learning about a word's meaning and order information.

Borrowing terms used in associative memory (Murdock, 1982, 1992), information about word co-occurrence is referred to as

*context* information and information about word usage—position relative to other words in a sentence—as *order* information. Similar to Murdock's (1982, 1992) models, context information comes from vector addition, and order information comes from vector convolution.

BEAGLE processes text one sentence at a time, learning context and order information for each word before proceeding to the next sentence. Front-end routines are used to parse text into discrete units of meaning, where a unit is usually a sentence (other types of punctuation, such as semicolons and parentheses, are also interpreted as marking a complete unit of meaning).

Across many sentences, the representation of a word becomes similar to the representation of words that frequently co-occur in sentences with it and to those words that frequently occur in similar contexts. For example, the representation of *robin* is similar to the representation for *fly* because the two frequently co-occur in the same sentence (a direct relationship). However, *robin* is also similar to *hawk* even though the two may never have co-occurred in the same sentence. This is because *robin* and *hawk* both tend to co-occur in contexts with similar words, such as *fly* and *feathers* (an indirect or latent relationship). In addition, *robin* and *hawk* share order similarity in that they are often used in similar positions relative to other words in their respective sentences. *Robin* and *hawk* have shared order information, as do *fly* and *sing,* but *robin* and *sing* do not have shared order information.

### *Representation*

Words are represented by high-dimensional holographic vectors. The first time a word is encountered, it is assigned a random *environmental* vector, $\boldsymbol{e}_i$. Environment vector elements are sampled at random from a Gaussian distribution with $\mu = 0$ and $\sigma = 1/\sqrt{D}$, where $D$ is the vector dimensionality. Each subsequent time the same word is encountered, the same environmental vector is used in coding context and order information; thus, a word's environmental representation does not change. A word's *memory* (lexical) vector, $\boldsymbol{m}_i$, however, changes each time the word is encountered in a new context unit. Each time a word is encountered, its environmental vector is used in the coding of its new context and order information, and this new information is added to $\boldsymbol{m}_i$. The environmental vectors are intended to represent the physical characteristics of words in the environment (e.g., orthography, phonology, etc.), whereas the memory vectors represent internal memory for contexts and positions in which the environmental vectors have been encountered. At this point, we are agnostic about the actual environmental features for words; hence, we assume no structural similarities between words and represent each with a different random representation. Environment vectors are generated at random the first time a word is encountered, so each word is physically unique from other words in $\boldsymbol{e}$. By uniqueness, we specifically mean that the expected cosine between any two vectors is zero. To add structural similarities between words, the $\boldsymbol{e}$ matrix could be generated to reflect known structural similarities between words (see Heiser, 1988; Van der Heijden, Malhas, & Van den Roovaart, 1984), but that is not the focus of the present work.

### *Learning Context Information*

The context information for a word in a sentence, $\boldsymbol{c}_i$, is the sum of the environmental vectors for the other $n - 1$ words in the sentence:

$$\boldsymbol{c}_i = \sum_{j=1}^{n} \boldsymbol{e}_j, \quad \text{where } i \neq j. \tag{1}$$

Context vectors are computed for each word in the sentence. A word's lexical representation, $\boldsymbol{m}_i$, is then updated by adding the new context information to it:

$$\boldsymbol{m}_i = \boldsymbol{m}_i + \boldsymbol{c}_i. \tag{2}$$

For each sentence a word is encountered in, the new context information is constructed from the environmental vectors of the other words in the sentence, and the lexical representation is updated by adding the new context vector to it. A word's lexical representation is, thus, a superposition of vectors that reflects the word's history of co-occurrence with other words within sentences.

High-frequency function words (e.g., *the, a, of*) can pollute a lexical representation formed in this way because they frequently co-occur in sentences with many words. For example, the lexical vector for *dog* could be most similar to the lexical vector for *the* simply because the two always co-occur in the same sentence. With large enough dimensionality, *the* may be a near neighbor of almost every word.

We have used continuous entropy functions to allow the model to degrade the contribution of high-frequency function words gracefully as it learns that they tend to appear often and in many different sentences. These functions do not require any a priori knowledge of word frequency. For large-enough text samples, an entropy function can learn which words in the sentence contribute useful semantic information and can approximate the outcome that would be observed if word frequencies were known a priori. For computational efficiency, however, the simulations reported here employ a standard stop list of 280 function words in the calculation of context information. If a word appears in the stop list, it is omitted from the calculation of context information. Stop lists of this sort are commonly used in corpus-based models of semantics (e.g., A. E. Smith & Humphreys, 2006; Steyvers & Griffiths, in press; etc.).

As an example of context information, consider coding the word *dog* in the sentence "A dog bit the mailman." The context information would be the sum of the vectors for the other words in the sentence, without the stop list words *a* and *the*:

$$\boldsymbol{c}_{dog} = \boldsymbol{e}_{bit} + \boldsymbol{e}_{mailman}.$$

The context of *dog* in this sentence is the superposition of the words *bit* and *mailman*. This new sentence information is then added to $\boldsymbol{m}_{dog}$, which may already contain past superimposed vectors, perhaps *tail, wag, fetch,* and so on.

A direct relationship has been formed between *dog, bit,* and *mailman* because they have some of the same random information summed into their lexical representations. For example, the context information for *bit* and *mailman* in the same sentence is

Table 1
*Examples of Nearest Neighbors to Various Target Words in Context Space*

| COMPUTER | | HOSPITAL | | HEART | | COGNITIVE | | LEXICAL | | BEER | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| data | .81 | doctor | .68 | artery | .72 | development | .62 | syntactic | .51 | wine | .56 |
| computers | .68 | medical | .65 | vessels | .71 | child | .57 | semantic | .44 | liquor | .54 |
| processing | .68 | nurse | .63 | pumps | .71 | intellectual | .55 | prefixes | .43 | whiskey | .53 |
| processed | .62 | patients | .62 | arteries | .69 | learning | .54 | derivational | .42 | drinks | .47 |
| storage | .62 | patient | .62 | blood | .68 | stages | .51 | context | .42 | alcohol | .42 |
| program | .61 | emergency | .59 | veins | .66 | research | .49 | meaning | .41 | drink | .40 |
| programmer | .61 | care | .59 | circulation | .65 | personality | .49 | meanings | .40 | alcoholic | .40 |
| microcomputer | .61 | clinic | .57 | pumping | .63 | piaget's | .47 | suffixes | .39 | vodka | .40 |
| keyboard | .60 | doctors | .55 | clot | .62 | piaget | .47 | inflectional | .39 | ethyl | .37 |
| memory | .60 | nursing | .53 | clotting | .61 | psychological | .47 | synonym | .37 | rum | .37 |
| input | .60 | physician | .53 | aorta | .61 | psychologists | .47 | word | .37 | beverage | .35 |
| user | .59 | ward | .52 | capillaries | .59 | behavior | .46 | words | .35 | bottle | .35 |
| cpu | .58 | call | .51 | transfusion | .58 | skills | .46 | convey | .35 | depressant | .33 |
| electronic | .58 | hospitals | .51 | vessel | .57 | theory | .45 | dictionary | .34 | ethanol | .33 |
| information | .58 | inpatient | .51 | arterial | .57 | understanding | .45 | grammatical | .34 | sipped | .32 |
| software | .57 | surgery | .50 | ventricles | .56 | concept | .45 | connotation | .33 | cola | .32 |
| disk | .56 | office | .49 | beat | .56 | communication | .45 | denotation | .33 | intoxicated | .31 |
| computing | .56 | birth | .48 | liver | .55 | education | .45 | morphemic | .33 | tasted | .30 |
| digital | .55 | ambulance | .48 | heartbeat | .55 | motivation | .44 | mythos | .33 | lemonade | .30 |
| analyze | .53 | sick | .48 | hypertension | .53 | knowledge | .44 | phrase | .32 | pint | .29 |

*Note.* Numbers following neighbor words are vector cosines.

$$\boldsymbol{c}_{bit} = \boldsymbol{e}_{dog} + \boldsymbol{e}_{mailman},$$

and

$$\boldsymbol{c}_{mailman} = \boldsymbol{e}_{dog} + \boldsymbol{e}_{bit}.$$

What is less obvious is that a latent relationship in the lexicon is also being formed between *dog* and *wolf*: Although the *dog* is biting the mailman in one region of the corpus, the *wolf* is biting/chasing/attacking a deer/hunter in another. The more often *dog* and *wolf* are found in separate sentences doing similar things, the more similar their lexical representations become to each other. The more frequently two words share context (either directly or indirectly), the more common random vectors are summed into their lexical representations and, hence, the more similar their lexical representations become to one another.

Before we describe the coding of order information in BEAGLE, we first provide examples of the semantic structure that emerges from this simple vector superposition algorithm when applied to real-world language. For the simulations reported here, vector dimensionality was set to 2,048, and 90,000 words were learned into the lexicon. The model was trained on the TASA corpus (compiled by Touchstone Applied Science Associates; see Landauer et al., 1998), which contains a collection of English text that is approximately equivalent to what the average college-level student has read in his or her lifetime. TASA is the same corpus on which the Web version of LSA is trained.[4]

Table 1 illustrates some nearest neighbors and their similarities[5] to various target words (target words are capitalized) in the lexical space when trained on the TASA corpus with only context information. There were 90,000 words learned in total; the table displays only a sample of structure from the lexicon. As the table shows, vectors representing semantically similar words have developed similar patterns of elements as a function of shared context. Some of this co-occurrence information comes from direct relationships and some from indirect relationships.

The lexicon can also be thought of as a multidimensional space, with similar vectors being more proximal in the space and dissimilar vectors being more distant. Figure 3 shows a two-dimensional scaling plot of various financial, science, and sports terms learned from the textbase. Initially, there is chaotic structure in this space, with the words positioned randomly around the origin (the cross hair). With textbase experience, however, structure forms from the accumulation of common random vectors in each lexical representation. For example, the financial terms become similar to one another and distinct from the science terms and sports terms. Although *soccer, baseball,* and *football* may never have directly co-occurred with one another, they have had some of the same random information summed into their lexical representations, such as *player, sport,* and *ball*. Sentences and documents can also be plotted in terms of their proximity to concepts.

Random vector accumulation is a simple yet powerful method of acquiring semantic representations for words from contextual co-occurrence. The representations gradually form structure from initial randomness via noise reduction across experience. As mentioned previously, however, co-occurrence information is not the only source of statistical information from which a word's meaning can be induced. Transition information about a word defines its lexical class and grammatical behavior as well as contributing to its meaning. Transition information has been absent from most semantic space models, and the addition of this information may benefit semantic space accounts in a variety of tasks.

We used random vector accumulation because it allows the model to learn co-occurrence into a fixed-dimensional representation, whereas representation dimensionality in other semantic space models depends largely on the input matrix and the optimi-

[4] We are grateful to Tom Landauer for providing the TASA corpus.

[5] *Similarity* in this article refers specifically to the cosine of the angle between two vectors (a normalized dot product).

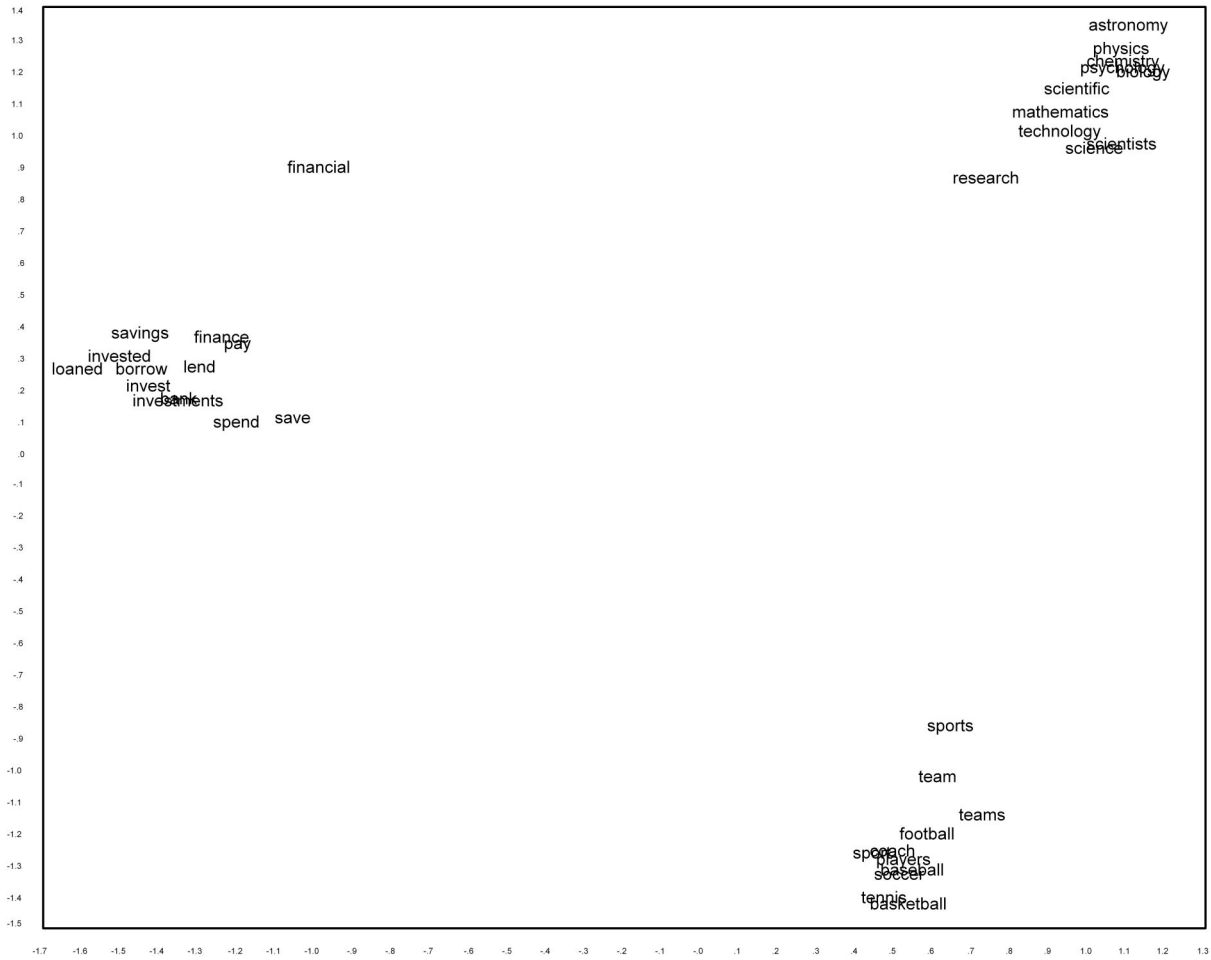


*Figure 3.* An example of word clustering in a subset of the context space with financial, science, and sports concepts.

zation criterion. Using the circular convolution operation described previously, we can now represent a word's order history in the same fixed dimensionality and can merge the two types of information into a unified representation.

### *Learning Order Information*

A word's order information is calculated by binding it with all *n*-gram chunks in the sentence that include it. Binding in BEAGLE is accomplished via directional circular convolution. The position of the word being coded is represented by a constant random placeholder vector, $\mathbf{\Phi}$ (sampled from the same element distribution from which the environment vectors were constructed). Phi is fixed across learning. This method uniquely codes associations for the bigram, trigram, quadgram, and other such neighbors around the word. The *n*-gram convolutions are unique from one another even if they contain many of the same words. The order information for a word in a sentence is the sum of all the *n*-gram convolutions to which it belongs. Because circular convolution is used, all *n*-grams are represented in the same dimensionality, and hence, they can all be summed into a single order vector, $\boldsymbol{o}_i$, which represents the word's associative position relative to the other words in the sentence:

$$\boldsymbol{o}_i = \sum_{j=1}^{p\lambda-(p^2-p)-1} bind_{i,j}, \tag{3}$$

where $p$ is the position of the word in the sentence, and $bind_{i,j}$ is the $j$th convolution binding for the word being coded. The word's order vector is then added to its lexical vector (as was the context vector), which becomes a pattern of elements representing the word's position relative to words in all sentences learned.

Lambda (λ) is a chunking parameter that sets the maximum neighbors a word can be bound with (because the number of possible *n*-grams increases as a nonlinear function of position and sentence length). Lambda is not a theoretical limit but, rather, a practical one. The number of possible convolutions becomes unmanageably large in very long sentences. Computing all possible convolutions can drastically increase compute times, and very-high-order *n*-grams are unlikely to contribute generalizable order

Table 2
*Examples of Nearest Neighbors to Various Target Words in Order Space*

| MIKE | | SIX | | WENT | | BELOW | | HER | | MINUTES | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| dan | .86 | eight | .68 | ran | .84 | above | .86 | his | .89 | seconds | .88 |
| tom | .84 | five | .65 | came | .84 | beneath | .84 | my | .83 | moments | .76 |
| pete | .83 | seven | .63 | rushed | .83 | under | .83 | their | .79 | hours | .63 |
| ben | .83 | four | .62 | hurried | .82 | beyond | .83 | your | .76 | weeks | .62 |
| grandpa | .83 | nine | .62 | returned | .80 | across | .83 | its | .75 | days | .59 |
| jeff | .83 | ten | .59 | goes | .79 | near | .82 | the | .73 | months | .59 |
| ted | .82 | three | .59 | walked | .77 | on | .82 | our | .72 | inches | .55 |
| jim | .82 | twelve | .57 | moved | .77 | against | .82 | them | .69 | milliseconds | .55 |
| tim | .82 | eighteen | .55 | continued | .77 | through | .82 | him | .66 | innings | .47 |
| charlie | .81 | twenty | .53 | began | .77 | behind | .82 | me | .64 | yards | .42 |
| pam | .81 | thirty | .53 | proceeded | .76 | around | .82 | a | .63 | decades | .42 |
| dad | .81 | forty | .52 | clung | .76 | on | .81 | myself | .61 | beers | .41 |
| tommy | .81 | two | .51 | refused | .75 | in | .81 | herself | .60 | paces | .41 |
| joe | .80 | fifteen | .51 | fell | .75 | between | .80 | himself | .60 | cents | .40 |
| dave | .80 | several | .51 | rode | .75 | throughout | .80 | whom | .59 | ounces | .40 |
| bobby | .79 | eleven | .50 | drove | .75 | inside | .79 | an | .59 | years | .39 |
| jack | .79 | seventeen | .49 | appealed | .74 | within | .79 | these | .58 | centimeters | .37 |
| danny | .78 | fifty | .48 | sought | .73 | upon | .78 | any | .57 | spoonfuls | .36 |
| woody | .78 | twentyfive | .48 | liked | .72 | before | .76 | yourself | .57 | bites | .35 |
| andy | .77 | fourteen | .48 | listened | .71 | among | .76 | this | .57 | centimetres | .35 |

*Note.* Numbers following neighbor words are vector cosines.

structure.[6] For the simulations reported here, lambda was set to seven after Miller and Selfridge's (1950) classic experiments with order of approximation to English.

As an example, consider coding the order information for the word *dog* in the sentence "A dog bit the mailman." Because function words are very important to syntactic structure, no stop list or entropy weighting is used in the calculation of order information. The order information for *dog* would be all of the *n*-gram bindings around it.

$$\left.\begin{aligned} bind_{dog,1} &= \boldsymbol{e}_a \circledast \boldsymbol{\Phi} \\ bind_{dog,2} &= \boldsymbol{\Phi} \circledast \boldsymbol{e}_{bit} \end{aligned}\right\}\text{Bigrams}$$

$$\left.\begin{aligned} bind_{dog,3} &= \boldsymbol{e}_a \circledast \boldsymbol{\Phi} \circledast \boldsymbol{e}_{bit} \\ bind_{dog,4} &= \boldsymbol{\Phi} \circledast \boldsymbol{e}_{bit} \circledast \boldsymbol{e}_{the} \end{aligned}\right\}\text{Trigrams}$$

$$\left.\begin{aligned} bind_{dog,5} &= \boldsymbol{e}_a \circledast \boldsymbol{\Phi} \circledast \boldsymbol{e}_{bit} \circledast \boldsymbol{e}_{the} \\ bind_{dog,6} &= \boldsymbol{\Phi} \circledast \boldsymbol{e}_{bit} \circledast \boldsymbol{e}_{the} \circledast \boldsymbol{e}_{mailman} \end{aligned}\right\}\text{Quadgrams}$$

$$bind_{dog,7} = \boldsymbol{e}_a \circledast \boldsymbol{\Phi} \circledast \boldsymbol{e}_{bit} \circledast \boldsymbol{e}_{the} \circledast \boldsymbol{e}_{mailman}\}\text{Tetragram}$$

$$\boldsymbol{o}_{dog} = \sum_{j=1}^{7} bind_{dog,j}$$

Each of the *n*-gram convolutions produced is unique, and each pattern is stored in the $\boldsymbol{o}$ vector with superposition. In the above sentence, the trigram convolution for *a dog bit* is unique from the bigram convolution for *a dog*; higher order *n*-gram associations are different from their lower order constituents.

For example, $bind_{dog,1}$ codes the bigram association of *a* immediately preceding the word being coded (*dog*). Furthermore, $bind_{dog,2}$ records *bit* immediately succeeding the word being coded, and $bind_{dog,3}$ codes the trigram association that the word being coded appeared flanked by *a* on its left and *bit* on its right. Even though the trigram *a dog bit* contains the bigrams *a dog* and *dog bit*, the trigram association vector is unique compared with either of its bigram constituents. All of the above convolutions are unique from one another, but because circular convolution is being used as the coding operation, they compress into the same dimensionality and can all be summed into a single-order vector that represents *dog*'s position in this sentence relative to the other words. This $\boldsymbol{o}$ vector is then added to *dog*'s lexical representation, which contains the position of *dog* relative to other words in all sentences learned.

The lexical representation thus acquires a pattern of elements that reflects the word's history of position association with other words in sentences. As text is experienced, more common associations become stronger (from superposition), and less common associations are dampened. This produces a natural weighting in the lexicon where lower order *n*-grams are more important to a word's order history than higher order *n*-grams simply because lower order *n*-grams are more likely to be consistently encountered as stable chunks across experience.

To demonstrate the structure learned by the convolution order algorithm, Table 2 shows some nearest neighbors to various target words (target words are capitalized) in the lexicon when trained on TASA with order information only (2,048 dimensions and 90,000 words). It is clear from the table that a different pattern of structure has emerged with the order-encoding algorithm, namely, lexical classes. An action verb, such as *went,* is most similar to other action verbs and gradually to other types of verbs. A locative, such as *below,* is most similar to other locatives and then to some temporal prepositions. A near neighbor to a noun is unlikely to be a verb in this space, as was possible in the context space.

[6] Information about long-distance dependencies in sentences can be bootstrapped from the context information.

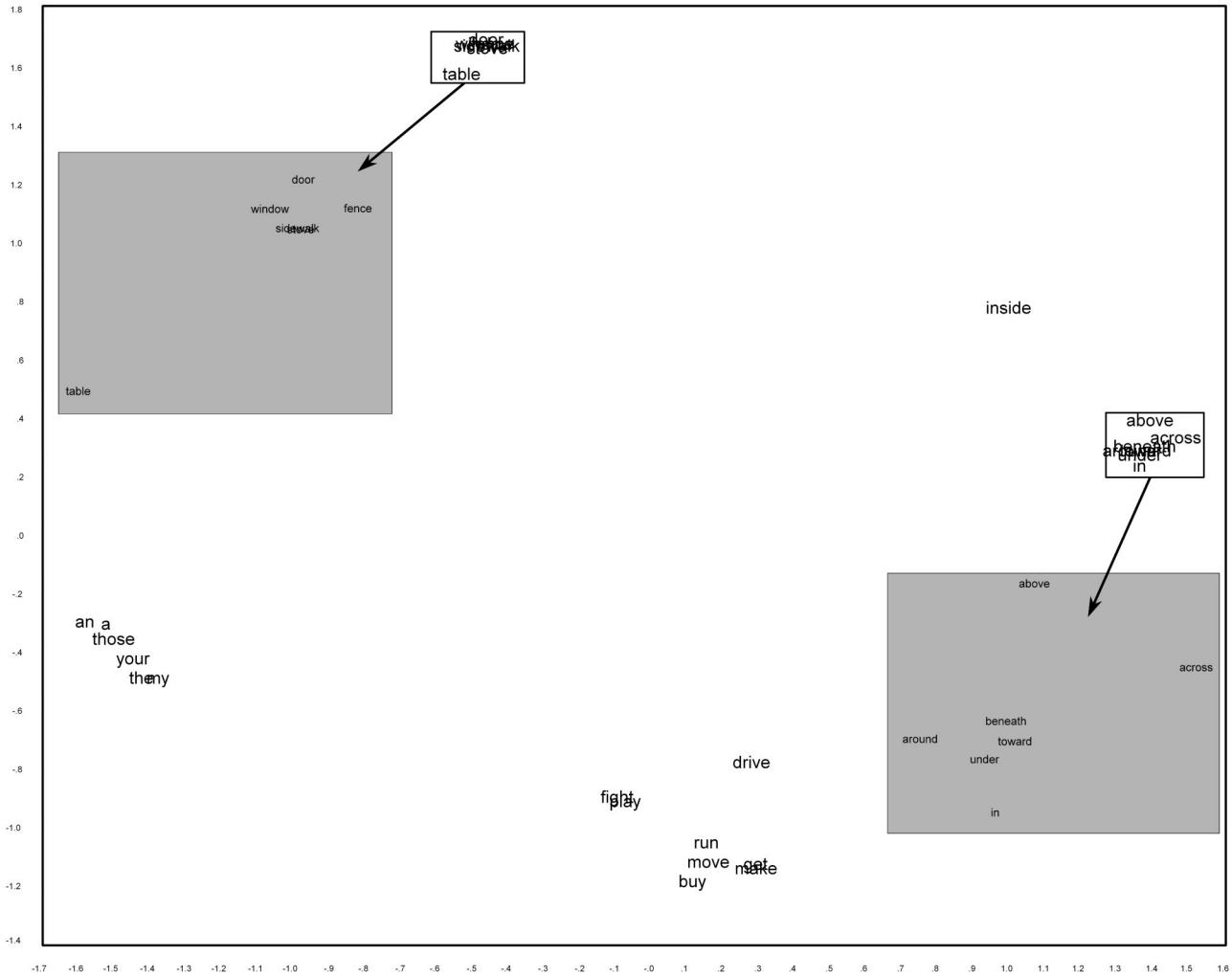


*Figure 4.* An example of word clustering in a subset of the order space with nouns, prepositions, verbs, and determiners (clockwise from top). The gray regions show expansions of the dense neighborhoods.

Figure 4 shows a two-dimensional scaling plot for four parts of speech: nouns, locatives, verbs, and determiners (clockwise from top group). The shaded regions show expansions of dense areas. Because of similar associations during learning, words that are commonly found in similar positions relative to other words have developed similar patterns of vector elements; that is, they have had common random vectors convolved into their lexical representations. Some of this transition information comes from direct relationships, and some comes from indirect relationships. Note that the relationships learned from order information are very different from those learned from context information, and a comprehensive model should be able to consider both types of information together.

### *Contextual and Order Information Together*

For demonstrative purposes, we have described the structure learned from context and order information separately. In practice, BEAGLE codes context and order information into the same lexical vector for a word each time it is encountered in a new context unit.[7]

$$\boldsymbol{m}_i = \boldsymbol{m}_i + \sum_{j=1}^{N} \boldsymbol{e}_j + \sum_{j=1}^{p\lambda-(p^2-p)-1} bind_{i,j} \tag{4}$$

$$= \boldsymbol{m}_i + \boldsymbol{c}_i + \boldsymbol{o}_i. \tag{5}$$

The composite lexical vector contains the superposition of a word's context and order information across all the sentences in which it has been experienced. This composite representation has the powers of both context and order information together: It contains semantic and associative information, and a word's

[7] The vectors representing context and order information are each normalized to a length of one prior to averaging; otherwise, one type of information may overpower the other depending on sentence length.

position information can be retrieved from the order information (demonstrated in the next section) given the context of surrounding words.

Figure 5 shows scaling plots that demonstrate the organization between words learned by the context (see Figure 5A) and order (see Figure 5B) mechanisms using the same words. For the context information (see Figure 5A), verbs are proximal to the nouns upon which they operate. For example, *food* is related to *eat, car* is related to *drive,* and *book* is related to *read,* but *eat, drive,* and *read* are not highly related to one another, nor are *food, car,* and *book*. The context information accounts for primarily semantic associations.

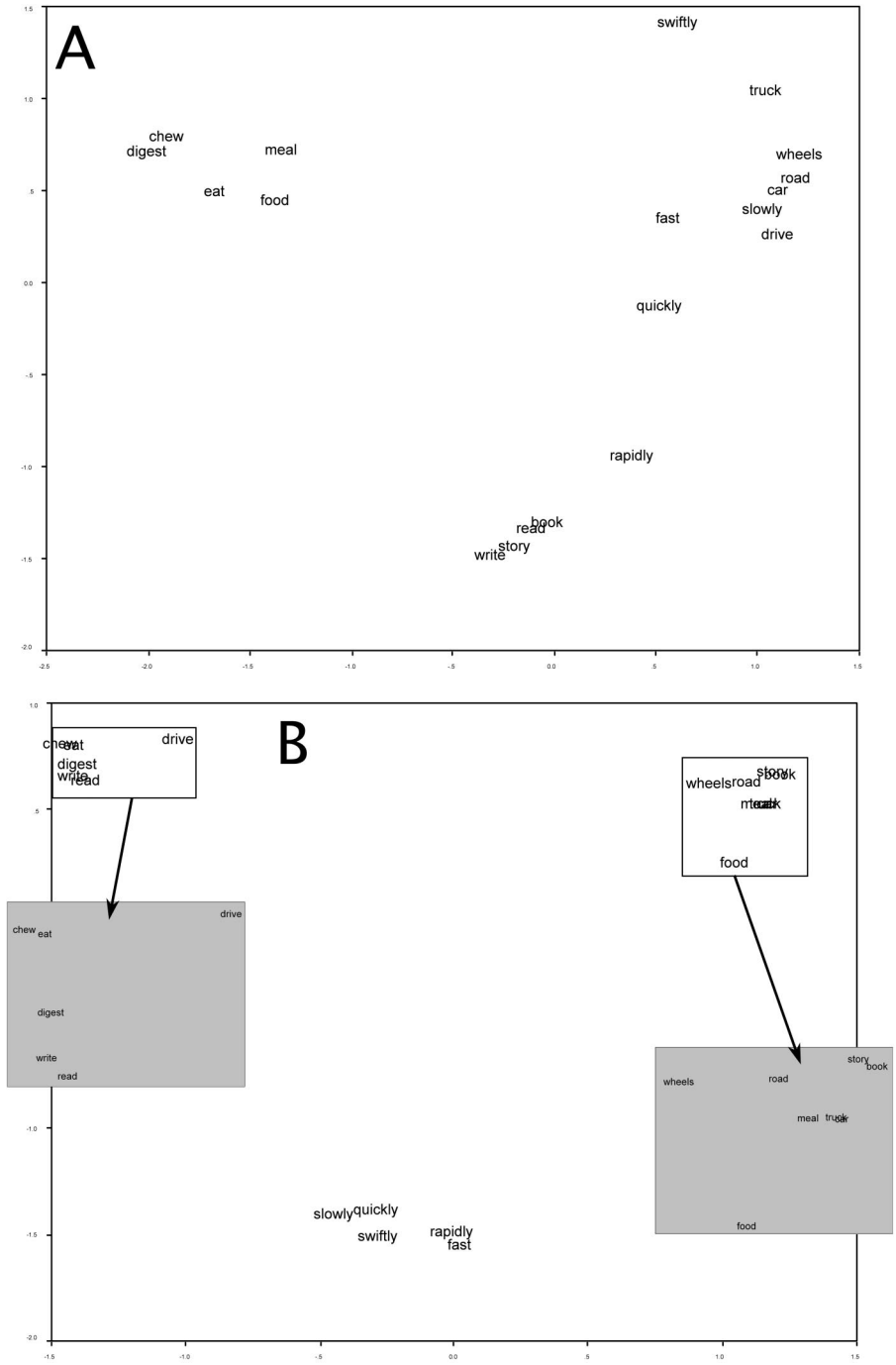


*Figure 5.* A: A subset of nouns, verbs, and adjectives in the context-only space; in context space, verbs are proximal to the nouns they operate upon from similar contextual experience. B: The same words in the order-only space; in order space, parts of speech form proximal clusters from similar experience.

By contrast, Figure 5B shows the structure of the same words as learned by the order algorithm (using convolution). In order space, words that appear in similar positions relative to other words in sentences have developed similar patterns from accumulation of common association vectors. *Drive, eat,* and *read* are now all proximal to one another and cluster distinctly from the nouns (*car, food,* and *book* now being similar to one another). In the context space, the adverbs are close to their verbs (e.g., *quickly* and *slowly* are proximal to *drive*). In the order space, however, the adverbs have formed their own cluster. The context algorithm has learned semantic relationships, whereas the order algorithm seems to have learned information about lexical classes. Semantic class information is also represented in the order space: Although nouns are distinct from verbs, within nouns, animates tend to be distinct from inanimates. Within animates, for example, fish cluster together, and birds cluster together. In the context space, however, fish also cluster with *gills* and *swim,* and birds cluster with *wings* and *fly.* Contextual and order information complement each other.

Table 3 shows the top 12 neighbors (and their cosines) to *eat, car, reading,* and *slowly* in the full versions of each space (i.e., using all words in the lexicon). For verbs such as *eat* or *reading,* the neighbors in context space are predominantly semantic associates, whereas the neighbors in order space are similar verbs. It is interesting to note that the nearest neighbors in the composite space may not necessarily be close to the target in either context or order space alone. The organization in the composite space often emphasizes higher order semantic relationships, for example, the idea that *eat* is related to *grow* or that *reading* is related to *understanding* (neither *grow* nor *understanding* is on the list of top neighbors in context or order space alone).

### *Retrieving Order Information From the Lexicon*

In BEAGLE, order information is incorporated into context information to produce a higher fidelity representation of word meaning. As a by-product of using convolution as an order-binding mechanism to represent associative position of the word in sentences, the order information can also be retrieved from the lexical representations, allowing the model to predict word transitions in sentences from its learned history (without explicitly coded rules of grammaticality or transition).

Coded into a word's lexical representation is its preferred position relative to other words it has occurred with in sentences. This information may be retrieved from the lexicon in two ways: Order information may be *decoded* from lexical representations to determine words that are likely to precede or succeed a given order context, or an order pattern may be compared with the lexicon to determine which lexical entries *resonate* with it (i.e., which words have been learned in the order context with a high frequency). Decoding retrieves an instance from the learned associative infor-

Table 3
*Twelve Nearest Neighbors to* Eat, Car, Reading, *and* Slowly *in Context, Order, and Composite Spaces*

| EAT | | | | | | CAR | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Context | | Order | | Composite | | Context | | Order | | Composite | |
| eaten | .85 | buy | .91 | feed | .76 | driver | .72 | boat | .95 | truck | .80 |
| eating | .81 | get | .90 | grow | .71 | drive | .71 | ship | .94 | road | .76 |
| food | .70 | sell | .89 | produce | .70 | driving | .68 | truck | .94 | driver | .75 |
| hunt | .67 | move | .89 | die | .68 | road | .67 | house | .93 | bus | .75 |
| digest | .65 | save | .89 | digest | .68 | drove | .67 | bus | .93 | train | .73 |
| ate | .65 | sleep | .88 | kill | .67 | wheels | .67 | computer | .93 | garage | .73 |
| grow | .64 | keep | .88 | chew | .66 | truck | .67 | fire | .93 | highway | .72 |
| need | .64 | swallow | .88 | survive | .66 | seat | .63 | train | .93 | house | .71 |
| foods | .63 | avoid | .88 | hunt | .66 | drivers | .62 | bank | .92 | street | .70 |
| plants | .62 | win | .87 | provide | .66 | parked | .60 | camera | .92 | fire | .70 |
| insects | .60 | catch | .87 | cook | .65 | cars | .59 | ball | .92 | horse | .70 |
| nutritious | .60 | produce | .87 | preserve | .65 | street | .59 | dog | .91 | wagon | .69 |

| READING | | | | | | SLOWLY | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Context | | Order | | Composite | | Context | | Order | | Composite | |
| read | .73 | writing | .73 | writing | .72 | quickly | .67 | quickly | .70 | quickly | .68 |
| book | .70 | making | .66 | learning | .57 | turned | .62 | carefully | .53 | rapidly | .52 |
| books | .68 | studying | .63 | studying | .56 | walked | .60 | rapidly | .52 | suddenly | .50 |
| reading | .63 | teaching | .60 | understanding | .55 | moved | .59 | freely | .47 | moved | .48 |
| facts | .62 | planning | .60 | books | .53 | moving | .58 | easily | .47 | turned | .48 |
| readers | .62 | describing | .60 | teachers | .53 | left | .58 | quietly | .46 | running | .47 |
| authors | .61 | using | .59 | mathematics | .53 | saw | .57 | swiftly | .45 | walked | .47 |
| write | .60 | cutting | .58 | speech | .52 | move | .56 | clearly | .45 | reached | .46 |
| words | .59 | seeing | .57 | comprehension | .51 | suddenly | .56 | suddenly | .44 | stopped | .46 |
| comprehension | .58 | taking | .57 | skills | .51 | reached | .55 | costly | .43 | carefully | .46 |
| learn | .58 | selecting | .57 | information | .50 | close | .55 | softly | .43 | rose | .45 |
| library | .57 | watching | .56 | language | .49 | street | .55 | efficiently | .42 | left | .45 |

*Note.* Numbers following neighbor words are vector cosines.

mation in a holographic vector with inverse convolution and compares the retrieved instance with the environment to determine goodness of fit. Resonance, on the other hand, compares an associative instance in the environment with memory to determine which memory vectors are highly excited by the pattern. The specific details of encoding and decoding in BEAGLE are detailed in Appendix A.

*Retrieving order information by decoding.* Learned items may be decoded from the order information in a lexical representation using inverse circular convolution (a.k.a., circular correlation), $y \approx x \circledast z$:

$$y \approx \sum_{j=0}^{n-1} x_{j \bmod n} \cdot z_{(i+j) \bmod n}.$$

For example, to predict the next word in a sequence from the lexicon, information is decoded from the lexical representations for the words in the sequence so far. The retrieved vector is a noisy version of the environmental vector for the most likely next transition in the sequence, given the preceding transitions.

Partially because convolution throws away some information, partially because composite compression introduces noise, and partially because correlation is only an approximate inverse to convolution, the decoded vector is a noisy version of its original form.[8] Nonetheless, the retrieved vector has a higher cosine with its original environmental form than with any of the other environment vectors (see Murdock, 1982, 1993, for characteristics of convolution–correlation and retrieval facsimiles).

For example, consider *martin luther king jr,* a frequently occurring and fairly unique chunk in the TASA corpus. The lexical vectors for *martin, luther, king,* and *jr* have relatively little other associative information coded into them because they frequently appear as a stable chunk (with the exception of *king,* which appears in many other name chunks as well). Decoding $\boldsymbol{m}_{luther}$ to the left[9] retrieves a facsimile of $\boldsymbol{e}_{martin}$, and decoding $\boldsymbol{m}_{luther}$ to the right retrieves a facsimile of $\boldsymbol{e}_{king}$.

Figure 6 illustrates the similarity of the vectors decoded from $\boldsymbol{m}_{luther}$ (either right or left) to all 90,000 environmental vectors used in learning. Figure 6A shows the cosine of all environmental vectors $\boldsymbol{e}_{1..N}$ to the vector retrieved by decoding $\boldsymbol{m}_{luther}$ to the right (i.e., words that succeeded *luther*). The retrieved vector most clearly resembles the environmental vector for $\boldsymbol{e}_{king}$ that was learned to succeed *luther* and is only randomly similar to the 89,999 other environmental vectors. By contrast, Figure 6B shows the cosine of all environmental vectors to the vector retrieved by decoding $\boldsymbol{m}_{luther}$ to the left (i.e., words that preceded *luther*). The retrieved vector most clearly resembles the environmental vector for $\boldsymbol{e}_{martin}$ that was learned to precede *luther* and shows only random similarity to the others. From the same lexical representation of $\boldsymbol{m}_{luther}$, different directional associations can be decoded.

As more *n*-grams are included in the probe, the decoded response is more constrained. For example, the fidelity of the vector retrieved in its similarity to $\boldsymbol{e}_{king}$ increases as consistent words are added because there are more sources to decode from. Given *luther* ____, only $\boldsymbol{m}_{luther}$ can be used to decode from. However, given *martin luther* ____, the blank position can be decoded from $\boldsymbol{m}_{luther}$ but also from $\boldsymbol{m}_{martin}$, given that *luther* must be in the intervening position between *martin* and the position being predicted.

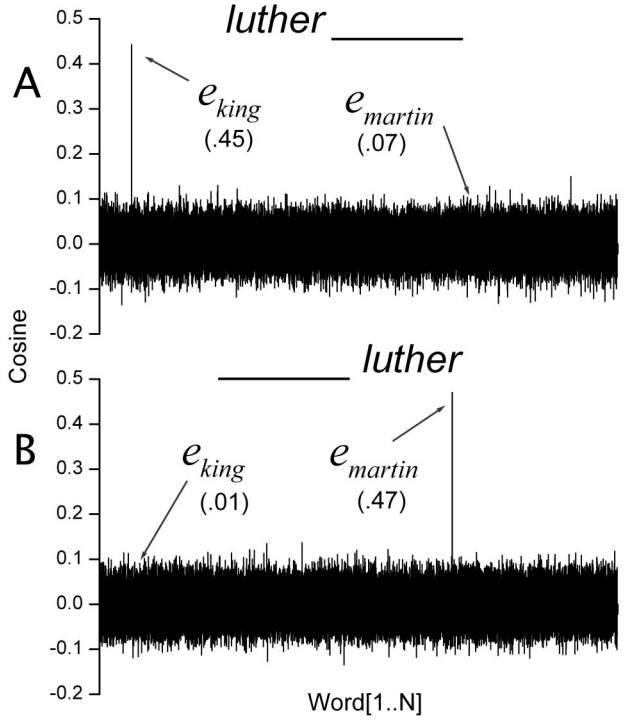


*Figure 6.* Cosine between decoded vectors and all 90,000 environmental vectors used in learning. A: Similarity of the vector decoded from *luther* in the succeeding position. Clearly, this decoded vector is most similar to the environmental vector representing *king*. B: Similarity of the vector decoded from *luther* in the preceding position. This decoded vector is most similar to the environmental vector representing *martin*. Forward and backward associations can be decoded from the same lexical representation.

Given *martin luther* ____ *jr*, the blank position can be further estimated by decoding $\boldsymbol{m}_{jr}$ backwards. Adding context around a word can either modify the retrieval if it changes the order constraint or emphasize a word if it is consistent with the information retrieved from the other memory vectors. If the words all agree on what word should be retrieved (i.e., *luther* ____ and *martin luther* ____ both agree that $\boldsymbol{e}_{king}$ should be retrieved), then the fidelity of the retrieved vector to $\boldsymbol{e}_{king}$ simply increases. For example, given *luther* ____, *martin luther* ____, and *martin luther* ____ *jr,* the similarity of the retrieved vector to $\boldsymbol{e}_{king}$ increases to .45, .65, and .70, respectively, as the additional words are added around the position being decoded. The specifics of decoding are described in detail in Appendix A.

*Retrieving order information with resonance.* As demonstrated in the previous section, order information may be decoded from a specific lexical trace. However, order information is also distributed across the lexicon. For example, in learning the sequence "A dog bit the mailman," the transition information for *dog bit* is stored in the lexical entry for *dog,* $\boldsymbol{m}_{dog} = \boldsymbol{m}_{dog} + (\boldsymbol{\Phi} \circledast \boldsymbol{e}_{bit})$. However, information about the same transition is also stored in

[8] Correlation is more stable in a system with noise than is retrieval with the exact inverse (Plate, 2003).

[9] TASA contains no information about *king luther.*

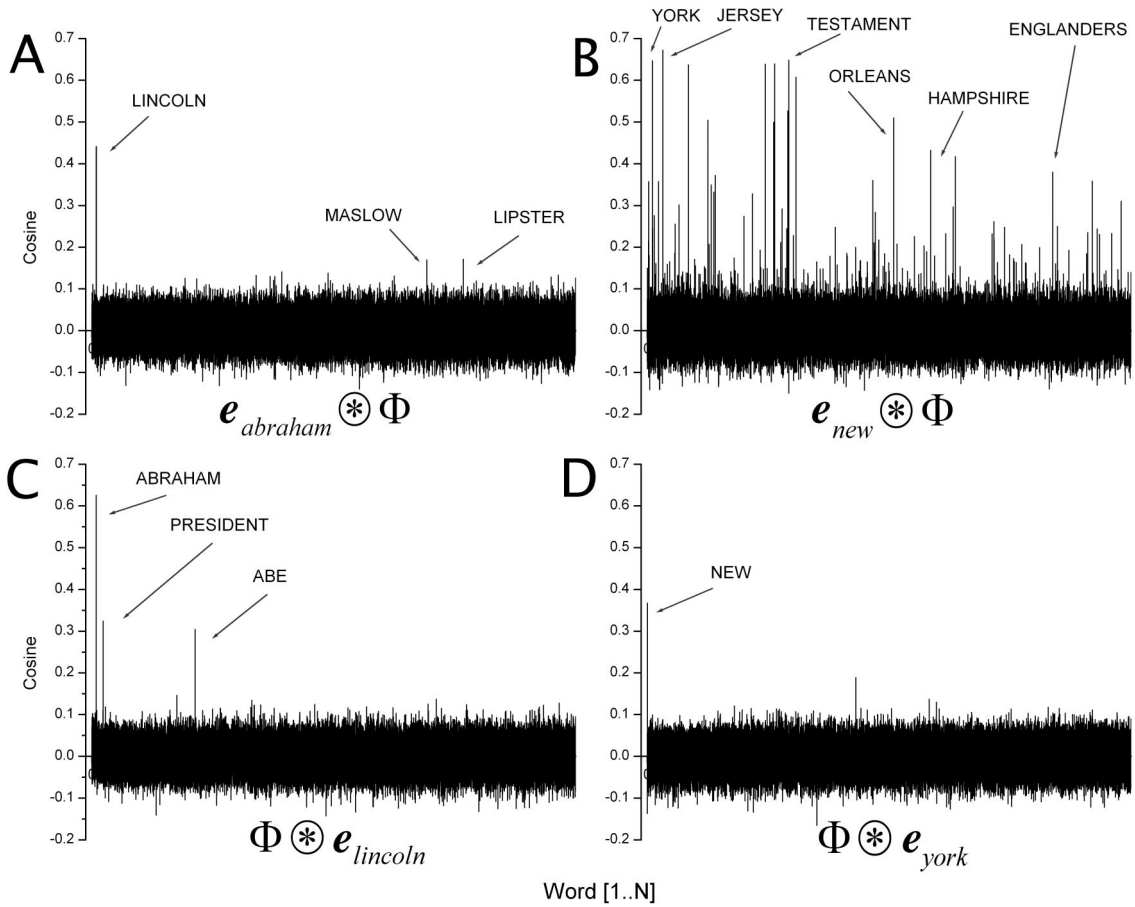


*Figure 7.* The state of the lexicon when probed with a vector: succeeding *abraham* (A), succeeding *new* (B), preceding *lincoln* (C), and preceding *york* (D).

the lexical vector for *bit*, $\boldsymbol{m}_{bit} = \boldsymbol{m}_{bit} + (\boldsymbol{e}_{dog} \circledast \boldsymbol{\Phi})$. Furthermore, the same bigram transition is simultaneously stored in *mailman* as part of a larger chunk ($\boldsymbol{e}_{dog} \circledast \boldsymbol{e}_{bit} \circledast \boldsymbol{e}_{the} \circledast \boldsymbol{\Phi}$). Transition information is distributed across lexical entries, and various lexical vectors may assist in retrieving order information when presented with a particular sequential pattern.

To determine which lexical entries are highly activated by a position in a sentence or fragment, the *n*-gram convolutions around the position are built and summed into a probe vector, as in learning (the position is replaced by the placeholder vector $\boldsymbol{\Phi}$). Lexical vectors that have been frequently found in the vacant position during learning resonate with the order vector for that position because many of the *n*-gram tokens for the vacant position also fit their order histories.

*Resonance* is the tendency of an object to absorb more energy when the probed frequency matches the object's natural frequency than it does at other frequencies. For example, when a tone is sounded by a piano, strings resonate depending on the match of their natural frequencies to the pitch of the tone. Resonance is used as a metaphor to describe the lexicon's response to a probe: Similar lexical vectors resonate with the order probe depending on the amount of shared variance they have with it. Holographic lexical vectors, however, may have several resonant frequencies coded within them, responding to several possible input patterns depending on the word's learned history.[10]

For example, when the vector $\boldsymbol{\Phi} \circledast \boldsymbol{e}_{thomas}$ (the words that preceded *thomas*) is compared with the lexicon, $\boldsymbol{m}_{dylan}$ is the most similar lexical vector (cos = .34), with all others responding with no more than random similarity. When the vector $\boldsymbol{e}_{thomas} \circledast \boldsymbol{\Phi}$ (words that succeeded *thomas*) is compared with the lexicon, however, many lexical vectors respond (*jefferson* [.52], *aquinas* [.43], *edison* [.38], etc.) because they all have the particular pattern coded into their representations, whereas only $\boldsymbol{m}_{dylan}$ has the pattern $\boldsymbol{\Phi} \circledast \boldsymbol{e}_{thomas}$ coded into its representation.

Figure 7 illustrates lexical resonance and the asymmetric nature of retrieval. Figure 7A shows the cosine of each lexical vector $\boldsymbol{m}_{1..N}$ in response to the probe vector $\boldsymbol{e}_{abraham} \circledast \boldsymbol{\Phi}$ (i.e., words that have been preceded by *abraham*). $\boldsymbol{m}_{lincoln}$ rises out of the lexicon in resonance with the probe vector; *maslow* and *lipster* also resonate, although less so. The other lexical representations remain

[10] Although it often has many other names (homophony, chorus of instances, etc.), resonance is a popular metaphor when describing memory retrieval (e.g., Brooks, 1978; Hintzman, 1984; Kwantes, 2005; Mewhort & Johns, 2005; Ratcliff, 1978; Semon, 1909/1923).

Table 4
*Top 10 Lexical Representations Resonating to a Binding Preceding or Succeeding a Target Word*

| KING | | | | PRESIDENT | | | | WAR | | | | SEA | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Φ** king | | king **Φ** | | **Φ** president | | president **Φ** | | **Φ** war | | war **Φ** | | **Φ** sea | | sea **Φ** | |
| luther | .33 | midas | .51 | vice | .46 | eisenhower | .40 | precivil | .43 | ii | .51 | sargasso | .37 | urchins | .50 |
| barbaric | .29 | lear | .47 | taftlike | .25 | kennedy | .36 | civil | .35 | veterans | .38 | caspian | .35 | gull | .49 |
| burger | .28 | henry | .47 | literalist | .24 | reagan | .31 | spanishamerican | .35 | material | .25 | aegean | .31 | anemone | .44 |
| seventyyearold | .25 | pellinore | .46 | poliocrippled | .18 | nixon | .28 | revolutionary | .29 | ended | .25 | mediterranean | .30 | otter | .39 |
| etruscan | .24 | minos | .45 | incumbent | .17 | carter | .28 | korean | .28 | chariots | .22 | baltic | .29 | anemones | .39 |
| legendary | .23 | farouk | .44 | twentyseventh | .17 | johnson | .27 | trojan | .28 | hawks | .21 | adriatic | .28 | otters | .36 |
| hanoverian | .22 | dionysius | .42 | thirtyfifth | .17 | truman | .25 | francoprussian | .28 | raged | .19 | sharkfilled | .27 | captains | .34 |
| thirteenthcentury | .20 | tut | .40 | democratic | .16 | lincoln | .25 | world | .27 | broke | .18 | aral | .25 | urchin | .34 |
| battlescarred | .19 | arthur | .31 | middleoftheroad | .15 | lyndon | .25 | vietnam | .26 | seesawed | .17 | carribean | .20 | lamprey | .31 |
| rex | .12 | agamemnon | .31 | overrode | .15 | roosevelt | .18 | declare | .25 | aims | .15 | barents | .19 | cucumber | .28 |

*Note.* Numbers following lexical representations are vector cosines. **Φ** represents the constant placeholder vector used during encoding.

only randomly excited by the probe vector. By contrast, Figure 7C shows the state of the lexicon in response to the probe vector $\mathbf{\Phi} \circledast \boldsymbol{e}_{lincoln}$ (i.e., words that have been succeeded by *lincoln*). Although *lincoln* responded as the predominant trace to follow *abraham* (see Figure 7A), *abraham* is not the only trace that precedes *lincoln*; *president* and *abe* are also excited by the pattern $\mathbf{\Phi} \circledast \boldsymbol{e}_{lincoln}$. Furthermore, $\boldsymbol{m}_{president}$ responds preceding a number of other words (*kennedy, carter,* etc.).

The asymmetric behavior is further illustrated in Figures 7B and 7D. In Figure 7D, only $\boldsymbol{m}_{new}$ resonates with the pattern preceding $\boldsymbol{e}_{york}$. By contrast, Figure 7B shows the state of the lexicon in response to the pattern succeeding $\boldsymbol{e}_{new}$. Although *new* was the only word that preceded *york,* many words respond that want to succeed *new, york* being among them. Table 4 further demonstrates the asymmetric nature of resonance in the lexicon with nearest neighbors (and their cosines) to sample target words.

Resonance may be used to probe the lexicon with longer and more general order patterns as well. For example, for the phrase *he* ____ *to*, the encoding vector $[(\boldsymbol{e}_{he} \circledast \mathbf{\Phi}) + (\mathbf{\Phi} \circledast \boldsymbol{e}_{to}) + (\boldsymbol{e}_{he} \circledast \mathbf{\Phi} \circledast \boldsymbol{e}_{to})]$ is compared with the lexical vectors $\boldsymbol{m}_{1..N}$, and each cosine is computed. Words that were frequently experienced in the blank position during learning, given the order context of the surrounding words, have a higher relative cosine with the order vector. The eight most highly activated words by the probe *he* ____ *to* are *refused* (.59), *went* (.57), *came* (.55), *seemed* (.53), *said* (.52), *had* (.52), *likes* (.51), and *wanted* (.50).

Table 5 shows the eight most highly activated lexical representations for various other phrases. The first column shows words highly excited by *he* ____. It is interesting to note that the verbs most highly activated by only the bigram information in *he* ____ are quite different from those in the constrained *he* ____ *to* even though the latter contains the same bigram that constitutes the former. Verbs that fit *he* ____ were found in that bigram with a high frequency; given bigram information only, simple verbs that have been found as successors to *he* and that have not been found in extremely variable contexts match. When *he* ____ *to* is presented, however, simple verbs that fit the bigram information in *he* ____ now do not fit the trigram because they are rarely found flanked by both *he* and *to.* The activation rank ordering of *went* increases from *he* ____ to *he* ____ *to* to *he* ____ *to the* because *went* progressively fits all of the *n*-gram convolutions as they are added.[11]

Generally, the more *n*-grams in a probe that fit a word's history of order information, the more constrained the model's response is. For words that do not appear in many different contexts during learning, it is possible to retrieve the exact word that was learned in a particular probe. Table 6 shows exact phrases learned from the TASA corpus. The italicized word (e.g., *brainstem*) was removed from the probe, and its position was predicted from the lexicon using only order information; the top six words activated by the various probes are displayed.

For example, "The brainstem is much larger and more complex than the spinal cord" is an exact sentence learned from TASA. *Brainstem* is obviously not the most likely response to follow *the* when only the bigram association is considered even though it is a possible response to follow *the*. As more surrounding words are added around the blank position, the candidate words with higher relative activations become more constrained. When the entire sentence is presented, *brainstem* is the best fit to all the *n*-grams in the probe, and there is considerable difference between the most highly activated word, *brainstem* (.37), and the other words. Additional *n*-grams can add signal to appropriately fitting words and noise to inappropriate ones. It is important to note that this retrieval demonstration considers only order information, but taking advantage of context information (meaning in the sentences independent of order) can further constrain the responses.

For more frequent words that have been found in many contexts, the exact word cannot be retrieved even when presented with an exact sentence in which it was learned. However, the most highly activated words become more plausible. In the full phrases of Table 6, few candidate words other than the target plausibly fit the probe, and the correct target is the best fit. Table 7 shows some exact phrases from TASA in which the correct target cannot be

[11] Note that the activation for a word does not necessarily increase linearly as *n*-grams may be fit (as with *went*). A word's activation may actually decrease with additional appropriate *n*-grams even though its activation rank order may increase. The additional *n*-grams may add noise, but they add less noise to appropriately fitting words and more noise to words that do not fit the positional information.

Table 5
*Eight Most Highly Activated Words in Various Blank Positions*

| he **Φ** | | he **Φ** to the | | he **Φ** not | | the **Φ** is | |
|---|---|---|---|---|---|---|---|
| said | .73 | went | .59 | did | .64 | latter | .58 |
| had | .61 | came | .56 | could | .54 | world | .56 |
| asked | .59 | gestured | .50 | does | .52 | worry | .55 |
| was | .56 | clung | .49 | had | .45 | problem | .54 |
| exclaimed | .52 | said | .48 | dared | .43 | sun | .53 |
| went | .51 | presented | .47 | was | .35 | cerebrum | .53 |
| cried | .50 | listened | .45 | is | .34 | earth | .53 |
| kept | .50 | belonged | .44 | should | .33 | truth | .52 |

*Note.* Numbers following activated words are vector cosines. **Φ** represents the constant placeholder vector used during encoding.

retrieved; the most highly activated words are the lexicon's best attempt to fit the position given that the exact instance in memory has been lost because of averaging.

When a word has been experienced only a small number of times, it is possible to retrieve exact order instances in which it was learned. As a word is experienced more frequently in different contexts, however, ability to retrieve exact instances diminishes, but the abstracted representation become more generalizable because of averaging. Lexical representations undergo an instance-to-abstraction transformation in memory with experience. This does, however, increase the reliance on lower order *n*-grams for more frequent words. For example, the final two probes in Table 7 demonstrate that BEAGLE can make the error of overemphasizing lower order chunks with order retrieval, as *n*-gram models do. The responses of *associate* and *interact* in the *ball* sentence are driven by the lower order *n*-grams and miss information about the object being an inanimate noun. Augmenting order retrieval with position-independent semantic information can help to ameliorate this bias.

*Retrieving with both decoding and resonance.* When an order probe is presented, information about unknown positions may be estimated from the surrounding context of known words and from learned experience of word transitions. The unknown information may be decoded from the learned history of the words present in the probe. Simultaneously, the transitions in the probe may be compared with the entire lexicon to determine which words have

Table 6
*Six Most Highly Activated Words in a Position as the Context of Neighboring Words Is Expanded*

| Phrase | Activations |
|---|---|
| the *[brainstem]* | first (.95) best (.93) latter (.92) next (.90) same (.89) following (.94) |
| the *[brainstem]* is | latter (.58) following (.54) same (.54) first (.53) world (.53) best (.53) |
| the *[brainstem]* is much larger and more complex than the spinal cord | brainstem (.37) latter (.22) world (.20) epidermis (.19) following (.19) same (.18) |
| although *[ostriches]* | arguably (.26) profit (.20) single (.17) double (.17) moderate (.17) thorough (.16) |
| although *[ostriches]* cannot | wastefulness (.18) quasars (.17) prison (.16) prayer (.15) buckler (.14) diastase (.14) |
| although *[ostriches]* cannot fly they have other skills | ostriches (.24) diastase (.13) wastefulness (.13) consecutive (.12) keen (.12) buckler (.12) |
| electric *[eel]* | shocks (.33) motors (.33) current (.32) charges (.25) sparks (.23) generators (.18) |
| an electric *[eel]* | current (.36) shock (.25) generator (.23) charges (.23) swiftness (.20) eel (.20) |
| an electric *[eel]* can produce a discharge of several hundred volts | eel (.19) painters (.12) flexplate (.12) methanol (.12) current (.12) pcp (.12) |
| emperor *[penguins]* | mutsuhito (.39) trajan (.31) justinian (.29) honorius (.26) yuan (.20) claudius (.19) |
| *[penguins]* have | archaeologists (.45) scientists (.42) we (.40) biologists (.38) geologists (.36) researchers (.34) |
| the emperor *[penguins]* have come to their breeding grounds | penguins (.24) trajan (.14) would (.13) leninists (.13) could (.12) researchers (.12) |

*Note.* The italicized word is the correct response for this chunk. Numbers in parentheses following activated words are vector cosines.

Table 7
*Predicting Words Found in Many Contexts*

| Phrase | Activations |
|---|---|
| The penguin *[does]* not fly | does (.30) did (.28) could (.23) dared (.17) shalt (.16) would (.16) |
| I have to *[run]* now | go (.36) get (.36) work (.35) make (.33) eat (.31) move (.28) |
| I *[have]* to run now | ought (.43) want (.35) intend (.33) have (.30) wish (.30) hate (.30) |
| No one would be allowed to *[interfere]* with the ball | cope (.37) comply (.35) grapple (.33) associate (.32) interact (.28) interfere (.27) |
| Kim sat up in *[bed]* looking sad | vain (.24) amazement (.23) unison (.22) desperation (.21) astonishment (.20) disgust (.19) |

*Note.* The italicized word is the correct response for this chunk. Numbers in parentheses following activated words are vector cosines.

been found in the order context during learning (resonance). Depending on the range of transitions a word has been found in during learning, either decoding or resonance may return a more salient signal. The two retrieval processes complement each other; they frequently agree on candidate words but may disagree.

For example, consider the probe *thomas* _____. Decoding from $\boldsymbol{m}_{thomas}$ with correlation retrieves a vector that is very similar to $\boldsymbol{e}_{jefferson}$ (cos = .59) and mildly similar to $\boldsymbol{e}_{edison}$ (cos = .16), but no more than randomly similar to other environment vectors. $\boldsymbol{m}_{thomas}$ has the transition *thomas jefferson* packed into it so frequently that it is difficult to retrieve anything but $\boldsymbol{e}_{jefferson}$ from it. However, *thomas* preceded many other words as well, and the pattern $\boldsymbol{e}_{thomas} \circledast \boldsymbol{\Phi}$ is stored in many vectors in the lexicon. When the pattern $\boldsymbol{e}_{thomas} \circledast \boldsymbol{\Phi}$ is compared with the lexicon, many lexical vectors resonate with it, the highest ones being *jefferson* (.74), *aquinas* (.58), *hickey* (.55), *edison* (.55), *pickney* (.50), *alva* (.45), *wolfe* (.40), *wentworth* (.40), *paine* (.36), *malthus* (.34), and *hobbes* (.31), among others. Simply averaging the cosine of the decoded vector with an environmental vector and the cosine of the encoded environmental pattern with a lexical vector (i.e., averaging decoding and resonance) provides a good measure of a word's overall activation to the presented order probe.

Let $\boldsymbol{r}$ be the vector decoded from $\boldsymbol{m}_{thomas}$, and $\boldsymbol{p}$ be the vector representing the probe string, $\boldsymbol{e}_{thomas} \circledast \boldsymbol{\Phi}$. Averaging the cosines of the decoded vector with the environmental vectors and the probe vector with the lexical vectors produces a measure of a word's activation, $\boldsymbol{a}_i$, to the presented order probe:

$$\boldsymbol{a}_i = \frac{\cos(\boldsymbol{r},\boldsymbol{e}_i) + \cos(\boldsymbol{p},\boldsymbol{m}_i)}{2}.$$

If both resonance and decoding agree on a word, the word's activation is emphasized (positively or negatively); otherwise, the two retrieval sources compete.

Consider the probe ____ *aquinas.* Again, because $\boldsymbol{m}_{thomas}$ has the bigram *thomas jefferson* coded into it with such a high frequency, it does not resonate highly with $\boldsymbol{\Phi} \circledast \boldsymbol{e}_{aquinas}$ without further context. However, because $\boldsymbol{m}_{aquinas}$ has a narrow range of transition information coded into its representation, decoding retrieves a vector highly similar to $\boldsymbol{e}_{thomas}$ (cos = .48) and only randomly similar to the other environmental vectors. $\boldsymbol{e}_{thomas}$ may also be uniquely decoded from $\boldsymbol{m}_{edison}$, $\boldsymbol{m}_{malthus}$, $\boldsymbol{m}_{paine}$, and so on, as well.

Because resonance with $\boldsymbol{\Phi} \circledast \boldsymbol{e}_{aquinas}$ does not reliably resemble any vectors above chance but decoding does, summing the two sources produces only a response of *thomas* (predominantly from decoding). By contrast, because decoding from *thomas* ____ produces only *jefferson* but resonance retrieves many other traces, summing the two sources results in a list of candidates in terms of their activation (*jefferson, aquinas, edison,* etc.). Both sources are needed to retrieve order information, particularly in more variable contexts (e.g., predicting parts of speech rather than names).

*Complementing order retrieval with context information.* Context information may be used to further disambiguate transition information in a probe given shared meaning between the probe and lexical vectors. For example, consider two probes: (a) *he ____ to the highway* and (b) *he ____ to the audience*. Assume that only the sequence *he ____ to* has been learned as a stable trigram, equally activating *went, drove,* and *spoke,* but that the full sequences of (a) and (b) are novel (i.e., *audience* and *highway* only introduce noise in order retrieval). Including context information in the probe emphasizes *went* and *drove* if *highway* is present and *spoke* if *audience* is present because they have shared meaning. However, *highway* interferes with *spoke,* and *audience* interferes with *went* and *drove*. Context information emphasizes semantic relationships irrespective of matching order instances.[12]

For example, following *thomas,* several words are highly activated if using only order information, with *jefferson* being the strongest. The predominance of the bigram *thomas jefferson* may be overridden by additional context. Table 8 presents the activation associated with six legal completions to *thomas* ____ and how the activations change as the sentence context is modified. Even though *jefferson* is the most likely word to follow *thomas* in general, *edison* becomes the most likely word in the context *thomas ____ made the first phonograph*.

## *Establishing Model Performance*

BEAGLE's learning of context and order information together produces a representation that improves on limitations of semantic space models. Furthermore, BEAGLE's representations are gradually acquired from statistical sampling of words across experi-

[12] Alternatively, context information can be calculated using the learned memory representations.

Table 8
*An Example of Changing Word Activation With Both Context and Order Information*

| Probe | jefferson | edison | aquinas | paine | pickney | malthus |
|---|---|---|---|---|---|---|
| Thomas ____ | **.72** | .66 | .60 | .35 | .46 | .34 |
| Thomas ____ wrote the Declaration of Independence | **.44** | .30 | .24 | .29 | .17 | .14 |
| Thomas ____ made the first phonograph | .33 | **.45** | .29 | .17 | .21 | .12 |
| Thomas ____ taught that all civil authority comes from God | .30 | .26 | **.40** | .13 | .17 | .12 |
| Thomas ____ is the author of *Common Sense* | .29 | .21 | .19 | **.43** | .18 | .13 |
| A treaty was drawn up by the American diplomat Thomas ____ | .32 | .26 | .27 | .17 | **.92** | .15 |
| Thomas ____ wrote that the human population increased faster than the food supply | .23 | .22 | .21 | .12 | .14 | **.41** |

*Note.* Numbers are activation values for the target word to the blank position in the probe sequence. Boldface indicates the correct target to fit into that probe sequence.

ence. In addition, a great deal of word order information can be retrieved from the representations. Because of the common storage mechanism, it may prove to be unnecessary to build transition rules into higher order models of language comprehension if they use BEAGLE's holographic representations as input.

Before we proceed, it is important to demonstrate that the context information learned by BEAGLE allows the model to perform similarly to a model such as LSA on established tasks. Furthermore, it is important to determine whether the addition of order information to the representation interferes with the model's ability to use context information. A common performance evaluator for semantic models is the synonym section of the TOEFL (Landauer & Dumais, 1994, 1996, 1997; Turney, 2001). Each item consists of a target word and four alternative words; the task is to select the alternative that is most similar in meaning to the target. Accuracy on a set of 80 retired items was originally used by Landauer and Dumais (1997) as the criterion for dimensionality selection in LSA, and the TOEFL has since become a benchmark for measuring performance of semantic models.

We should note, however, that the TOEFL is a synonym test; hence, models of semantics need not necessarily perform at human levels without further mechanisms to identify synonymy. Nonetheless, comparing performance and target–alternative cosines between the models is a good start to establish correspondence.

We used the same 80 TOEFL items originally used by Landauer and Dumais (1997).[13] Both LSA and BEAGLE were trained on the same TASA corpus. All comparisons with LSA were in term space with 300 dimensions—this dimensionality is roughly optimal for performance on the TOEFL items. For BEAGLE, we used the standard 2,048-dimensional vectors. For each item, the alternative with the highest cosine to the target was selected as the response. If either the target or correct alternative was unknown, a score of .25 was assigned for the item (essentially a guess).

LSA correctly answered 55.31% of the items. Trained on only context information, BEAGLE was in close correspondence, correctly answering 55.60% of the items. Trained on both context and order information, BEAGLE did slightly better, correctly answering 57.81% of the items.

We computed Pearson correlation coefficients between all 320 target–alternative cosines for LSA, context-only BEAGLE, and composite BEAGLE. Considering their differences in both learning mechanisms and context (paragraph for LSA and sentence for BEAGLE), LSA and context-only BEAGLE were quite similar in their pairwise word similarities, $r(297) = .662$, $p < .001$. The pairwise word similarities between LSA and the composite BEAGLE were also positively correlated, although less so, $r(297) = .527$, $p < .001$ (note: $df = N - 2 -$ pairwise missing items). We tested the difference between these correlations with E. J. Williams's (1959) ratio for nonindependent correlations (see also Steiger, 1980). The correlation between LSA and context-only BEAGLE was significantly larger than the correlation between LSA and composite BEAGLE, $t(296) = -5.44$, $p < .001$.

The context information in BEAGLE reasonably resembles the information in the LSA vectors (in terms of TOEFL cosines). The addition of order information makes the BEAGLE representation less similar to the LSA representation than does context information alone. However, the composite representation performed slightly better on the TOEFL, whereas its similarity to LSA decreased, relative to the context-only representation. In addition to providing a model of sequential dependency, adding order information to BEAGLE may improve the fidelity of semantic representation.

Because of the limited number of comparisons in the TOEFL test, we conducted a much larger comparison between models using a database constructed by Maki, McKinley, and Thompson (2004). Maki et al. have constructed a database of nearly 50,000 pairs of words taken from the D. L. Nelson, McEvoy, and Schreiber (1998) norms. For each word pair, they computed semantic distance from WordNet (Fellbaum, 1998; Miller, 1990, 1999) and compared with classic studies of human similarity judgments. Maki et al. demonstrated that WordNet semantic distance is a better predictor of human judgments of semantic similarity than other independent semantic measures. We used semantic distance in WordNet as our standard against which to compare the models.

The Maki et al. (2004) database contains distance measures from WordNet and LSA cosines (in 419 dimensions[14]) for 49,559 word pairs. Using the same word pairs, we also computed cosines for LSA in 300 dimensions and for BEAGLE using context information only, order information only, or both context and order information. Table 9 presents the correlations between the mea-

[13] We thank Tom Landauer for the TOEFL items.

[14] The 419-dimensional ratings are included in the Maki et al. (2004) database and were independently computed by Jose Quesada at the University of Colorado at Boulder.

Table 9
*Correlations Between Semantic Space Models and Semantic Distance in WordNet*

| Variable | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1. WordNet | — | −.165 | −.158 | −.293 | −.242 | −.311 |
| 2. LSA (300 dimensions) | | — | .987 | .579 | .179 | .369 |
| 3. LSA (419 dimensions) | | | — | .569 | .174 | .364 |
| 4. B-Context | | | | — | .439 | .756 |
| 5. B-Order | | | | | — | .804 |
| 6. B-Composite | | | | | | — |

*Note.* B-context is BEAGLE trained on only context information. B-Order is BEAGLE trained on only order information. B-Composite is BEAGLE trained on both context and order information together. For all values, $p < .001$. LSA = latent semantic analysis; BEAGLE = bound encoding of the aggregate language environment.

sures. The WordNet values are distance measures; thus, similarity in the semantic space models should be negatively correlated with WordNet distance.

The 300-dimensional LSA solution was more highly correlated with WordNet than was the 419-dimensional solution reported by Maki et al. (2004), although other factors than dimensionality may have differed; we do not know the specifics of the 419-dimensional version. BEAGLE with only context information was more highly correlated with WordNet than was LSA (300 dimensions), $t(49,129) = -32.29$, $p < .001$, as was BEAGLE with order information only, $t(49,129) = -33.65$, $p < .001$ (using E. J. Williams's, 1959, ratio). The composite BEAGLE representation was more highly correlated with WordNet than was either context-only BEAGLE, $t(49,129) = -6.03$, $p < .001$, or order-only BEAGLE, $t(49,129) = -36.89$, $p < .001$. Thus, the composite representation provides a better correspondence to semantic distance measures than LSA or the context and order representations alone do.

Learning the context and order information into the same composite representation must produce some data loss due to compression. We computed a two-parameter regression model predicting WordNet distance from both context and order cosines to a one-parameter model using only cosines from the composite representation. The multiple correlation from the two-parameter regression ($r = .319$) accounted for less than one half of a percent more variance in WordNet semantic distances than did the one-parameter regression. This suggests that data loss due to composite storage of the context and order information in the lexicon is minimal.

In summary, we have adapted associative memory theory to learn semantic information using both order and context, without dimensional optimization—the result is a semantic space representation that incorporates order information and that can also function as a model of sequential dependency. The addition of order information to the context representation does not weaken the model's semantic representation; rather, it enriches the representation (based on WordNet semantic distance measures, but see also Jones, Kintsch, & Mewhort, 2006). In the next section, we compare the correspondence between the structure learned into BEAGLE's representations and data from humans on a variety of semantic tasks.

## Linking Representation to Human Data

The holographic representations learned by BEAGLE contain each word's history of contextual and order information. This section outlines a broad range of semantic and order phenomena that can be accounted for directly from the structure of these learned representations. The first subsection describes the similarity structure of clusters of words formed in the lexicon and focuses on semantic categorization, typicality, and the acquisition time course of these clusters. The second subsection focuses on predicting human response latency in various semantic priming tasks, and the third subsection applies the model to semantic constraint in stem completions. In many of these tasks, a process model is still needed to produce response latencies and errors. Although development of an appropriate process model is certainly no trivial task, the purpose of this section is to demonstrate that the BEAGLE vectors contain the similarity structure needed by such a process model.

### *The Structure of Semantic Categories*

Because of accumulation of common random vectors across experience, representations for words in common natural categories develop similar patterns of elements. The lexical representations come to cluster in a manner that is similar to the organization of semantic categories in human memory. A variety of semantic category effects and acquisition trends can be accounted for directly from the structure of these learned representations.

*Typicality.* Typical members of a semantic category can be processed more efficiently than atypical ones (Collins & Quillian, 1969; Rosch, 1973, 1975; E. E. Smith et al., 1974). Battig and Montague (1969) collected subjective responses to create empirical category clustering norms by asking subjects to generate exemplars for certain category labels. For example, *robin* was the most frequently produced exemplar for the bird category (produced by 85% of subjects), whereas *chicken* was the least frequently produced (produced by only 9% of subjects). Presumably, frequency of exemplar production reveals something about the semantic structure of the category of birds. Words that are listed more often as exemplars of a category are also verified faster as members of that category in sentence verification tasks (Rips, Shoben, & Smith, 1973). For example, "A robin is a bird" is verified almost 50 ms faster than "A chicken is a bird."

Typicality effects are easy to explain with feature list models but are difficult to accommodate with a semantic network. With a featural representation, less frequent category members have fewer overlapping features with their superordinate list than do more frequent members. To explain typicality with a semantic network representation, however, the pathways to less frequent members

would have to be made longer than those to more frequent members. In either case, typicality has to be manually built into the structure of the semantic representation, whereas BEAGLE's abstract representation is learned automatically from experience with text.

Rosch (1975) has suggested that when seeing a new bird, one may classify it by comparing to a prototypical "birdy-bird," such as a robin. That is, typical members of a category should be more similar to one another—they should cluster together more densely and closer to the center of the category. Hierarchical semantic structure is a natural characteristic of the vector representations learned by BEAGLE. Animals cluster distinctly from places or vehicles. Within animals, however, fish, birds, and snakes tend to cluster distinctly from one another as well. Category labels tend to be proximal to the exemplars of the categories they represent. Furthermore, the names for features of a category (e.g., *wings, beak*) also tend to be more proximal to exemplars of the categories they are features of.

Figures 8A and 8B show two-dimensional scaling plots for sports and vehicles, respectively. The smaller gray plots show expansions of the dense regions toward the center of each category. As the figures show, the more typical exemplars are positioned nearer to the algebraic center of the category (the gray regions). In Rosch's (1975) subjective rating norms, *football* was ranked number 1 (of 60) by subjects as a good example of a sport, and *basketball* was ranked number 3. *Curling,* on the other hand, was ranked 22.5 as a good example of a sport. The labels for each category (*sports* and *vehicle*) are also closer to the more typical exemplars. By contrast, the less typical exemplars of each category are sparsely distributed and further from (a) each other, (b) the typical exemplars, and (c) the category label. It is not difficult to see how such a representation could naturally produce a typicality effect: More typical members are more like one another, closer to the category center, and closer to the category label than are less typical ones.

As a concrete example of typicality, consider a classic experiment by Rosch (1975, Experiment 2). Rosch primed subjects with a category label and then simultaneously presented two exemplars; the response was simply a same–different category judgment. Typicality of the exemplars, defined by subjective ratings, was varied. The basic findings were that same responses are faster if primed by the category name and that both same and different responses are faster to typical category members than to atypical ones (e.g., *robin–sparrow* is faster than *chicken–peacock*, and *robin–car* is faster than *chicken–tractor*). More recent neurocognitive research has cross-validated the similarity of neural states implied by Rosch's typicality response time (RT) data (Kounios & Holcomb, 1992).

Figure 9 illustrates Rosch's (1975) RT data (11 categories) for same responses plotted against BEAGLE's predictions based on euclidean distance to both the category prototype and the category label using the same stimulus pairs.[15] For the distance-to-prototype simulation (open circles in Figure 9), the center of each category was computed as the mean of all example vectors for the category, and the mean distance of exemplars to the prototype in each typicality bin was computed. For the category label simulations (open triangles in Figure 9), the mean distance of the examples in each category to the label was computed. Distance (to either prototype or category label) decreased as a linear function of typicality both for distance to prototype, $F(1, 35) = 7.17$, $p < .05$, and for distance to label, $F(1, 35) = 8.77$, $p < .01$.

Figure 9 shows that the representations learned by BEAGLE emulate the basic RT findings of Rosch (1975) for two reasons. If, as Rosch has argued, semantic categories are represented by either algebraic prototypes (a birdy-bird) or a standard exemplar (*robin*), typicality need not be built artificially into either a model's representation or its processing mechanisms. Alternatively, if the semantic categories are defined by their labels, the representations learned by BEAGLE will also produce a typicality effect. The structure is particularly interesting given that BEAGLE does not have the benefit of perceptual knowledge of birds or fruits that the human subjects presumably did. Simple learning routines applied to large-scale redundancies in language are sufficient to produce semantic typicality effects.

*Labeling categories and exemplars.* The typicality data imply that exemplars of a natural category may be classified simply on the basis of proximity to their appropriate label in the BEAGLE space. For example, *color* has a high cosine with *blue, red,* and *yellow.* To test this notion, 14 exemplar words were selected from each of 11 natural categories (the exemplars are presented in Appendix B).[20] Each of the 154 exemplars was classified by assigning it to the closest of the 11 category labels in the BEAGLE space (i.e., the label vector with the smallest euclidean distance to the exemplar vector). The results are presented in Table 10.

Chance would dictate that only 1.27 of the 14 exemplars in each category would be correctly labeled (9%). As Table 10 shows, however, the simple label-proximity algorithm classified exemplars much better than chance would predict. VEGETABLES was the only category that was not classified significantly better than chance. *Tomato* was omitted from the list of fruits because it was incorrectly labeled as a vegetable from contextual experience (a common human mistake; a tomato is technically a fruit). Furthermore, many of the vegetables used produce flowers, hence the similarity to the FLOWER label. Across textual experience, flowers, fruits, and vegetables are used quite similarity.

Of particular interest is the pattern of error responses when an exemplar was misclassified. The final column of Table 10 shows the incorrectly labeled exemplars and the labels (in order) that were closer than the correct label to each exemplar. In most error responses, the second closest label was the correct one. Error responses, however, were certainly not random: *Orange* (as a fruit) was misclassified as a COLOR, for example, and the incorrect dog exemplars were labeled BIRD (another animal, and DOG was always the second choice). It is difficult to define exactly how one would discriminate a label word from an exemplar or feature word in the lexicon. If the model were to know which vectors are exemplars and which are labels, however, the structure of the learned representations might be sufficient to classify exemplars into categories based on the nearest category label.

*Behavior of features in the BEAGLE space.* Feature list representations were initially appealing because typical exemplars shared more semantic features with the category prototype than did atypical ones, and certain common features became very diagnos-

[15] Two words were replaced by equal typicality exemplars (based on the subjective norms) because they did not have lexical entries in BEAGLE—*artichoke* was replaced with *cabbage,* and *boysenberry* was replaced with *raisin*. In addition, to avoid polysemy problems, *saw* (as a tool) was replaced by *screwdriver* in the simulations.

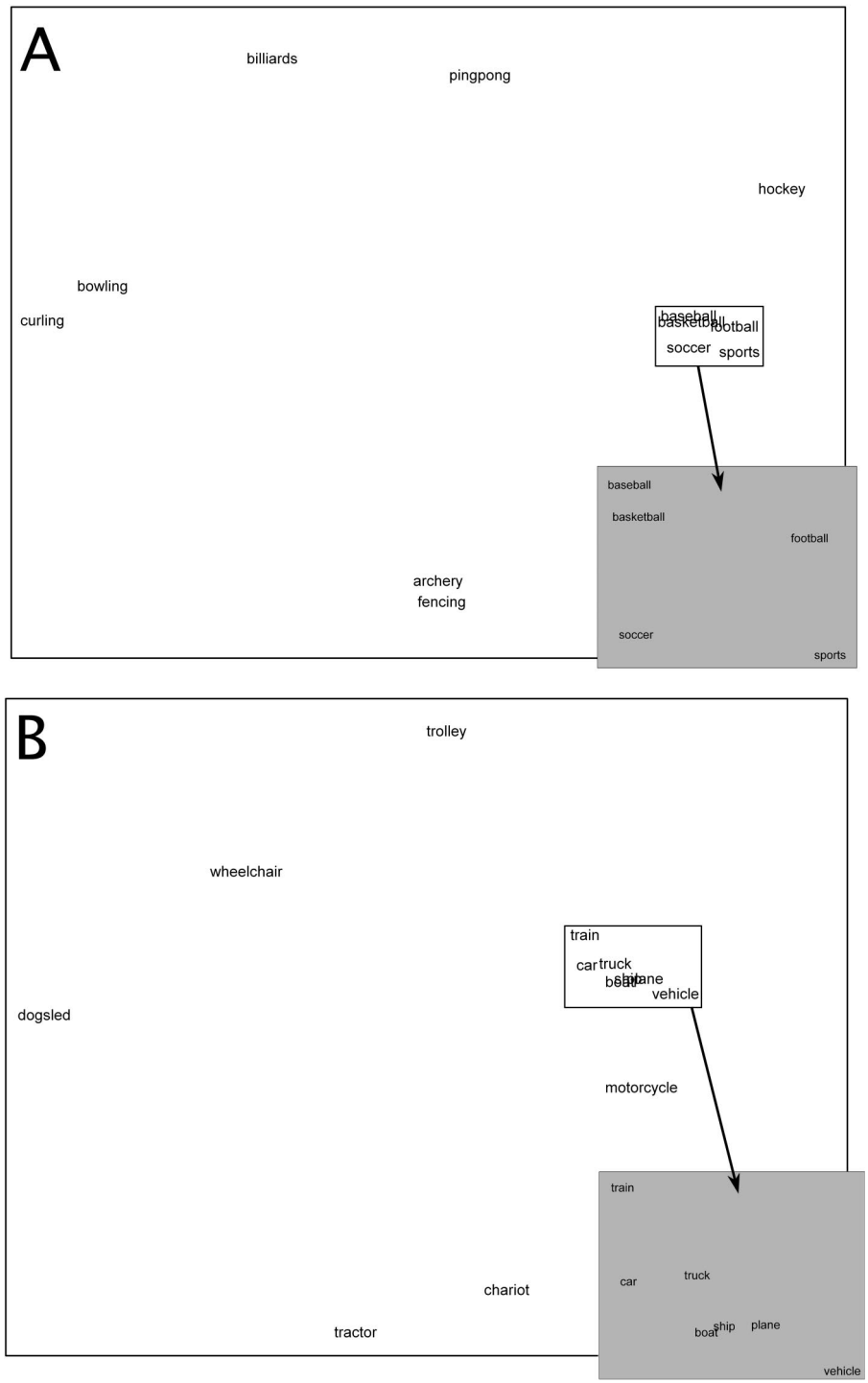


*Figure 8.* Examples of typicality from proximity of learned representations in the composite BEAGLE space. A: The structure of sports. B: The structure of vehicles. BEAGLE = bound encoding of the aggregate language environment.

tic of category membership (e.g., *wings, beak*) relative to less unique features. Furthermore, words that have a greater number of descriptive features tend to have faster lexical decision and naming times (Pexman, Holyk, & Monfils, 2003; Pexman, Lupker, & Hino, 2002).

In BEAGLE, typical exemplars of a category are typical because their statistical distributions are similar, resulting in a representation that puts them closer to the category center and the category label. The behavior of exemplar words and feature words in language naturally produces a representation in which typical

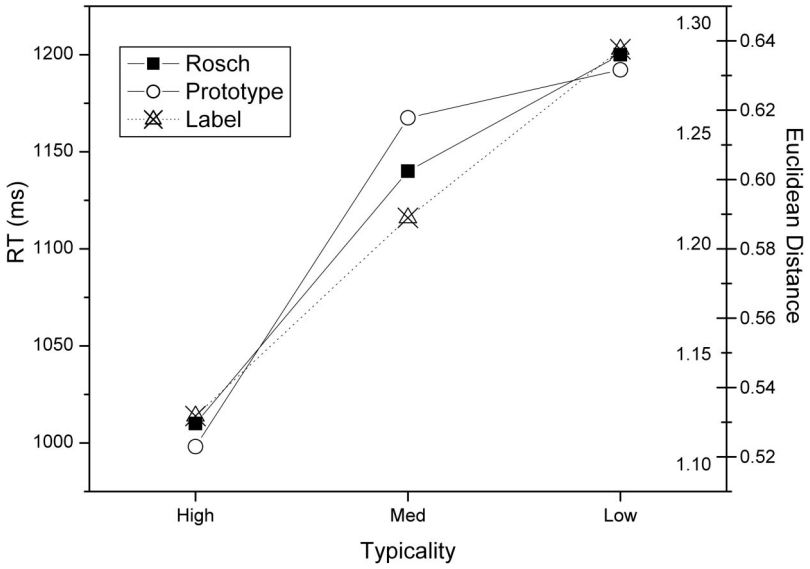


*Figure 9.* Rosch's (1975, Experiment 2) typicality data with predictions based on euclidean distance in the BEAGLE space. The dark squares represent Rosch's response time (RT) data from same responses and correspond to the scale on the left of the figure. The open circles represent the distance to prototype in the BEAGLE representations and correspond to the outside right scale on the figure. The crossed-triangles represent the distance to label in the BEAGLE representations and correspond to the inside right scale on the figure. BEAGLE = bound encoding of the aggregate language environment. Med = medium.

exemplars are more similar to category features than are the less typical ones. For example, near neighbors to VEHICLE are words representing features descriptive of the category, such as *wheels, driver, headlights, brakes, tires, seat,* and *engine*. These features are more proximal to a typical vehicle exemplar like *car* than they are to an atypical exemplar like *sled* as a product of the statistical distribution of the words in language.

To illustrate the similarity of semantic features to the categories of which they are descriptive, four feature words were selected for each of six categories from the category labeling simulation; the words are listed in Table 11. Category feature words were compared with both the label and exemplars of their category compared with other category labels and exemplars.

For the category label comparison, the euclidean distance was computed between each feature and its correct label (e.g., *wings, beak, fly,* and *feathers* to BIRD) compared with the euclidean distance between the same features and the other category labels (e.g., *wings, beak, fly,* and *feathers* to DOG, FISH, SPORT, VEHICLE, and DISEASE). Semantic features were significantly closer to their own category label ($M = 2.31$) than to the labels of the other five categories ($M = 2.78$), $F(1, 142) = 17.09$, $p < .001$.

Table 10
*Accuracy of Labeling Exemplars Using the Most Proximal Label Word for 11 Categories and 14 Exemplars Per Category*

| Category | Accuracy (%) | *t* | Errors |
|---|---|---|---|
| COLORS | 86 | 7.90* | *purple* = FLOWER; *green* = FLOWER, BIRD, FISH |
| SPORTS | 93 | 11.74* | *football* = BIRD |
| FISH | 100 | — | |
| BIRDS | 100 | — | |
| DOGS | 79 | 6.11* | *beagle, bulldog, foxhound* = BIRD |
| CITIES | 93 | 11.74* | *berlin* = COUNTRY |
| COUNTRIES | 100 | — | |
| FLOWERS | 79 | 6.11* | *daisy, daffodil, tulip* = BIRD |
| FRUITS | 71 | 4.99* | *orange* = COLOR; *grapefruit* = VEGETABLE; *strawberry* = FLOWER; *pear* = BIRD |
| VEGETABLES | 21 | 1.09 | *broccoli, parsley* = FRUIT; *cauliflower, radish, rhubarb* = FLOWER; *peas, eggplant, asparagus, onions, leeks* = FRUIT, FLOWER |
| DISEASES | 100 | — | |

*Note.* *t* values are one-sample *t* tests from chance ($df = 13$; $\mu = 9\%$). The Errors column lists exemplars that were incorrectly labeled and the labels (in order) that were closer to the exemplar than its correct label (exemplars are lowercase, labels uppercase).
* $p < .001$.

Table 11
*Semantic Features for Each Category Used in the Distance-to-Prototype and Distance-to-Label Simulations*

| Category | Features |
|---|---|
| BIRD | wings, beak, fly, feathers |
| DOG | bark, fetch, pet, fur |
| FISH | swim, gills, scales, fins |
| SPORT | players, crowd, score, skill |
| VEHICLE | wheel, driver, motor, brakes |
| DISEASE | symptoms, treatment, pain, sickness |

For the exemplar comparison, the euclidean distance was computed between each feature and the 14 exemplars of its category listed in Appendix B (e.g., *wings, beak, fly,* and *feathers* to *robin, sparrow,* etc.) compared with the euclidean distance between the same features and the 14 exemplars for each of the five other categories. Semantic feature words were significantly closer to exemplars of which they were descriptive ($M = 3.08$) than they were to exemplars from the other categories ($M = 3.43$), $F(1, 2014) = 160.85$, $p < .001$.

Semantic feature words may well be descriptive of their categories in that they are similar to members of the category from contextual experience. However, they need not be an inherent part of the semantic representation. BEAGLE demonstrates that a distributed representation, learned from simple mechanisms applied to statistical redundancies in language, can represent exemplars, labels, and features with the same abstract representation.

*Acquisition of semantic and lexical categories.* In the previous section, the learned lexical representations displayed cohesive structure between exemplars of a semantic category. However, this structure must be learned across many sentences in the corpus. Because the lexical representations begin from random vectors and gradually form structure from statistical sampling of words in the corpus, development of semantic structure in the lexicon across time can be studied.

For example, Figure 10 shows two-dimensional scaling plots for five exemplars of COUNTRIES, FISH, and COLORS. Figure 10A shows similarity between representations with only 1,000 sentences learned; structure between the representations early in learning is still close to random. By contrast, Figure 10B shows similarity between the same representations after the entire corpus has been learned; clearly, cohesion for the semantic groups has increased as noise is averaged out, and some categories are more cohesive than others.

Because dimensionality in BEAGLE is fixed and only the pattern of elements changes across learning, development of category cohesion can be studied continuously as a function of experience (i.e., the time course of movement from Figure 10A to Figure 10B). Figure 11 shows mean interexemplar cosine across learning for the 14 exemplars of COLORS, NUMBERS, FISH, and COUNTRIES listed in Appendix B. To avoid any potential artifact due to sentence/document order in the TASA corpus, sentences were sampled randomly without replacement. Thus, Figure 11 displays group cohesion as a function of progressive sampling.

In general, semantic category cohesion tends to develop as an exponential function of learning; however, some categories gain cohesion faster than others and ultimately to a greater extent. The rate of category cohesion development depends on both frequency and contextual variability of exemplar words. Figure 11 also illustrates that both words that have articles dedicated to them (e.g., COUNTRIES) and words that are defined simply by their usage across the corpus (e.g., NUMBERS and COLORS) can form cohesive categories.

Examining continuous learning can be particularly difficult with classic types of knowledge representation (e.g., semantic networks and feature lists) as well as contemporary semantic space models. In BEAGLE, dimensionality is fixed across learning, and only the pattern of vector elements changes with learning; hence, continuous development is a natural characteristic of a random accumulation model. More recent work with semantic networks has examined growth as a function of learning (e.g., Steyvers & Tenenbaum, 2005), but a comparison of modern distributed and localist models of semantic development is beyond the scope of this article.

Development of structure in the lexicon is not limited to semantic categories. For example, many studies have examined the differences between abstract and concrete words in various tasks (see Neath, 1997, for a review). Generally, concrete words (e.g., *dog*) are easier to process and recall than are abstract words (e.g., *justice*). In addition, concrete words normally have an earlier age of acquisition than do abstract words (McFalls, Schwanenflugel, & Stahl, 1996).

Paivio's (1971, 1986) dual-coding theory proposes that words representing concrete objects can be encoded using both verbal and visual codes (e.g., the word *dog* and an image of a dog), whereas abstract words have no visual code (see Pylyshyn, 1973, 1981, for criticisms of Paivio's theory). In BEAGLE, concrete and abstract words can become similar to other members of their class from indirect information.

Figure 12 shows the time course of group cohesion development in the lexicon for abstract and concrete words (80 words per group; sentences were sampled in random order). The abstract and concrete words used in this simulation were taken from Paivio, Yuille, and Madigan (1968); they were balanced for frequency, length, and age of acquisition, but the concrete words had a higher mean imageability rating and concreteness rating than did the abstract words (see Neath, 1998, Appendix Table I). In BEAGLE, the concrete words had a higher mean interword cosine (.6121) than did the abstract words (.2034). More importantly, Figure 12 shows that the acquisition trends for the two groups are very different: Both groups become progressively more cohesive, but concrete words have a much steeper slope of acquisition than do abstract words.

It is possible that the differences between abstract and concrete words have to do with their natural statistical variability in language. As a group, concrete words are more similar to one another and have denser semantic neighborhoods than do abstract words. Furthermore, the cohesion benefit for concrete words is learned faster than that for abstract words as a characteristic of statistical sampling.

Because of acquisition of order information, development of lexical categories can also be examined in the BEAGLE lexicon. As a general example, consider the broad finding that nouns are learned faster than verbs (Fleischman & Roy, 2005; Gentner, 1982; Snedeker & Gleitman, 2004). An 18-month-old child's

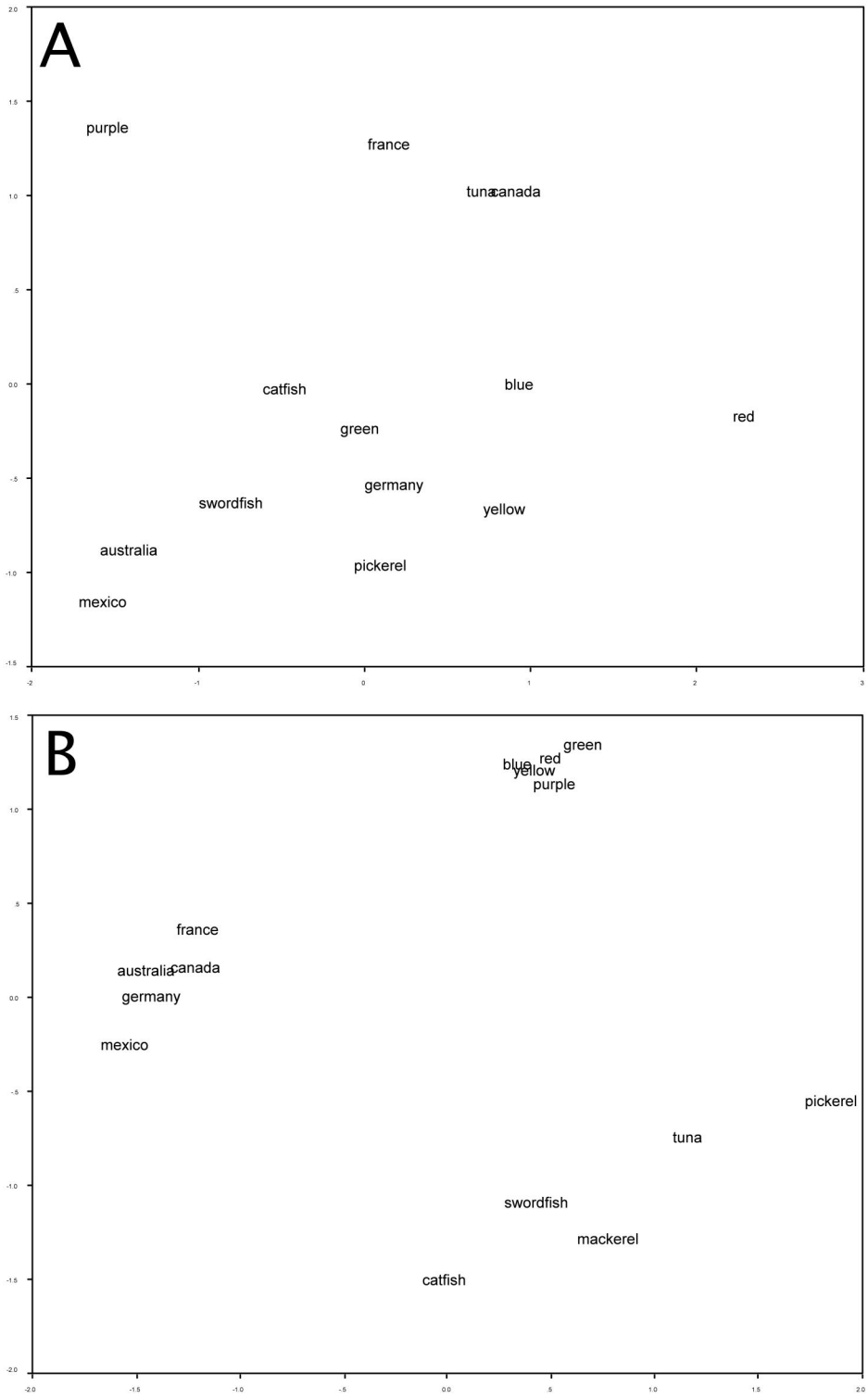


*Figure 10.* An example of semantic category development with exemplars of countries, colors, and fish. A: Structure between the exemplars with only 1,000 sentences sampled. B: Structure between the same exemplars when the entire text corpus has been learned.

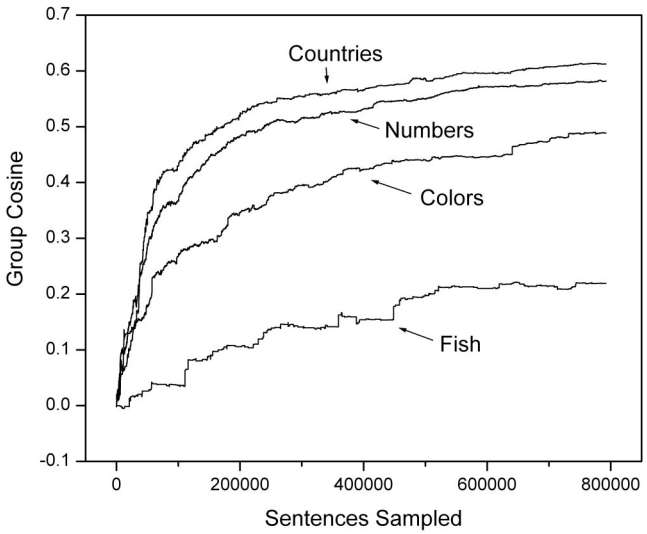


*Figure 11.* Development of semantic category cohesion in the lexicon as a function of progressive sentence sampling.

lexicon is composed of predominantly concrete nouns; verbs do not tend to stabilize until later even though the child's environment is usually split evenly between nouns and verbs (K. Nelson, 1974, 1981).

Why, given equal exposure to nouns and verbs, are nouns acquired first? Presently, the dominant hypothesis is *perceivability* (e.g., Gillette, Gleitman, Gleitman, & Lederer, 1999; Snedeker & Gleitman, 2004): Children use concrete nouns first because they represent concrete objects that may be physically manipulated in the child's environment. Recall from the abstract versus concrete simulation, however, that BEAGLE predicts concreteness from the structure of the language environment. In BEAGLE, the learning benefit for nouns over verbs reflects the structure learned from the statistical distribution of words in language.

Figure 13 shows cohesion of 20 nouns and 20 verbs (equated for frequency in TASA) as a function of sentences sampled (in random order). The words used are presented in Appendix C. Consistent with the empirical trend, the nouns are more cohesive as a group than the verbs, and this structure is learned faster for nouns than verbs. Obviously, the structure of sentences in TASA is different from the structure in a child's language environment. Nonetheless, the structure in the representations learned by BEAGLE is consistent with nouns as a group being learned faster than verbs.

Intuitively, it makes sense that perceivability must be a major factor influencing the benefit for learning nouns faster than verbs. However, it is possible that this finding is due in part to the statistical distribution of the two lexical classes across language; that is, acquisition trends may be partly explained as progressive sampling in a statistically redundant language environment. More research is needed to tease apart the effects of perceivability and large-scale statistical structure. In addition, the two sources of information are not completely separable—perceivability influences the statistical usage of words in language.

### *Priming*

The structure of semantic representation has been studied at great length using similarity priming techniques (for reviews, see McNamara, 2005; Neely, 1991). The common finding in priming is that a stimulus is processed more efficiently when it is preceded by a related stimulus. The assumption is that the first stimulus (the prime) facilitates processing of the second stimulus (the target) because they have shared information in their respective mental codes (Rosch, 1975). In semantic priming, the magnitude of facilitation depends on the semantic similarity between the prime and target. For example, *nurse* is processed more efficiently when preceded by *doctor* than when preceded by *bread* (Meyer & Schvaneveldt, 1971).

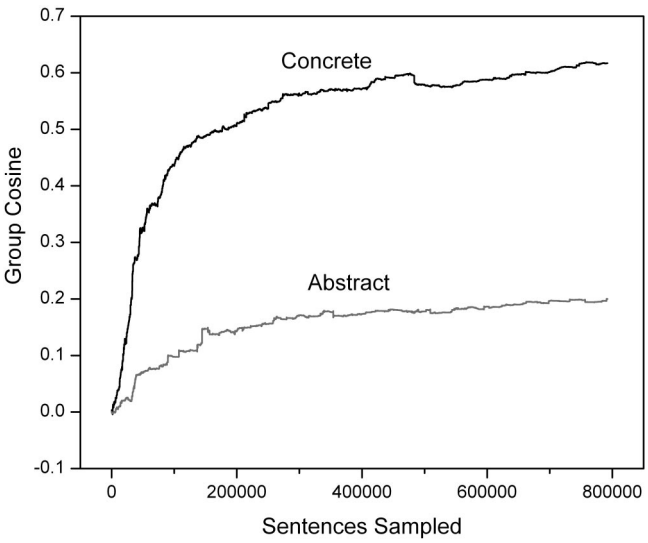


*Figure 12.* Development of cohesion within abstract and concrete words (from Paivio, Yuille, & Madigan, 1968) as a function of progressive sentence sampling.

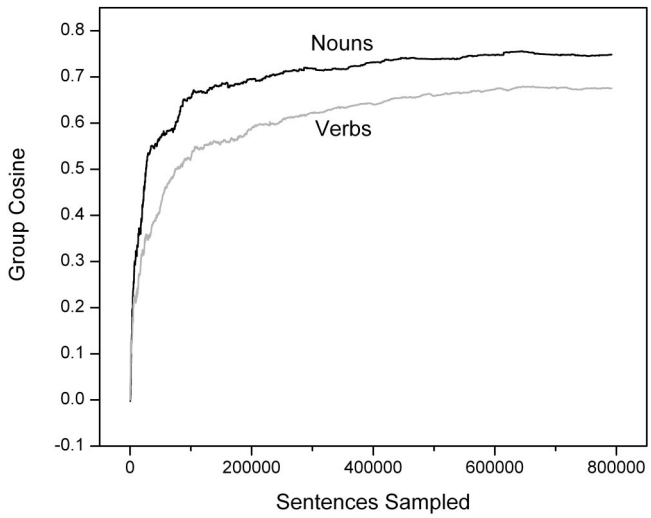


*Figure 13.* Development of lexical class cohesion in the lexicon as a function of progressive sentence sampling.

Table 12
*Priming Results From Chiarello et al. (1990) and Predictions Based on Lexical Similarity From the BEAGLE Representations*

| | Chiarello et al. (1990) | | | BEAGLE | | |
|---|---|---|---|---|---|---|
| Prime–target similarity | Unrelated | Related | Priming | Unrelated | Related | Priming |
| Semantic | 792 | 769 | 22 | .2392 | .4065 | −.1673 |
| Associated | 786 | 789 | −3 | .3267 | .4229 | −.0961 |
| Both | 798 | 770 | 28 | .2515 | .5113 | −.2598 |

*Note.* BEAGLE = bound encoding of the aggregate language environment.

Localist models account for semantic priming with the construct of spreading activation (Anderson & Bower, 1973; Collins & Loftus, 1975). When nodes in a network are activated, the activation spreads along the associated pathways to related nodes. The spread of activation makes the associated nodes already partially activated when a related concept is processed. Spreading activation is a crucial explanatory component of semantic networks (Balota & Lorch, 1986). In feature list representations, by contrast, semantic priming is accounted for by overlapping features between the prime and target. Whereas *robin* shares no features with *chair,* it has more shared features with *bat* and even more with *sparrow*.

It has been greatly debated whether semantic priming is due to strength of association or semantic overlap between the prime and target (Huchinson, 2003; Lucas, 2000; J. N. Williams, 1996). The aggregate results of many studies (see Huchinson, 2003) imply that priming exists for pairs that have either a semantic-only or an associated-only relationship. In addition, there exists an *associative boost* for prime–target pairs that have both types of relationships. It has been argued, however, that the semantic–associative distinction is a false dichotomy (e.g., Huchinson, 2003; McNamara, 2005; Steyvers, 2000); there are unlikely to be any purely associated or purely semantically related words.

In BEAGLE, facilitation should be directly predictable from structural similarity between lexical representations. Because the representations contain both semantic and associative information, overlap in either type should cause facilitation, and prime–target pairs that share both types of information should yield an associative boost. Unlike a binary feature representation, however, BEAGLE's abstract distributed representations allow shared information between all types of words (e.g., exemplars, category labels, and features: *robin, bird,* and *wings,* respectively). Furthermore, the indirectly learned contextual information affords similarity relations found in mediated priming.

*Semantic and associative priming.* As an example of simulating priming from the representations learned by BEAGLE, consider a study by Chiarello, Burgess, Richards, and Pollok (1990). They used prime–target pairs that had a relationship that was either semantic only (*deer–pony*), associative only (*bee–honey*), or both semantic and associative (*doctor–nurse*).[16] Chiarello et al. measured performance for presentations separately for each visual field; for simplicity here, we collapse across the visual fields.

Table 12 presents the latency data from Chiarello et al. (1990, Experiment 2: naming responses) and the corresponding cosines between the same words in BEAGLE. Note that a high cosine indicates greater similarity, which, in turn, predicts faster RTs in that condition relative to a condition with lower cosines. A negative priming difference in BEAGLE, thus, predicts positive facilitation in human data (i.e., the unrelated prime–target pairs are less similar than are the related pairs). Chiarello et al. found significant facilitation for semantic-only and semantic plus associated prime–target pairs.

In BEAGLE, the related prime–target pairs had significantly higher cosines than the unrelated pairs in the semantic-only condition, $F(1, 47) = 33.95$, $p < .001$; the associated-only condition, $F(1, 47) = 19.15$, $p < .001$; and the semantic plus associated condition, $F(1, 47) = 114.49$, $p < .001$. The magnitude of the facilitation predicted by BEAGLE differed across the three priming conditions, $F(2, 141) = 10.65$, $p < .001$. Student Newman-Keuls post hoc tests revealed that the differences between all three conditions were significant (i.e., all three groups are homogeneous subsets).

BEAGLE predicts Chiarello et al.'s (1990) findings of facilitation in the semantic-only and semantic plus associated conditions. Furthermore, facilitation was predicted to be greater in the semantic plus associated condition than in the semantic-only condition, as it was in the human data. BEAGLE does, however, predict a small but reliable facilitation effect in the associated-only condition, a finding not mirrored by the Chiarello et al. data. However, using the same stimuli, Chiarello et al. found facilitation in the associated-only condition in their Experiment 1 (lexical decision), and it has been reliably demonstrated in many other studies (see Huchinson, 2003).

To extend the demonstration of facilitation in associated-only prime–target pairs, predictions were generated from BEAGLE for an associated-only stimulus set used by Shelton and Martin (1992, Experiments 3 and 4). Shelton and Martin found robust facilitation priming for these stimuli, with faster responses when the target was preceded by an associated prime (*coffee–cup*: 517 ms) than when preceded by an unrelated prime (*mouse–cup*: 553 ms). Using the same prime–target pairs in BEAGLE, cosines between the target words and their associated primes ($M = .5887$) were significantly higher than between the targets and unrelated primes from the same list ($M = .2886$), $F(1, 35) = 126.35$, $p < .001$.

BEAGLE generally predicts weaker facilitation between prime–target pairs that share only an associative relationship (*bandage–doctor*) because associative information is learned only from shared context in the model. Semantic relationships (*lawyer–doctor*) tend to be stronger than purely associative relationships in the model, and pairs that have a relationship that is both semantic

[16] However, Steyvers (2000) has noted that many of the so-called semantic-only words in Chiarello et al.'s (1990) experiment were, in fact, often associates in the D. L. Nelson et al. (1998) norms.

and associative (*nurse–doctor*) are predicted to have an associative boost because their similarity is the result of multiple learning sources (local and global context and order information) and the sources are simply summed during learning. Note, however, that certain associative relationships can be more powerful than purely semantic ones.

*Mediated priming.* In a mediated priming task (Balota & Lorch, 1986), the relationship between the prime and target (e.g., *lion–stripes*) is through a mediated concept (e.g., *tiger*). Mediated priming often produces only a subtle effect on RT. However, even when it is not observed in RT data, mediated priming is still prominently observed in neurocognitive measures of brain function and is associated with the N400 component in evoked brain potential recordings (Chwilla & Kolk, 2000). McKoon and Ratcliff (1992) have suggested, however, that mediated priming may not be mediated at all but rather may reflect a weak associative relationship directly between the prime and target.

Mediated priming can be accounted for naturally in semantic networks by spreading activation but is difficult to explain by a feature list account. In a network representation, activation of a node spreads activation to connected nodes. Thus, when the *lion* node is activated, activation spreads to *tiger* because it is closely connected to *lion* and then to *stripes* because it is closely connected to *tiger*. In a feature list representation, however, *lion* and *stripes* cannot overlap (in fact, *stripes* should be a feature slot, not a distinct representation).

Mediated priming naturally occurs in BEAGLE representations without the need for a process like spreading activation. In BEAGLE, features (elements) are abstract values representing the distribution of samples across experience. No one feature has any meaning in isolation, but the word is represented by its complete pattern of elements (much like weights in a connectionist network). Even though *lion* and *stripes* may never directly co-occur in text, their representations have shared random vectors from the contexts in which they are found. *Lion* and *mane* become similar from shared context, as do *tiger* and *stripes*. Because *lion* and *tiger* become similar to each other, *mane* and *stripes* become similar to both and to each other even if they never directly co-occur. Thus, in a sense, BEAGLE supports McKoon and Ratcliff's (1992) notion that mediated priming is actually due to direct similarity between the prime and target. Unlike McKoon and Ratcliff's account, however, the two words need not be present simultaneously in short-term memory during encoding. The information that becomes shared between *lion* and *stripes* is learned from their statistical behavior in language, which is mediated through their respective relationships to *tiger*. In the learned representations, the similarity between *lion* and *stripes* is direct and exists even without a lexical entry for *tiger*.

For example, Balota and Lorch (1986) tested prime–target pairs that had either a direct relationship (e.g., *tiger–stripes*) or a relationship that was mediated through a concept related to both (e.g., *lion–stripes*). They used the same targets in all conditions but varied the primes.

Table 13 shows Balota and Lorch's (1986) RT results and corresponding predictions from BEAGLE using the same word pairs (unrelated primes were random re-pairings with another word from the related list, as Balota and Lorch, 1986, did; see their appendix). BEAGLE predicts the priming facilitation found in the human data. The related prime–target pairs had significantly higher cosines than the unrelated pairs, $F(1, 47) = 19.71$, $p < .001$, as did the mediated prime–target pairs, $F(1, 47) = 8.11$, $p < .001$. Furthermore, the predicted facilitation for the related condition (unrelated − related = −.1095) was significantly greater than the predicted facilitation for the mediated condition (unrelated − mediated = −.0644), $F(1, 47) = 7.73$, $p < .01$.

Table 13
*Priming Results From Balota and Lorch (1986) and Predictions Based on Lexical Similarity From the BEAGLE Representations*

| | Related | Mediated | Unrelated |
|---|---|---|---|
| Balota & Lorch (1986) | 549 | 558 | 575 |
| BEAGLE | .3647 | .3196 | .2653 |

*Note.* BEAGLE = bound encoding of the aggregate language environment.

The BEAGLE representations naturally predict mediated priming effects observed in human data without relying on a process construct such as spreading activation. In BEAGLE, the similarity between all prime–target pairs is direct and need not be mediated through another representation. Furthermore, the model predicts greater facilitation for prime–target pairs that have a direct semantic relationship than it does for mediated pairs,[17] another characteristic found in the human data.

### *Semantic Constraint in Stem Completions*

Stem and completion stimuli are commonly used to study semantic constraint and syntactic ambiguity resolution. The general finding is that the final word of a sentence is identified or judged more quickly when it is consistent with the meaning of the preceding sentence string (Fischler & Bloom, 1979; Schuberth & Eimas, 1977). The more specific the meaning of the stem is, the more constrained the list of candidate completion words will be. Although the precise mechanism or mechanisms of this effect are the subject of debate (Fischler & Bloom, 1985), in BEAGLE it is directly predictable from the fit between the context and order information in the stem and the lexicon. Lexical vectors that fit the meaning and order of the sentence more appropriately are more highly activated by presentation of the stem. To compute context information for a stem, we sum the learned lexical vectors for each word in the stem. To compute order information for the final position, we replace the position with the phi vector and convolve the environmental vectors for the words around this position.

For example, consider how well *boat* fits as a completion word in the following two stems (from Whittlesea, 1993):

1. She saved her money and bought the _____.
2. The stormy sea tossed the _____.

In BEAGLE, the context information from Stem 1 activates lexical vectors vaguely related to a number of things because the overall meaning of the stem is vague; some of the top neighbors are verbs

[17] The notion of mediation is now confusing because, in BEAGLE, even mediated relationships are actually due to directly shared statistical information.

such as *paid, purchased, earned, sold, spent,* and *received*. The target word *boat* is the 1,081st neighbor to the context vector for this sentence. Adding the order information (i.e., computing summed convolutions for the final position) generally constrains to words that also fit the grammaticality of the final position, which inhibits *paid, purchased,* and so on as likely candidates. The addition of order information increases *boat*'s activation over the above verbs and moves its rank order up to 244. Given the vagueness of the stem, however, few words are highly constrained by the context and order information. The terminal word could just as likely be *boat, car, house,* or a number of other concrete nouns.

The context vector for Stem 2, on the other hand, is more constrained and is similar to concepts like *wind, water, waves,* and *shore*. *Boat* is the 16th neighbor to the context vector for Stem 2. Adding the order information for the terminal position constrains the activations somewhat, moving *boat* up to the 10th neighbor. The lexical vector for *boat* has a cosine of .42 with the context and order vector for Stem 1 and a cosine of .56 for Stem 2. *Ship* fits even better than *boat* in Stem 2 and even worse in Stem 1.

In his study, Whittlesea (1993) examined the effect of processing fluency on a word-naming task with stem and completion stimuli similar to the above example. Specifically, he used 10 high- and 10 low-expectation stems (see his Appendix A) with different terminal words. The subject's task was to read the stem and press a button to reveal the completion word, reading it aloud as quickly as possible. Pronunciation latencies were faster to completion words preceded by a high-constraint stem (661 ms; e.g., Stem 2) than by a low-constraint stem (735 ms; e.g., Stem 1).

In BEAGLE, the context and order information for the same stems was computed for the terminal position, and the two types of information were summed into a composite probe vector. The cosine was computed between the probe vector and the lexical vector for the terminal word for each stimulus. The model predicts Whittlesea's (1993) finding of greater congruency between stem and completion if the context is semantically more predictive. Specifically, high-expectation stem and completions had significantly higher cosines ($M = .3517$) than did low-expectation stem and completions ($M = .2666$), $F(1, 18) = 6.30$, $p < .05$.

In a related study, Whittlesea (2002) used a larger stimulus set (see his appendix) that varied stem constraint while using the same completion words. An example of a high-constraint stem is "After the accident he was covered in . . . ," and an example of a low-constraint stem is "On the corner of the table there was a bit of . . . ," with *glass* and *blood* being legal completion words for either stem. Whittlesea normed these stimuli by having subjects rate whether the completion was predictable from the stem; the norming data are of interest here. Subjects rated the completion words as more predictable from the high-constraint stems (.81) than from the low-constraint stems (.27).

In BEAGLE, the context and order information was computed for the final position of the same 240 stems from Whittlesea's (2002) experiment, and the cosine was calculated between the stem vectors and their respective completion words. High-constraint stem and completions ($M = .3500$) were significantly more similar than low-constraint stem and completions ($M = .2751$), $F(1, 119) = 49.92$, $p < .001$.

In a similar study, Taraban and McClelland (1988, Experiment 1) used stem and completion stimuli to tease apart the effects of phrase attachment and expectation in stem completions. They designed sentences in which the terminal word was a noun-phrase attachment (NPA; "John ordered a pizza with *pepperoni*") or a verb-phrase attachment (VPA; "John ordered a pizza with *enthusiasm*"). According to the principle of minimal attachment (Frazier & Rayner, 1982), the NPA sentences should take longer to read than the VPA sentences because they have more constituent branches. However, Taraban and McClelland selected NPA words to be more expected from the stem than the VPA words, pitting expectation against minimal attachment.

Table 14
*Mean Expectation Ratings and Reading Times (ms) From Taraban and McClelland (1988, Experiment 1) and Stem-Completion Cosines From the BEAGLE Representations*

| | Taraban & McClelland (1988) | | BEAGLE |
|---|---|---|---|
| Attachment type | Expectation | Reading time | Cosine |
| NPA (expected) | 3.97 | 644 | .3758 |
| VPA (unexpected) | 2.99 | 738 | .3166 |

*Note.* BEAGLE = bound encoding of the aggregate language environment; NPA = noun-phrase attachment; VPA = verb-phrase attachment.

Taraban and McClelland (1988) normed their stimuli by having subjects rate how expected the completion word was from the stem on a 5-point scale. Expectation was subsequently used to predict reading times in the experiment. Table 14 presents Taraban and McClelland's data along with BEAGLE's predictions for the same stimuli.[18] In BEAGLE, the NPA completion words were more similar to the stems than were the VPA completions, $F(1, 17) = 5.29$, $p < .05$.

The key difference between Whittlesea's (2002) stimuli and Taraban and McClelland's (1988) stimuli is whether the stem or completion was varied. Whittlesea used the same completion words in both conditions and varied the stems to manipulate expectation. By contrast, Taraban and McClelland used the same stems in both conditions and varied the completion words to manipulate expectation. In either case, both stems and completions are coded as single vectors in BEAGLE, and both manipulations of expectation are directly predictable from the learned lexical representations.

Experiment 2 of Taraban and McClelland (1988) examined types of noun fillers in their expectation from the stems. For example, given the stem "The janitor cleaned the storage area with the . . . ," fillers may be either (a) fully consistent (*broom*), (b) less expected filler (*solvent*), (c) less expected role (*manager*), or (d) less expected attachment (*odor*). The stimuli were first normed by subjects on a 5-point scale in terms of expectation of completion given the stem and overall plausibility of the sentences; the subjective ratings predicted reading times in a subsequent experiment. Their data are presented in Table 15 along with BEAGLE's cosine predictions for the same stimulus set.

Although the pattern of cosines from BEAGLE is consistent with the data from Taraban and McClelland's (1988) Experiment

[18] Three words in Experiment 1 had no lexical entry in BEAGLE and were substituted for synonyms: *flaunted = exhibited, pepperoni = meat,* and *hideout = shelter*.

Table 15
*Mean Expectation and Plausibility Ratings With Reading Times (ms) to the Four Types of Completion Attachments From Taraban and McClelland (Taraban and McClelland 1988, Experiment 2) and Stem-Completion Cosines From the BEAGLE Representations*

| | Taraban & McClelland (1988) | | | BEAGLE |
|---|---|---|---|---|
| Attachment type | Expectatation | Plausibility | Reading time | Cosine |
| Fully consistent | 4.10 | 4.20 | 355 | .2818 |
| Less expected filler | 2.05 | 2.90 | 365 | .2527 |
| Less expected role | 1.95 | 3.10 | 395 | .2463 |
| Less expected attachment | 1.94 | 3.20 | 400 | .2625 |

*Note.* BEAGLE = bound encoding of the aggregate language environment.

2, the omnibus analysis of variance across the conditions was only marginally significant, $F(3, 69) = 1.97$, $p \approx .10$. Taraban and McClelland found no significant difference in expectation or plausibility ratings between the less expected filler, less expected role, and less expected attachment conditions (see Table 15), but the fully consistent condition was rated significantly higher than the less expected filler, less expected role, and less expected attachment conditions on both measures. In BEAGLE, an orthogonal contrast comparing the fully consistent condition with the mean of the less expected filler, less expected role, and less expected attachment conditions mirrored Taraban and McClelland's findings, $F(1, 23) = 4.12$, $p < .05$.

The representations learned by BEAGLE provide a higher fidelity representation of word meaning that incorporates both context and order information. The resulting representations have been shown to possess the necessary structure to account for human data in a variety of semantic and word prediction tasks.

## General Discussion

The purpose of this article is to demonstrate that a holographic lexicon, representing both word meaning and word order, can be learned by a simple summation–association mechanism applied to the large-scale statistical redundancies present in text. Furthermore, a broad range of psycholinguistic phenomena can be accounted for directly from the structure of lexical representations learned in this way. The model is not intended as a model of language but rather as a memory model that provides an account of the representations to be used by higher order models of language comprehension (e.g., Kintsch, 1988, 1998). The notion is to account for as much behavior as possible from the structure of the knowledge representation before building rules and complexity into a model. As Estes (1994) has noted, "it is clear a priori that once we know what performance occurs in any situation, we can describe it in terms of rules" (p. 245). However, hardwired rules and complex mechanisms should be used as a last resort in modeling, only after simple explanations have been exhausted.

Gibson (1966) warned perceptual theorists not to postulate computational mechanisms for the perceptual system when the necessary information for perception could be obtained from the environment via the sensory system. Similarly, Simon (1996) reminded cognitive theorists not to build complexity into a model's algorithms when the behavior can be understood in terms of a simple organism responding to redundancies in a complex environment. Discussing the path taken by an ant on a beach, Simon noted that the ant's path is "irregular, complex, hard to describe. But the complexity is really a complexity in the surface of the beach, not a complexity in the ant" (Simon, 1996, p. 51).

The work presented here reiterates Simon's (1996) advice in the domain of knowledge representation, reminding theorists not to build unnecessary complexity into either the representation or processing mechanisms when it can be more easily explained from the structure learned by a simple mechanism applied to statistical redundancies in a complex language environment. At some level of comprehension, complex rules may be recruited to account for behavior, and it is possible that these rules cannot be learned from the environment alone. Built-in rules to account for language comprehension should not be the default modeling approach but rather should be examined only for behavior that is too complex to account for from the structure of the simple representation. BEAGLE requires minimum innate control structure in learning and illustrates that a wide range of data may be explained directly from statistical abstraction of the language environment.

BEAGLE assumes that the lexical representation for a word is a pattern of elements across an arbitrary number of dimensions. The true value for each element in a word's representation must be estimated over statistical sampling. There exists a distribution of values for a particular element; with successive observations of the word in language, noise is averaged out, and the true value emerges because of the central limits theorem.

Furthermore, the knowledge representation itself in a model should be as simple as possible and should not contain unnecessary complexity. BEAGLE demonstrates that distributed representations of word meaning and usage can be automatically learned using very simple learning mechanisms rather than by having complexity built artificially into the knowledge representation, as is the case with classic feature list or semantic network representations.

### *Comparing BEAGLE With Other Semantic Space Models*

BEAGLE extends existing semantic space models by introducing a new way to implement the core principle of inducing word meaning from usage in text. BEAGLE improves on criticisms of existing semantic space models (a) by incorporating word order and (b) by exploiting an incremental learning algorithm. Furthermore, because it adapts learning and representation principles from associative memory theory (Murdock, 1982, 1992), the model uses

the same mechanisms to learn word meaning that may be used to form other memory representations and that have been effective in accounting for judgments of frequency and recency (Murdock & Smith, 2001), recognition and serial recall (Mewhort & Popham, 1991; Murdock, 1992), and free recall (Franklin & Mewhort, 2002). These same mechanisms allow the model to retrieve word transition information stored in its lexicon.

The major difference between the representations learned by BEAGLE and those learned by LSA is that BEAGLE includes a consideration of order information in addition to context information. The hyperspace analogue to language (HAL; Burgess & Lund, 2000; Lund & Burgess, 1996) does measure distance (in word steps) in a moving window as it learns but does not explicitly encode order information, nor does it have a mechanism to retrieve sequential dependencies. For the priming data presented in this article, for example, neither LSA nor HAL predicts the pattern of priming found in the human data; whereas HAL underestimates the magnitude of associative priming, LSA overestimates associative priming and underestimates semantic priming. In addition, HAL is unable to predict mediated similarity (Livesay & Burgess, 1998). This incongruence is also found in a variety of other priming data (see Jones et al., 2006). Priming is an example of behavior that depends on both semantic and associative information, and only a representation that considers both types of information can correctly simulate the aggregate human data.

BEAGLE represents both context and order information in a composite representation, giving it a higher fidelity representation of word meaning. As a by-product of using convolution as an order-encoding mechanism, the model can invert the routine and retrieve word transitions from the same representations. Thus, it is a model of both word meaning and sequential dependency information without requiring additional storage or assumptions. Note, however, that convolution–correlation is a primitive model of sequential dependency: Its virtue is that it demonstrates that learned positional information may be retrieved from semantic memory, and for that reason, BEAGLE takes a small step toward an empiricist theory of language. Although BEAGLE cannot currently replace generative models of language, it suggests a reduced role for rule-based processing.

More recent co-occurrence models use Bayesian methods to consider how often words appear together in contexts and apart (e.g., Griffiths & Steyvers, 2002; Griffiths, Steyvers, & Tenenbaum, 2005; Hoffman, 2001; A. E. Smith & Humphreys, 2006; Steyvers & Griffiths, in press). Specifically, topic models (e.g., Griffiths, Steyvers, & Tenenbaum, 2005) use probabilistic methods to infer the generative topics from which documents were created. The idea is that a document is a mixture of generative topics, where a topic is a probability distribution over words.

Griffiths, Steyvers, Blei, and Tenenbaum (2005) have integrated the generative processes from a probabilistic topic model and a hidden Markov model to fuse a model of meaning with one of sequential dependencies. Like BEAGLE, the model considers both syntactic and semantic information as it processes text. However, the model has been tested only on automated lexical class tagging and document classification rather than fitting to human data, and a comparison between the two models at this point would be imprudent without data on common tasks.

Another difference between BEAGLE and other semantic space models concerns the nature of a word's context. In LSA, context is specifically a document (as it is in topic models; e.g., Griffiths, Steyvers, & Tenenbaum, 2005). HAL has no concept of a context but, rather, moves an *n*-word window continuously along a text corpus, bridging sentences. In BEAGLE, however, a word's context is specifically the sentence because order information must be computed within each sentence to correctly consider syntactic information.[19] Nonetheless, when BEAGLE is trained on context information only, the similarity structure of the representations learned is very similar to those learned by LSA when both are trained on the same text corpus. Changing BEAGLE to compute context information across the paragraph or document, rather than the sentence, produces representations with similarity only subtly more like LSA. Thus, it does not appear that the size of the context is particularly important to computation of context information in BEAGLE.

It is possible that BEAGLE is particularly sensitive to its initial conditions. With models such as HAL, LSA, or topics, the initial data are constant across several runs (e.g., a Term × Document matrix of the textbase in the case of LSA).[20] Hence, these models are likely to produce the same representation on multiple training runs of the same input corpus. By contrast, BEAGLE begins with random noise and gradually accumulates structure across statistical sampling. The environmental representations for a word are created at random and are different on any two training runs of the same corpus (unless the seed is stored). It follows that the lexical representations for a word on any two runs are only randomly similar as well. The semantic similarity is in the structural similarities between words within the lexicon; even though lexicons from any two runs are different, the pattern of interword cosines is remarkably similar. Thus, the representation for *dog* is different on two training runs, but the similarity of *dog* with *cat* is highly congruent on both runs. We have compared the similarity structure on multiple lexicons trained on the same corpus, and the interword similarity structures between lexicons are quite consistent from different random initial conditions. Provided that the initial conditions in BEAGLE are indeed close to random, the exact initial conditions seem to be unimportant.

A criticism that is common to both semantic space models and SRNs has to do with the role of supervision in learning word meanings; this criticism is also valid against BEAGLE. Although children obviously benefit from feedback and supervision when learning word meanings, McClelland has noted that most unsupervised models are analogous to "learning language by listening to the radio" (as cited in Elman, 1990, p. 201). Although semantic space models can learn word meanings without the need for feedback, supervision is certainly an important factor in human word learning, and semantic models need to account for it. Elman (2004) has suggested that feedback can be derived from the stimulus environment: An SRN predicts the next word in a sequence

[19] Because sentences within paragraphs are more related than sentences across paragraphs, Tom Landauer (personal communication, 2005) has suggested that an additional random vector representing the paragraph context could be added to calculation of the sentence context for each word as it is encountered in a new paragraph. This would produce greater similarity for words that have appeared together in paragraphs. For similar usage of random vectors to represent context, see Dennis and Humphreys (2001).

[20] The topics model is more stochastic because of Gibbs sampling.

and then looks ahead to see what the correct word actually is, using the feedback to adjust its connection weights. Plate (2003) has shown, however, that an SRN can learn as well without feedback for error correction; thus, Elman's notion of supervision in an SRN may be unnecessary.

### *Modularity and Composite Systems*

Traditionally, there has been a widespread assumption that knowledge of a word's meaning, knowledge of its lexical class, and knowledge of its grammatical usage are separate types of information involving different forms of representation stored separately. Information about a word's meaning is stored in the lexicon as a dictionary-type definition, and knowledge of its grammatical usage is represented in the form of rules or production systems. Although BEAGLE is by no means a model of syntax but, rather, a model of memory, it can produce limited syntactic behavior by simply inverting its learning routines from the structure of the memory representation. It is not directly obvious at what point the structure of memory can no longer simulate transition behavior and rules must be relied upon (e.g., novel strings or long-range dependencies).

In addition to representing both types of information as vector patterns, BEAGLE represents both with the same vector pattern. The composite representation containing a superposition of context and order information functions as well at the tasks reported here as a concatenated system, with context and order information represented by distinct vectors. In keeping with Occam's razor, thus, there does not appear to be a need for separate representations when a single representation accounts for the data equally well. Meaning and order information may be represented in the same form and within the same representation, as Murdock (1982) has demonstrated with memory for items and associations.

Because both context information and order information are represented in the same vector, multiple meanings for lexically ambiguous words can be stored without the need for multiple representations. Multiple meanings of an ambiguous word are simultaneously activated when the word is processed (Foss, 1970; Onifer & Swinney, 1981; Tanenhaus, Leiman, & Seidenberg, 1979) even when one meaning is clearly dominant (Burgess, Seidenberg, & Tanenhaus, 1988). These results are often taken as evidence that the multiple meanings of a polysemous word have their own distinct lexical representations.

In BEAGLE, a polysemous word, such as *bank,* has only one lexical representation. The pattern of elements representing *bank*'s meaning is dominated by the more frequent financial institution sense; however, information about the river shore sense is also stored in the same pattern of elements and may be disambiguated when the word is used in context. A sentence context that is sufficiently biased toward one meaning of an ambiguous word can enhance activation of the congruent meaning (Tabossi, 1988). Even when this effect is not observed in response latency, it can be by using neurocognitive measures (Van Petten & Kutas, 1987). Both the frequency of each meaning and the prior context affect the activation of different word meanings (Sereno, O'Donnell, & Rayner, 2006).

When used in different contexts, such as "I robbed the *bank* at gunpoint" and "I was fishing from the river *bank,*" different meanings of *bank* emerge from BEAGLE's lexicon; however, the same lexical representation of $\boldsymbol{m}_{bank}$ responds in both contexts. Multiple meanings for polysemous words can be stored together within the same holographic representation—different meanings emerge from a single representation, depending on context and order, when it is presented in a sentence.

### *Conclusion*

Language and comprehension are extremely complicated behaviors. We do not claim to have a full theory of these behaviors. Rather, we claim that knowledge underpinning the behaviors is learned and that it can be acquired using simple mechanisms operating on a large scale. Whereas LSA has shown that a word's contextual history can be learned, we have shown that its order history can also be learned and, furthermore, that the two types of information can be stored together in a composite holographic representation. In short, we have pushed LSA's view of language to a new level by bringing syntactic issues within the range of the empiricist program. It would be imprudent to claim that BEAGLE captures the complexity of syntax in natural language, but it is fair to claim that BEAGLE pushes the empiricist program closer to that goal. Finally, models of higher order comprehension can be simplified if they adopt BEAGLE's use of order information. Order information may allow such models to limit the application of rule-based mechanisms.

Learning and representation in the model are not limited to text but may be applied to statistical redundancies in many environments. In this article, we have trained BEAGLE only on text. However, text is an impoverished version of the full input available to the human system. Modern models should also consider the grounding of meaning in perception and action (see Glenberg, de Vega, & Graesser, in press, for recent progress). Text is a convenient set of data on which to train BEAGLE. However, the model could learn regularities from visual or auditory input equally well (given appropriate front-end representation). More work on integrating multisensory data in statistical models is needed, as it is in the role of supervision and environmental feedback in statistical learning.

## References

Anderson, J. R., & Bower, G. H. (1973). *Human associative memory.* Washington, DC: Winston.

Balota, D. A., & Lorch, R. F. (1986). Depth of automatic spreading activation: Mediated priming effects in pronunciation but not in lexical decision. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 12,* 336–345.

Battig, W. R., & Montague, W. E. (1969). Category norms for verbal items in 56 categories: A replication and extension of the Connecticut category norms. *Journal of Experimental Psychology Monographs, 80*(3, Pt. 2), 1–46.

Booth, T. L. (1969). Probabilistic representation of formal languages. In *10th Annual IEEE Symposium on Switching and Automata Theory* (pp. 74–81). New York: Institute of Electrical and Electronics Engineers.

Brooks, L. R. (1978). Nonanalytic concept formation and memory for instances. In E. Rosch & B. B. Lloyd (Eds.), *Cognition and categorization* (pp. 169–211). Hillsdale, NJ: Erlbaum.

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

Burgess, C., Seidenberg, M. S., & Tanenhaus, M. K. (1989). Context and lexical access: Implications of nonword interference for lexical ambigu-

ity resolution. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15,* 620–632.

Chiarello, C., Burgess, C., Richards, L., & Pollok, A. (1990). Semantic and associative priming in the cerebral hemispheres: Some words do, some words don't . . . sometimes, some places. *Brain & Language, 38,* 75–104.

Chomsky, N. (1965). *Aspects of a theory of syntax.* Cambridge, MA: MIT Press.

Chomsky, N. (1980). *Rules and representations.* New York: Columbia University Press.

Christiansen, M. H., & Chater, N. (1999). Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science, 23,* 157–205.

Chwilla, D. J., & Kolk, H. J. (2000). Mediated priming in the lexical decision task: Evidence from event-related potentials and reaction time. *Journal of Memory and Language, 42,* 314–341.

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

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

Collins, A. M., & Quillian, M. R. (1972). How to make a language user. In E. Tulving & W. Donaldson (Eds.), *Organization of memory* (pp. 309–351). New York, NY: Academic Press.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society For Information Science, 41,* 391–407.

Dennis, S., & Humphreys, M. S. (2001). A context noise model of episodic word recognition. *Psychological Review, 108,* 452–478.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science, 14,* 179–211.

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning, 7,* 195–225.

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition, 48,* 71–99.

Elman, J. L. (1995). Language as a dynamical system. In R. F. Port & T. van Gelder (Eds.), *Mind as motion: Explorations in the dynamics of cognition* (pp. 195–223). Cambridge, MA: MIT Press.

Elman, J. L. (2004). An alternate view of the mental lexicon. *Trends in Cognitive Science, 8,* 301–306.

Estes, W. K. (1994). *Classification and cognition.* Oxford, England: Oxford University Press.

Fellbaum, C. (1998). *WordNet: An electronic lexical database.* Cambridge, MA: MIT Press.

Firth, J. R. (1957). A synopsis of linguistic theory, 1930–1955. In J. R. Firth (Ed.), *Studies in linguistic analysis* (pp. 1–32). Oxford, England: Blackwell.

Fischler, I. S., & Bloom, P. A. (1979). Automatic and attentional processes in the effects of sentence contexts on word recognition. *Journal of Verbal Learning and Verbal Behavior, 18,* 1–20.

Fischler, I. S., & Bloom, P. A. (1985). Effects of constraint and validity of sentence contexts on lexical decisions. *Memory & Cognition, 13,* 128–139.

Fleischman, M., & Roy, D. (2005). Why are verbs harder to learn than nouns: Initial insights from a computational model of intention recognition in situated word meaning. In B. G. Bara, L. Barsalou, & M. Bucciarelli (Eds.), *Proceedings of the 27th Annual Conference of the Cognitive Science Society* (pp. 684–689). Mahwah, NJ: Erlbaum.

Foss, D. J. (1970). Some effects of ambiguity upon sentence comprehension. *Journal of Verbal Learning and Verbal Behavior, 9,* 699–706.

Franklin, D. R. J., & Mewhort, D. J. K. (2002). An analysis of immediate memory: The free-recall task. In N. J. Dimpoloulos & K. F. Li (Eds.), *High performance computing systems and applications* (pp. 465–480). New York: Kluwer Academic.

Frazier, L., & Rayner, K. (1982). Making and correcting errors during sentence comprehension: Eye movements in the analysis of structurally ambiguous sentences. *Cognitive Psychology, 14,* 178–210.

Gabel, R. A., & Roberts, R. A. (1973). *Signals and linear systems.* New York: Wiley.

Gentner, D. (1982). Why are nouns learned before verbs: Linguistic relativity versus natural partitioning. In S. Kuzaj (Ed.), *Language development: Vol. 2. Language, cognition, and culture* (pp. 301–334). Hillsdale, NJ: Erlbaum.

Gibson, J. J. (1966). *The senses considered as perceptual systems.* Boston: Houghton Mifflin.

Gillette, J., Gleitman, H., Gleitman, L., & Lederer, A. (1999). Human simulation of vocabulary learning. *Cognition, 73,* 135–176.

Glenberg, A. M., de Vega, M., & Graesser, A. G. (Eds.). (in press). *Symbols, embodiment, and meaning: A workshop and debate.*

Griffiths, T. L., & Steyvers, M. (2002). A probabilistic approach to semantic representation. In W. D. Gray & C. D. Schunn (Eds.), *Proceedings of the Twenty-Fourth Annual Conference of the Cognitive Science Society* (pp. 381–386). Mahwah, NJ: Erlbaum.

Griffiths, T. L., Steyvers, M., Blei, D. M., & Tenenbaum, J. B. (2005). Integrating topics and syntax. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems* (Vol. 17, pp. 537–544). San Mateo, CA: Morgan Kaufmann Publishers.

Griffiths, T. L., Steyvers, M., & Tenenbaum, J. B. (2005). *Topics in semantic representation.* Manuscript submitted for publication.

Heiser, W. J. (1988). Selecting a stimulus set with prescribed structure from empirical confusion frequencies. *British Journal of Mathematical and Statistical Psychology, 41,* 37–51.

Hintzman, D. L. (1984). Minerva 2: A simulation model of human memory. *Behavior Research Methods, Instruments, & Computers, 16,* 96–101.

Hoffman, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning Journal, 42,* 177–196.

Howard, M. W., & Kahana, M. J. (2002). A distributed representation of temporal context. *Journal of Mathematical Psychology, 46,* 269–299.

Huchinson, K. A. (2003). Is semantic priming due to association strength or feature overlap? A microanalytic review. *Psychonomic Bulletin & Review, 10,* 785–813.

Jelinek, F., & Lafferty, J. D. (1991). Computation of the probability of initial substring generation by stochastic context-free grammars. *Computational Linguistics, 17,* 315–323.

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.

Kintsch, W. (1988). The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review, 95,* 163–182.

Kintsch, W. (1998). *Comprehension: A paradigm for cognition.* New York: Cambridge University Press.

Kintsch, W. (2001). Predication. *Cognitive Science, 25,* 173–202.

Kounios, J., & Holcomb, P. J. (1992). Structure and process in semantic memory: Evidence from event-related potentials and reaction time. *Journal of Experimental Psychology: General, 121,* 459–479.

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

Laham, R. D. (2000). *Automated content assessment of text using latent semantic analysis to simulate human cognition.* Unpublished doctoral dissertation, University of Colorado at Boulder.

Landauer, T. K., & Dumais, S. T. (1994). Latent semantic analysis and the measurement of knowledge. In R. M. Kaplan & J. C. Burstein (Eds.), *Educational Testing Service Conference on Natural Language Processing Techniques and Technology in Assessment and Education*. Princeton, NJ: Educational Testing Service.

Landauer, T. K., & Dumais, S. T. (1996). How come you know so much? From practical problem to theory. In D. Hermann, C. Hertzog, C. McEvoy, P. Hertel, & M. Johnson (Eds.), *Basic and applied memory research: Vol. 1. Theory in context* (pp. 105–126). Mahwah, NJ: Erlbaum.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem:

The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review, 104,* 211–240.

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

Landauer, T. K., Laham, D., Rehder, B., & Schreiner, M. E. (1997). How well can passage meaning be derived without using word order? A comparison of latent semantic analysis and humans. In M. G. Shafto & P. Langley (Eds.), *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society* (pp. 412–417). Mahwah, NJ: Erlbaum.

Livesay, K., & Burgess, C. (1998). Mediated priming in high-dimensional semantic space: No effect of direct semantic relationships or co-occurrence. *Brain & Cognition, 28,* 203–208.

Lucas, M. (2000). Semantic priming without association: A meta-analytic review. *Psychonomic Bulletin & Review, 7,* 618–630.

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

Maki, W. S., McKinley, L. N., & Thompson, A. G. (2004). Semantic distance norms computed from an electronic dictionary (WordNet). *Behavior Research Methods, Instruments, & Computers, 36,* 421–431.

Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics, 19,* 313–303.

Masson, M. E. J. (1995). A distributed memory model of semantic priming. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 21,* 3–23.

McFalls, E. L., Schwanenflugel, P. J., & Stahl, S. A. (1996). Influence of word meaning on the acquisition of a reading vocabulary in second-grade children. *Reading & Writing, 8,* 235–250.

McKoon, G., & Ratcliff, R. (1992). Spreading activation versus compound cue accounts of priming: Mediated priming revisited. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18,* 1155–1172.

McNamara, T. P. (2005). *Semantic priming: Perspectives from memory and word recognition.* New York: Psychology Press.

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

Metcalfe-Eich, J. (1982). A composite holographic associative recall model. *Psychological Review, 89,* 627–661.

Mewhort, D. J. K., & Johns, E. E. (2005). Sharpening the echo: An iterative-resonance model for short-term recognition memory. *Memory, 13,* 300–307.

Mewhort, D. J. K., & Popham, D. (1991). Serial recall of tachistoscopic letter strings. In W. Hockley & S. Lewandowsky (Eds.), *Relating theory and data: Essays in honor of Bennet B. Murdock* (pp. 425–443). Hillsdale, NJ: Erlbaum.

Meyer, D. E., & Schvaneveldt, R. W. (1971). Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. *Journal of Experimental Psychology, 90,* 227–234.

Miller, G. A. (1951). *Language and communication.* New York: McGraw-Hill.

Miller, G. A. (Ed.). (1990). WordNet: An on-line lexical database [Special issue]. *International Journal of Lexicography, 3*(4).

Miller, G. A. (1999). Nouns in WordNet. In C. Fellbaum (Ed.), *WordNet: An electronic lexical database* (pp. 23–46). Cambridge, MA: MIT Press.

Miller, G. A., & Selfridge, J. A. (1950). Verbal context and the recall of meaningful material. *American Journal of Psychology, 63,* 176–185.

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

Murdock, B. B. (1992). Item and associative information in a distributed memory model. *Journal of Mathematical Psychology, 36,* 68–98.

Murdock, B. B. (1993). Derivations for the chunking model. *Journal of Mathematical Psychology, 37,* 421–445.

Murdock, B. B., & Smith, D. G. (2001). Judgments of frequency and recency in a distributed memory model. *Journal of Mathematical Psychology, 45,* 564–602.

Neath, I. (1997). Modality, concreteness and set-size effects in a free reconstruction of order task. *Memory & Cognition, 25,* 256–263.

Neath, I. (1998). *Human memory: An introduction to research, data, and theory.* Pacific Grove, CA: Brooks/Cole.

Neely, J. H. (1991). Semantic priming effects in visual word recognition: A selective review of current findings and theories. In D. Besner & G. W. Humphreys (Eds.), *Basic processes in reading: Visual word recognition* (pp. 264–336). Hillsdale, NJ: Erlbaum.

Nelson, D. L., McEvoy, C. L., & Schreiber, T. A. (1998). *The University of South Florida word association, rhyme, and word fragment norms.* Retrieved November 2, 2006, from http://w3.usf.edu/FreeAssociation

Nelson, K. (1974). Concept, word, and sentence: Interrelations in acquisition and development. *Psychological Review, 81,* 267–285.

Nelson, K. (1981). Individual differences in language development: Implications for development and language. *Developmental Psychology, 17,* 170–187.

Onifer, W., & Swinney, D. A. (1981). Accessing lexical ambiguities during sentence comprehension: Effects of frequency of meaning and contextual bias. *Memory & Cognition, 15,* 225–236.

Osgood, C. E. (1941). Ease of individual judgment processes in relation to polarization of attitudes in the culture. *Journal of Social Psychology, 14,* 403–418.

Osgood, C. E. (1952). The nature and measurement of meaning. *Psychological Bulletin, 49,* 197–237.

Osgood, C. E. (1971). Exploration in semantic space: A personal diary. *Journal of Social Issues, 27,* 5–64.

Paivio, A. (1971). *Imagery and mental processes.* New York: Holt, Rinehart & Winston.

Paivio, A. (1986). *Mental representations: A dual-coding approach.* Oxford, England: Oxford University Press.

Paivio, A., Yuille, J. C., & Madigan, S. A. (1968). Concreteness, imagery, and meaningfulness values for 925 nouns. *Journal of Experimental Psychology, 76*(1, Pt. 2), 1–25.

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics, 31,* 71–106.

Perfetti, C. A. (1998). The limits of co-occurrence: Tools and theories in language research. *Discourse Processes, 25,* 363–377.

Pexman, P. M., Holyk, G. C., & Monfils, M. (2003). Number-of-features effects and semantic processing. *Memory & Cognition, 31,* 842–855.

Pexman, P. M., Lupker, S. J., & Hino, Y. (2002). The impact of feedback semantics in visual word recognition: Number-of-features effects in lexical decision and naming tasks. *Psychonomic Bulletin & Review, 9,* 542–549.

Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks, 6,* 623–641.

Plate, T. A. (2003). *Holographic reduced representations* (CSLI Lecture Notes No. 150). Stanford, CA: CSLI Publications.

Plaut, D. C. (1995). Semantic and associative priming in a distributed attractor network. In Cognitive Science Society (Ed.), *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society* (pp. 37–42). Hillsdale, NJ: Erlbaum.

Pustejovsky, J. (1996). *The generative lexicon.* Cambridge, MA: MIT Press.

Pylyshyn, Z. W. (1973). What the mind's eye tells the mind's brain: A critique of mental imagery. *Psychological Bulletin, 80,* 1–24.

Pylyshyn, Z. W. (1981). The mental imagery debate: Analog media versus tacit knowledge. *Psychological Review, 88,* 16–45.

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review, 85,* 59–108.

Rips, L. J., Shoben, E. J., & Smith E. E. (1973). Semantic distance and the verification of semantic relations. *Journal of Verbal Learning and Verbal Behavior, 12,* 1–20.

Rosch, E. H. (1973). On the internal structure of perceptual and semantic

categories. In T. E. Moore (Ed.), *Cognitive development and the acquisition of language* (pp. 111–144). New York: Academic Press.

Rosch, E. H. (1975). Cognitive representations of semantic categories. *Journal of Experimental Psychology: General, 104,* 192–233.

Salton, G. (1973). Recent studies in automatic text analysis and document retrieval. *Journal of the Association of Computing Machinery, 20,* 258–278.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the Association of Computing Machinery, 18,* 613–620.

Schuberth, R. E., & Eimas, P. D. (1977). Effects of context on the classification of words and nonwords. *Journal of Experimental Psychology: Human Perception and Performance, 3,* 27–36.

Semon, R. (1923). *Mnemic psychology* (B. Duffy, Trans). London: Allen & Unwin. (Original work published in 1909)

Serafin, R., & Di Eugenio, B. (2004). FLSA: Extending latent semantic analysis with features for dialogue act classification. In Association for Computational Linguistics (Ed.), *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics* (pp. 692–699). East Stroudsburg, PA: Association for Computational Linguistics.

Sereno, S. C., O'Donnell, P. J., & Rayner, K. (2006). Eye movements and lexical ambiguity resolution: Investigating the subordinate-bias effect. *Journal of Experimental Psychology: Human Perception and Performance, 32,* 335–350.

Servan-Schreiber, D. A., Cleeremans, A., & McClelland, J. L. (1991). Graded state machines: The representation of temporal contingencies in simple recurrent networks. *Machine Learning, 7,* 161–194.

Shelton, J. R., & Martin, R. C. (1992). How semantic is automatic semantic priming? *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18,* 1191–1210.

Simon, H. A. (1996). *The sciences of the artificial* (3rd ed.). Cambridge, MA: MIT Press.

Smith, A. E., & Humphreys, M. S. (2006). Evaluation of unsupervised semantic mapping of natural language with Leximancer concept mapping. *Behavior Research Methods, 38,* 262–279.

Smith, E. E., Shoben, E. J., & Rips, L. J. (1974). Structure and process in semantic memory: A featural model for semantic decisions. *Psychological Review, 81,* 214–241.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence, 46,* 159–216.

Snedeker, J., & Gleitman, L. (2004). Why it is hard to label our concepts. In D. G. Hall & S. R. Waxman (Eds.), *Weaving a lexicon* (pp. 257–294). Cambridge, MA: MIT Press.

Solan, Z., Horn, D., Ruppin, E., & Edelman, S. (2005). Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences, USA, 102,* 11629–11634.

Steiger, J. H. (1980). Tests for comparing elements of a correlation matrix. *Psychological Bulletin, 87,* 245–251.

Steyvers, M. (2000). *Modeling semantic and orthographic similarity effects on memory for individual words.* Unpublished doctoral dissertation, Indiana University Bloomington.

Steyvers, M., & Griffiths, T. (in press). Probabilistic topic models. In T. Landauer, D. McNamara, S. Dennis, & W. Kintsch (Eds.), *Latent semantic analysis: A road to meaning.* Mahwah, NJ: Erlbaum.

Steyvers, M., & Tenenbaum, J. B. (2005). The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. *Cognitive Science, 29,* 41–78.

Tabossi, P. (1988). Accessing lexical ambiguity in different types of sentential contexts. *Journal of Memory and Language, 27,* 324–340.

Tanenhaus, M. K., Leiman, J. M., & Seidenberg, M. S. (1979). Combinatory lexical information and language comprehension. In G. Altmann (Ed.), *Cognitive models of speech processing* (pp. 383–408). Cambridge, MA: MIT Press.

Taraban, R., & McClelland, J. L. (1988). Constituent attachment and thematic role assignment in sentence processing: Influences of content-based expectations. *Journal of Memory and Language, 27,* 597–632.

Turney, P. (2001). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In L. De Raedt & P. Flach (Eds.), *Proceedings of the 12th European Conference on Machine Learning* (pp. 491–502). Freiburg, Germany: European Conference on Machine Learning.

Van der Heijden, A. H., Malhas, M. S., & Van den Roovaart, B. P. (1984). An empirical interletter confusion matrix for continuous-line capitals. *Perception & Psychophysics, 35,* 85–88.

Van Petten, C., & Kutas, M. (1987). Ambiguous words in context: An event-related potential analysis of the time course of meaning activation. *Journal of Memory and Language, 26,* 188–208.

Whittlesea, B. W. A. (1993). Illusions of familiarity. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 19,* 1235–1253.

Whittlesea, B. W. A. (2002). Two routes to remembering (and another to remembering not). *Journal of Experimental Psychology: General, 131,* 325–348.

Wiemer-Hastings, P. (2000). Adding syntactic information to LSA. In L. R. Gleitman & A. K. Joshi (Eds.), *Proceedings of the Twenty-Second Annual Conference of the Cognitive Science Society* (pp. 989–1015). Mahwah, NJ: Erlbaum.

Wiemer-Hastings, P. (2001). Rules for syntax, vectors for semantics. In J. D. Moore & K. Stenning (Eds.), *Proceedings of the Twenty-Third Annual Conference of the Cognitive Science Society* (pp. 1140–1145). Mahwah, NJ: Erlbaum.

Williams, E. J. (1959). The comparison of regression variables. *Journal of the Royal Statistical Society, Series B, 21,* 396–399.

Williams, J. N. (1996). Is automatic priming semantic? *European Journal of Cognitive Psychology, 8,* 113–161.

## Appendix A

## Learning and Retrieving Order Information From the Lexicon

The learning of context information in the bound encoding of the aggregate language environment (BEAGLE) model is straightforward and is described in detail in the model description section of the main text of this article: A word's context history is the sum of the environmental vectors for all word tokens it has occurred in sentences with. The learning and retrieval of order information, however, have necessarily been simplified in the model description section, and the algorithms for encoding and decoding of order information are described here in greater detail.

### Encoding Order Information

When learning order information, BEAGLE produces directional associations by permuting the argument vectors in Positions 1 and 2 differently prior to each pairwise convolution (adapted from Plate, 1995). When the program initializes, encoding remapping functions are created for Positions 1 and 2, $E_1$ and $E_2$, respectively. Each function creates a scrambling of the elements in a vector passed to it; the scrambling order is determined randomly at initialization and is then constant throughout learning. A vector, $\mathbf{v}$, has the order of its elements scrambled differently by $E_1$ or $E_2$. Thus, $E_1(\mathbf{v})$ contains the same elements as $E_2(\mathbf{v})$ but in a different order, and hence, the expected cosine of $E_1(\mathbf{v})$ and $E_2(\mathbf{v})$ is zero.

During encoding, *dog bite* and *bite dog* produce unique associations because the order of the arguments differ, even though convolution is itself commutative: $E_1(\boldsymbol{e}_{dog}) \circledast E_2(\boldsymbol{e}_{bite}) \neq E_1(\boldsymbol{e}_{bite}) \circledast E_2(\boldsymbol{e}_{dog})$. For example, when encoding the position of $B$ in the sequence *ABCD,* the following operations would be computed to yield its order vector in this "sentence."

| | To code: |
|---|---|
| $Bind_{B,1} = E_1(A) \circledast E_2(\boldsymbol{\Phi})$ | $A\_$ |
| $Bind_{B,2} = E_1(\boldsymbol{\Phi}) \circledast E_2(C)$ | $\_C$ |
| $Bind_{B,3} = E_1[E_1(A) \circledast E_2(\boldsymbol{\Phi})] \circledast E_2(C)$ | $A\_C$ |
| $Bind_{B,4} = E_1[E_1(\boldsymbol{\Phi}) \circledast E_2(C)] \circledast E_2(D)$ | $\_CD$ |
| $Bind_{B,5} = E_1(E_1[E_1(A) \circledast E_2(\boldsymbol{\Phi})] \circledast E_2(C)) \circledast E_2(D)$ | $A\_CD$ |

$$\mathbf{o}_B = \sum_{j=1}^{5} Bind_{B,j}$$

### Decoding With Circular Correlation

Learned items may be decoded from the order information in a holographic vector using inverse circular correlation, $y \approx x \circledcirc z$. To decode environmental vectors from memory properly, the encoding mapping functions must be undone in the appropriate order. Thus, let $D_1$ and $D_2$ (decoding mappings) be the inverse of $E_1$ and $E_2$, respectively. If $\boldsymbol{m}_{dog}$ is encoded as $\boldsymbol{m}_{dog} = [E_1(\boldsymbol{\Phi}) \circledast E_2(\boldsymbol{e}_{bite})] + [E_1(\boldsymbol{e}_{feed}) \circledast E_2(\boldsymbol{\Phi})]$ (i.e., both *dog bite* and *feed dog* are summed into the representation), $\boldsymbol{e}_{bite}$ may be decoded as approximately equal to $D_2[E_1(\boldsymbol{\Phi}) \circledcirc \boldsymbol{m}_{dog}]$. From the same lexical vector, the function $D_1[E_2(\boldsymbol{\Phi}) \circledcirc \boldsymbol{m}_{dog}]$ decodes a facsimile of $\boldsymbol{e}_{feed}$. The example of decoding to the right or left of the word *luther,* demonstrated in the main text of the article, was specifically

$$D_1[E_2(\boldsymbol{\Phi}) \circledcirc \boldsymbol{m}_{luther}] \approx \boldsymbol{e}_{martin},$$

and

$$D_2[E_1(\boldsymbol{\Phi}) \circledcirc \boldsymbol{m}_{luther}] \approx \boldsymbol{e}_{king}.$$

As more $n$-grams are included in the probe (up to the value of $\lambda$), the decoded response will be more constrained. For $n$-grams larger than a bigram, decoding is computed in a slightly different way to the left and right of a blank position.

For example, assume that a lexical vector, $\boldsymbol{m}_B$, has the chunk $ABC$ coded into it:

$$\begin{aligned} \boldsymbol{m}_B = {} & [E_1(A) \circledast E_2(\boldsymbol{\Phi})] + && \rightarrow A\_\_ \\ & [E_1(\boldsymbol{\Phi}) \circledast E_2(C)] + && \rightarrow \_\_C \\ & (E_1[E_1(A) \circledast E_2(\boldsymbol{\Phi})] \circledast E_2(C)). && \rightarrow A\_C \end{aligned}$$

To use information from $\boldsymbol{m}_B$ to determine which vector fits in $AB\_$, information from the last two traces coded is needed. $\_C$ contains information about what vector succeeds $B$ and may be decoded as $D_2[E_1(\boldsymbol{\Phi}) \circledcirc \boldsymbol{m}_B] \approx C$. However, the trace $A\_C$ also contains information about what vector succeeds $B$ conditional on $A$ preceding $B$. To decode this, $A\_$ may be built and correlated with $\boldsymbol{m}_B$; specifically,

$$D_2[E_1(E_1(A) \circledast E_2(\boldsymbol{\Phi})) \circledcirc \boldsymbol{m}_B] \approx C.$$

If $C$ succeeds $B$ only as a bigram but the sequence $AB$ always precedes another symbol, $D$, then the vector retrieved from $\boldsymbol{m}_B$ when probed with $B\_$ will be similar to $C$, and the vector retrieved when probed with $AB\_$ (i.e., conditional on $A$ preceding $B$) will be more similar to $D$.

Because coding was done left to right with convolution, decoding backwards a number of steps is slightly more complicated. Again assume that

$$\boldsymbol{m}_B = [E_1(A) \circledast E_2(\boldsymbol{\Phi})] + [E_1(\boldsymbol{\Phi}) \circledast E_2(C)] + (E_1[E_1(A) \circledast E_2(\boldsymbol{\Phi})] \circledast E_2(C)).$$

To use information from $\boldsymbol{m}_B$ to determine which vector fits in $\_BC$, $\boldsymbol{m}_B$ must be iteratively unpacked. $A\_$ contains information about what vector preceded $B$ and may be decoded as $D_1[E_2(\boldsymbol{\Phi}) \circledcirc \boldsymbol{m}_B] \approx A$. However, $A\_C$ also contains information about what vector preceded $B$ conditional on $C$ succeeding $B$. The trace $A\_C$ was packed two vectors at a time, left to right, specifically, $E_1[E_1(A) \circledast E_2(\boldsymbol{\Phi})] \circledast E_2(C)$, and thus, it must be iteratively unpacked from right to left to predict the first position. $D_1[E_2(C) \circledcirc \boldsymbol{m}_B]$ unpacks approximately as $E_1(A) \circledast E_2(\boldsymbol{\Phi})$, and $D_1(E_2(\boldsymbol{\Phi}) \circledcirc [E_1(A) \circledast E_2(\boldsymbol{\Phi})])$ unpacks a facsimile of $A$. The entire operation to decode the symbol that precedes $B$ from the memory, $\boldsymbol{m}_B$, may be written as

$$D_1[E_2(\boldsymbol{\Phi}) \circledcirc \boldsymbol{m}_B] + D_1[E_2(\boldsymbol{\Phi}) \circledcirc D_1(E_2(C) \circledcirc \boldsymbol{m}_B)] \approx A.$$

Furthermore, $\boldsymbol{m}_C$ also contains a dependency that $A$ appeared in Position 1 if $B$ is seen in Position 2 ($AB\_$), and this information can be

(*Appendixes continue*)

used to further disambiguate the blank position. All chunks up to λ can be used to retrieve order information, but decoding becomes more complicated to demonstrate.

As a concrete example, consider decoding the blank position in the triplet *martin luther* _____.

$\boldsymbol{e}_{king} \approx D_2[E_1(\boldsymbol{\Phi}) \circledast \boldsymbol{m}_{luther}] +$ *luther* __ from $\boldsymbol{m}_{luther}$

$D_2[E_1(E_1(\boldsymbol{e}_{martin}) \circledast E_2(\boldsymbol{\Phi})) \circledast \boldsymbol{m}_{luther}]$ *martin luther* __ from $\boldsymbol{m}_{luther}$

+

$D_2[E_1(E_1(\boldsymbol{\Phi}) \circledast E_2(\boldsymbol{e}_{luther}) \circledast \boldsymbol{m}_{martin}]$. *martin luther* __ from $\boldsymbol{m}_{martin}$

Adding together these three decoded vectors produces a vector with a similar pattern of elements to $\boldsymbol{e}_{king}$. Decoding the first position in ____ *luther king,* however, would be

$\boldsymbol{e}_{martin} \approx D_1[E_2(\boldsymbol{\Phi}) \circledast \boldsymbol{m}_{luther}] +$ __ *luther* from $\boldsymbol{m}_{luther}$

$D_1[E_2(\boldsymbol{\Phi}) \circledast D_1(E_2(\boldsymbol{e}_{king}) \circledast \boldsymbol{m}_{luther}] +$ __ *luther king* from $\boldsymbol{m}_{luther}$

$D_1[E_2(\boldsymbol{e}_{luther}) \circledast D_1(E_2(\boldsymbol{\Phi}) \circledast \boldsymbol{m}_{king}]$. __ *luther king* from $\boldsymbol{m}_{king}$

Decoding the middle position of *martin* _____ *king* would come from four sources:

$\boldsymbol{e}_{luther} \approx D_2[E_1(\boldsymbol{\Phi}) \circledast \boldsymbol{m}_{martin}] +$ *martin* __ from $\boldsymbol{m}_{martin}$

$D_1[E_2(\boldsymbol{\Phi}) \circledast \boldsymbol{m}_{king}] +$ __ *king* from $\boldsymbol{m}_{king}$

$D_2[E_1(\boldsymbol{\Phi}) \circledast D_1(E_2(\boldsymbol{e}_{king}) \circledast \boldsymbol{m}_{martin}] +$ *martin* __ *king* from $\boldsymbol{m}_{martin}$

$D_2[E_1(\boldsymbol{e}_{martin}) \circledast D_1(E_2(\boldsymbol{\Phi}) \circledast \boldsymbol{m}_{king})]$. *martin* __ *king* from $\boldsymbol{m}_{king}$

Sentences and collocation phrases may be progressively unpacked from the holographic vectors. To use the same chunk again as an example, when the lexical vector for *martin* is decoded to the right, $\boldsymbol{e}_{luther}$ is retrieved. If *martin* were decoded to the left, however, no word could be reliably retrieved.[A1] Given *martin,* if *luther* is further decoded to the right, *king* is retrieved, and with this sequence of three words, if *king* is decoded to the right, *jr* is retrieved, producing the learned collocate *martin luther king jr.* If *jr* is further decoded right, no consistent word can be retrieved.

Descriptive source code, test data, and demonstrations for BEAGLE may be found online at http://www.indiana.edu/~clcl/BEAGLE

[A1] The phrase *doctor martin luther king jr* is not found in TASA.

## Appendix B

### List of Exemplars by Category

| COLORS | SPORTS | FRUIT | COUNTRIES |
|---|---|---|---|
| red | football | apple | Canada |
| blue | baseball | orange | Mexico |
| green | basketball | banana | Australia |
| yellow | hockey | grape | France |
| purple | rugby | strawberry | Germany |
| white | volleyball | plum | Japan |
| black | gymnastics | pear | China |
| gray | soccer | lemon | Brazil |
| orange | triathlon | lime | Spain |
| cyan | wrestling | cherry | India |
| magenta | boxing | grapefruit | Netherlands |
| beige | golf | apricot | Cuba |
| pink | lacrosse | pineapple | Poland |
| brown | motocross | coconut | Russia |

| DISEASES | VEGETABLE | FLOWERS | DOGS |
|---|---|---|---|
| Alzheimer's | squash | poppy | terrier |
| cancer | broccoli | rose | beagle |
| arthritis | peas | daisy | bulldog |
| anemia | cauliflower | lotus | collie |
| diabetes | eggplant | orchid | doberman |
| leukemia | asparagus | marigold | hound |
| hepatitis | onions | tulip | poodle |
| polio | peppers | carnation | husky |
| meningitis | radish | lilac | greyhound |
| smallpox | beans | sunflower | dachshund |
| influenza | parsley | geranium | chihuahua |
| malaria | rhubarb | daffodil | foxhound |
| cholera | parsnip | lavender | mastiff |
| tuberculosis | leeks | honeysuckle | spaniel |

| BIRDS | FISH | CITIES | VEHICLES |
|---|---|---|---|
| robin | trout | Boston | car |
| sparrow | cod | Ottawa | truck |
| hawk | shark | London | bicycle |
| bluejay | smelt | Chicago | automobile |
| pigeon | tuna | Munich | bus |
| pelican | catfish | Denver | van |
| duck | salmon | Berlin | taxi |
| seagull | swordfish | Tokyo | train |
| swan | herring | Houston | motorcycle |
| eagle | mackerel | Canberra | plane |
| crow | sturgeon | Dublin | boat |
| woodpecker | bass | Oakland | trolley |
| penguin | flounder | Detroit | ship |
| goose | pickerel | Pittsburgh | bike |

## Appendix C

### List of Nouns and Verbs Used in Lexical Class Simulation

| Nouns | Verbs |
|---|---|
| car | go |
| sun | drink |
| ball | eat |
| earth | read |
| tree | move |
| girl | walk |
| house | get |
| road | stop |
| bus | play |
| door | die |
| rabbit | hide |
| box | speak |
| bed | kick |
| dog | think |
| table | ride |
| toy | fly |

Received October 27, 2005
Revision received July 1, 2006
Accepted July 3, 2006 ■