Cognitive Science 25 (2001) 173–202

COGNITIVE SCIENCE

http://www.elsevier.com/locate/cogsci

Predication

Walter Kintsch*

Institute of Cognitive Science, University of Colorado, Boulder, CO 80309-0344, USA

Abstract

In Latent Semantic Analysis (LSA) the meaning of a word is represented as a vector in a high-dimensional semantic space. Different meanings of a word or different senses of a word are not distinguished. Instead, word senses are appropriately modified as the word is used in different contexts. In N-VP sentences, the precise meaning of the verb phrase depends on the noun it is combined with. An algorithm is described to adjust the meaning of a predicate as it is applied to different arguments. In forming a sentence meaning, not all features of a predicate are combined with the features of the argument, but only those that are appropriate to the argument. Hence, a different "sense" of a predicate emerges every time it is used in a different context. This predication algorithm is explored in the context of four different semantic problems: metaphor interpretation, causal inferences, similarity judgments, and homonym disambiguation. © 2001 Cognitive Science Society, Inc. All rights reserved.

1. Introduction

Most words in most languages can be used in several different ways so that their meaning is subtly or not so subtly modified by their context. Dictionaries, therefore, distinguish multiple senses of a word. Each sense of a word is typically illustrated with an example. To demonstrate the diversity of word senses, consider this selection form Webster's Collegiate Dictionary from the 30 senses listed for the verb *run* (intransitive):

the horse runs
the ship runs before the wind
the cat ran away
the salmon run every year

**E-mail address:* wkintsch@psych.colorado.edu (W. Kintsch).

PII: S0364-0213(01)00034-9

my horse ran last
the bus runs between Chicago and New York
a breeze ran through the trees
a vine runs over the porch
the machine is running
the colors run
blood runs in the veins
the ship ran aground
the apples run large this year.

The meaning of the predicate *run* is different in each of these examples: *the horse runs* in a different way than *the machine* or *the colors*—and *run away* and *run aground* are different yet, although all of these uses of *run* have a core meaning in common. The exact meaning of a predicate depends on the argument it operates upon. Predication creates new meanings in every context by combining the meaning of the argument and appropriately selected aspects of the meaning of the predicate. It is not the whole meaning of *run* that applies to *the vines running over the porch*, or *the blood running in the veins*, but only features[1] that are relevant to the argument of the predication.

Multiple senses are by no means rare, especially for verbs (hundreds of senses for semantically impoverished verbs like *give* and *take* have been distinguished). Dictionaries, however, don't really claim to be exhaustive in their listing of word senses. However, George A. Miller and his colleagues, with WordNet, have made an explicit attempt to catalogue word senses for use in linguistic and psychological research, as well as for artificial intelligence applications (Miller, 1996; Fellbaum, 1998). WordNet includes over 160,000 words and over 300,000 relations among them. For instance, the verb *run* has 42 senses in WordNet; in addition, 11 senses are listed for the noun *run*. Thus, WordNet is an extremely ambitious enterprise, hand-crafted with great care. To develop a word net for the entire English language is, however, also an extraordinarily difficult task, for not only can there be no guarantee that even the most dedicated lexicographer has not missed a sense of a word or some relation between words that may suddenly become relevant, but language change assures that new and unforeseeable word uses will forever develop. At best, such a system must remain open and continuously subject to modification.

The proposal made here is very different: there is no need to distinguish between the different senses of a word in a lexicon, and particularly the mental lexicon. The core meaning of each word in a language is well defined, but is modified in each context. Word senses emerge when words are used in certain special, typical contexts. Indeed, every context generates its own word sense. The differences between the contextual meanings of a word may be small or large, but they are always present. The decontextualized word meaning is nothing but an abstraction, though a very useful one. Specifically, in predication the meaning of the predicate is influenced by the argument of the predication.

A claim like this is of course empty unless one can specify precisely how a word meaning is defined and how it is contextually modified to give rise to various senses. Recent developments in statistical semantics have made this possible. Latent Semantic Analysis

(LSA) allows us to define the meaning of words as a vector in a high-dimensional semantic space. A context-sensitive composition algorithm for combining word vectors to represent the meaning of simple sentences expressing predication will be described below.

Lexical semantics is a diverse field. Hand-coding word meanings, as in WordNet (Miller, 1996), or hand-coding a complete lexical knowledge base, as in the CYC project (Lenat & Guha, 1990), has been the traditional approach. It is a valuable approach, but limited, both theoretically and practically. A catalogue is not a theory of meaning, and most cognitive scientists are agreed that to intuit meanings with any precision is a most difficult if not impossible task, but many don't care, because "the rough approximation (provided by a dictionary definition) suffices, because the basic principles of word meaning, (whatever they are), are known to the dictionary user, as they are to the language learner, independently of any instruction and experience." (Chomsky, 1987; 21).

The alternative to listing meanings is a generative lexicon in which word senses are not fixed but are generated in context from a set of core meanings. Approaches differ widely, however, as to what these core meanings are, how they are to be determined, and about the generation process itself. A long-standing tradition, with roots in the practice of logicians, seeks to generate complex semantic concepts from a set of atomic elements, much as chemical substances are made up of the chemical elements (Katz, 1972; Schank, 1975). A recent example is the work of Wierzbicka (1996), where word meanings are defined in terms of a small set of semantic primitives in a semantic metalanguage that rigorously specifies all concepts. Natural languages are interpreted with respect to that semantic metalanguage.

Alternatively, structural relations rather than elements may be considered the primitives of a semantic system. An early example of such a system (Collins & Quillian, 1969) was constructed around the IS-A relationship. A notable contemporary example of a generative lexicon of this type is Pustejovsky (1996). Pustejovsky employs a number of primitive semantic structures and develops a context sensitive system of symbolic rules focused on the mesh between semantic structure and the underlying syntactic form.

LSA contrasts starkly with semantic systems built on primitives of any kind, both logic- and syntax-based approaches. In the tradition of Wittgenstein (1953), it is claimed that word meanings are not to be defined, but can only be characterized by their "family resemblance." LSA attempts to provide a computational underpinning for Wittgenstein's claim: it derives the family resemblance from the way words are used in a discourse context, using machine learning, neural-net like techniques. The advantage of LSA is that it is a fully automatic, corpus based statistical procedure that does not require syntactic analysis. In consequence, however, LSA does not account for syntactic phenomena, either; the present paper shows how this neglect of syntax can be remedied, at least in a small way, with respect to simple predication.

In the present paper, LSA will be introduced first. Then, the predication algorithm will be discussed. Finally, a number of applications of that algorithm will be described to demonstrate that it actually performs in the way it is supposed to perform for a few important semantic problems: metaphor interpretation, causal inference, similarity judgments, and homonym disambiguation.

2. LSA: vectors in semantic space

LSA is a mathematical technique that generates a high-dimensional semantic space from the analysis of a large corpus of written text. The technique was originally developed in the context of information retrieval (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990) and was adapted for psycholinguistic analyses by Landauer and his colleagues (Landauer & Dumais, 1997; Landauer, Foltz, & Laham, 1998; Landauer, 1999).

LSA must be trained with a large corpus of written text. The raw data for LSA are meaningful passages and the set of words each contains. A matrix is constructed whose columns are words and whose rows are documents. The cells of the matrix are the frequencies with which each word occurred in each document. The data upon which the analyses reported below are based consist of a training corpus of about 11 million words (what a typical American school child would read from grade 3 through grade 14), yielding a co-occurrence matrix of more than 92,000 word types and more than 37,000 documents. Note that LSA considers only patterns of word usage; word order, syntax, or rhetorical structure are not taken into account.

Word usage patterns, however, are only the input to LSA which transforms these statistics into something new—a high-dimensional semantic space. LSA does this through dimension reduction. Much of the information in the original pattern of word usage is accidental and inessential. Why did an author choose a particular word in a specific place rather than some other alternative? Why was this particular document included in the corpus rather than some other one? LSA discards all of this excess information and focuses only upon the essential semantic information in the corpus. To tell what is essential and what is distracting information, LSA uses a standard mathematical technique called singular value decomposition,[2] which allows it to select the most important dimensions underlying the original co-occurrence matrix, discarding the rest. The matrix is decomposed into components associated with its singular values, which are ordered according to their importance. The 300 most important components define the semantic space. The dimensionality of the space is chosen empirically: a (roughly) 300-dimensional space usually compares best with human performance.

LSA thus makes the strong psychological claim that word meanings can be represented as vectors in a semantic space of approximately 300 dimensions. But not only word meanings are represented as vectors in this space, documents are similarly represented as well. And new documents—sentences, paragraphs, essays, whole book chapters—can also be represented as vectors in this same space. This is what makes LSA so useful. It allows us to compare arbitrary word and sentence meanings, determine how related or unrelated they are, and what other words or sentences or documents are close to them in the semantic space. A word of caution is necessary here: LSA knows only what it has been taught. If words are used that did not appear in the training corpus, or which are used differently than in the training corpus, LSA, not unlike a person, does not recognize them correctly or at all.

The measure that is used to calculate semantic relatedness is the cosine between two vectors. As a first approximation, readers unfamiliar with this concept may think of cosines as analogous to correlation coefficients. The cosine varies from -1 to $+1$, $+1$ denoting identity and 0 denoting unrelatedness. Most cosines between words are positive, though small negative values are common (the average cosine for randomly chosen word pairs is .02, with a standard deviation of .06). The more closely two words are related semantically, the higher their cosine. For instance, the singular and plural forms of a sample of 100 common nouns had a mean cosine of .66, with a standard deviation of .15.

A second measure that is often useful is the length of a vector, which, like the cosine, is defined mathematically. Intuitively, the vector length tells us how much information LSA has about this vector. Thus, the length of sentence vectors is generally greater than the length of word vectors, and the length of paragraph vectors is even greater. Words that LSA knows a lot about (because they appear frequently in the training corpus, in many different contexts) have greater vector lengths than words LSA does not know well. Thus, *horse* has a vector length of 2.49, while *porch* has a vector length of .59. Function words that are used frequently in many different contexts have low vector lengths (*the* and *of* have vector lengths of .03 and .06, respectively, and their cosine is .99—LSA knows nothing about them and cannot tell them apart since they appear in all contexts).

All we can do, however, is compare one vector with another. Inspecting the 300 numbers that compose it tells us little, for the dimensions of the semantic space are not identifiable. The only way we can tell what a given vector means is to find out what other words or sentence vectors are close to it. Thus, we can ask LSA to list the words closest to a given vector in the semantic space. The semantic neighborhood of a word tells us a great deal about the word. Indeed, we shall make considerable use of semantic neighborhoods below.

Often we have some specific expectations about how a vector should be related to particular words or phrases. In such cases it is most informative to compute the cosine between the vector in question and the semantic landmark we have in mind. In most of the examples discussed below when we need to determine what a vector that has been computed really means, it will be compared to such landmarks. Suppose we compute the vectors for *horse* and *porch*. To test whether what has been computed is sensible or not, we might compare these vectors to landmarks for which we have clear-cut expectations. For instance, the word *gallop* should have higher cosine with *horse* than with *porch* (the cosines in fact are .75, and .10, respectively), but the word *house* should have a higher cosine with *porch* than with *horse* (the cosines are .08 for *horse* and .65 for *porch*). This is not a very powerful test, but it is intuitively compelling and simple. What the particular landmarks are is not terribly important, as long as we have clear shared semantic expectations. Someone else might have chosen *race* instead of *gallop,* or *door* instead of *house,* or many other similar word pairs, with qualitatively equivalent results.

Readers can make their own computations, or check the ones reported here, by using the web site of the Colorado LSA Research group: http://lsa.colorado.edu. First select the appropriate semantic space and dimensionality. The semantic space used here is the "General Reading through First Year of College" space with 300 dimensions and term-to-term comparisons. To find the semantic neighborhood of *horse,* one types "horse" into the Nearest-Neighbor-box and chooses "pseudodoc". To find the cosine between *horse* and *gallop,* one types "horse" and into one box and "gallop" into the other box of the One-to-Many-Comparison.

LSA has proved to be a powerful tool for the simulation of psycholinguistic phenomena as well as in a number of applications that depend on an effective representation of verbal meaning. Among the former are Landauer and Dumais (1997), who have discussed vocabulary acquisition as the construction of a semantic space, modeled by LSA; Laham's (1997) investigation of the emergence of natural categories from the LSA space; and Foltz, Kintsch, & Landauer's (1998) work on textual coherence. To mention just three of the practical applications, there is first, the use of LSA to select instructional texts that are appropriate to a student's level of background knowledge (Wolfe, Schreiner, Rehder, Laham, Foltz, Landauer, & Kintsch, 1998). Second, LSA has been used to provide feedback about their writing to 6th-grade students summarizing science or social science texts (E. Kintsch, Steinhart, Stahl, Matthews, Lamb, and the LSA Research Group, 2000). The application of LSA that has aroused the greatest interest is the use of LSA for essay grading. LSA grades the content of certain types of essays as well and as reliably as human professionals (Landauer, Laham, Rehder, & Schreiner, 1997). The human-like performance of LSA in these areas strongly suggests that the way meaning is represented in LSA is closely related to the way humans operate. The present paper describes an LSA-based computational model, which accounts for another aspect of language use, namely, how meaning can be modified contextually in predication. The model is discussed first and illustrated with some simple examples of predication. Then the model is used to simulate several more complex kinds of language processing.

3. LSA-semantics: predication

The elements of an LSA-semantics are the word vectors in the semantic space. The standard composition rule for vectors in LSA has been to combine vectors by computing their centroid. Consider propositions of the form PREDICATE[ARGUMENT], where A is the vector corresponding to ARGUMENT and P is the vector corresponding to PREDICATE. According to the standard LSA practice, the meaning of the proposition is given by the centroid of A and P. In n dimensions, if $A = \{a_1, a_2, a_3, \ldots a_n\}$ and $P = \{p_1, p_2, p_3 \ldots p_n\}$, the centroid $(A,P) = \{a_1 + p_1, a_2 + p_2, a_3 + p_3 \ldots a_n + p_n\}$. This is unsatisfactory, because the vector P is fixed and does not depend on the argument A, in contradiction to the argument above that P means something different, depending on the argument it takes. Every time we use P in a different context A, we do not predicate all of P about A, but only a subset of properties of P that are contextually appropriate for A. This subset may be quite unusual and specific to that context (as in some of the examples above) or it may be large and diffuse, in which case the centroid may provide an adequate description of the meaning of the whole proposition.

To capture this context dependency an alternative composition rule, the predication algorithm, is proposed here. The essential characteristic of this algorithm is to strengthen features of the predicate that are appropriate for the argument of the predication. This is achieved by combining LSA with the construction-integration model of text comprehension (Kintsch, 1988, 1998). Specifically, items of the semantic neighborhood of a predicate that are relevant to an argument are combined with the predicate vector, in proportion to their relevance through a spreading activation process.

The 300 numerical values of a word vector define the meaning of a word in LSA. This is a context-free definition, or rather, meaning is defined with respect to the whole training corpus. Another way of representing aspects of the meaning of a word is by looking at its neighbors in the semantic space. The closest 20 or 100 neighbors tell us something about the meaning of a word, though not as much as the vector itself, which positions the word in the semantic space with respect to all other words. The closest neighbors, however, index some important features of the word and contexts in which it is used.

Consider a proposition of the form P(A), where P and A are terms in the LSA semantic space represented by vectors. In order to compute the vector for P(A), the construction-integration model of Kintsch (1988, 1998) will be used. Let {**S**} be the set of all items in the semantic space except for P and A. The terms I in {**S**} can be arranged in a semantic neighborhood around P: their relatedness to P (the cosine between each item and P) determines how close or far a neighbor they are. Almost all items in the space will be at the periphery of the neighborhood, with cosines close to 0, but some items will cluster more or less densely around P. Let cos(P, I) be the cosine between P and I in {**S**}. Furthermore, let cos(A, I) be the cosine between A and item I in {**S**}.

A network consisting of the nodes P, A, and all I in {**S**} can be constructed. One set of links connects A with all other nodes. The strength s(A,I) of these links is codetermined by how closely related they are to both A and P:

$$s(A,I) = f(\cos(A,I), \cos(P,I))$$

The function *f* must be chosen in such a way that $s(A,I) > 0$ only if I is close to both P and A. A second set of links connects all items I in {**S**} with each other. These links have low negative strengths, that is, all items I interfere with each other and compete for activation. In such a self-inhibiting network, the items most strongly related to A and P will acquire positive activation values, whereas most items in the network will be deactivated because they are not related to both A and P. Thus, the most strongly activated nodes in this network will be items from the neighborhood of P that are in some way related to A.

The *k* most strongly activated items in this network will be used in the construction of the vector for P(A). Specifically, the vector computed by the predication procedure is the weighted average of the *k* most activated items in the net described above, including P and A, where the weights are the final activation values of the nodes.

An example will help to clarify how this predication algorithm works. Consider the sentences *The horse ran*, which has the predicate *ran* and the argument *horse* (Fig. 1). In predication, we first compute the neighborhood of the predicate *ran*—a set of items ordered by how strongly related they are to *ran*. For the sake of simplicity, only three items from the neighborhood of *ran* are shown in Fig. 1: *stopped*, *down*, and *hopped*, which have cosines with ran of .69, .60, and .60, respectively. A network is constructed containing these three items, the predicate *ran* and the argument *horse*, as shown in Fig. 1. The neighbors are connected to *ran* with links whose strength equals the cosine between each neighbor and *ran*. Next, the cosines between the neighbors and the arguments *horse* are computed (which equal .21, .18, and .12, respectively) and the corresponding links are added in Fig. 1. We also add a link between *ran* and *horse*, with a strength value equal to the cosine between them (.21). Finally, inhibitory links are inserted between each pair of neighbor-nodes. Activation is then spread in this network (using the CI program described in Kintsch, 1998), until a steady state

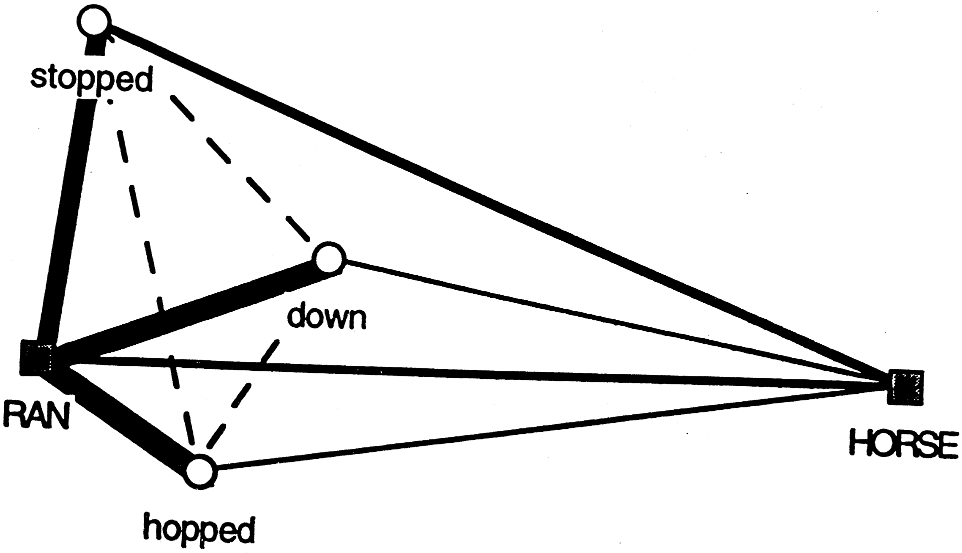


Fig. 1. A fragment of an integration network. Shown are the predicate *ran* and the argument *horse*, and three items from the neighborhood of *ran*. Solid lines indicate positive connections among nodes; dashed lines indicate inhibitory connections.

is reached. This integration process will select items that are close to *ran*, but also relevant to *horse*: in this case, *ran* and *stopped* will be most strongly activated, *down* will be somewhat activated, and *hopped* will receive no activation.

In the computations below two approximations are used:

(1) First, instead of constructing a huge network comprising all items in the semantic space, almost all of which would be rejected anyway, only the *m* closest neighbors of a predicate will be considered. The size of *m* varies because in order to select the terms that are most relevant to both A and P, a smaller or larger neighborhood must be searched, depending on how closely related A and P are. Thus, for most sentences combining familiar terms in expected ways, $m = 20$ works well, because terms related to A will be found even among the closest neighbors of P. For metaphors, on the other hand, where the predicate and argument can be quite distant, the crucial terms are usually not found among the top 100 neighbors of the predicate, and *m* needs to be larger, say 500 neighbors. A neighborhood of 1500, on the other hand, is too large: the terms selected from such a large neighborhood by the predication algorithm may only have a tenuous relationship to P and hence misrepresent it.

(2) Instead of using a weighted average of P and the *k* most relevant neighbors for the vector representing P(A), the weights will be neglected. Since only small values of *k* are used and the differences in activation among the top few terms are usually not dramatic, this computational shortcut has little effect. It greatly simplifies the calculation of predicate vectors, however. Since the most highly activated terms in the neighborhood of a predicate are those with the highest cosine to the argument, one merely has to add the *k* top-ranked terms to A and P. Thus, the vector for P(A) is

> computed as the centroid of P, A, and the *k* most activated neighbors of P (normally, LSA represents the meaning of P(A) simply as the centroid of P and A; the predication algorithm biases this vector by including *k* contextually appropriate neighbors of P).

The parameter *k* must neither be too small nor too large. If too few terms are selected, a relevant feature might be missed; if too many terms are selected, irrelevant features will be introduced. Values between $k = 1$ and $k = 5$ have been found to be most appropriate. When processing is more superficial, as in the similarity judgments discussed below, $k = 1$ gives the best results. If deeper understanding is required, *k*-values of 3 or 5 appear optimal. Selecting more than 5 terms usually introduces unwanted noise.

4. Some simple examples of predication

How satisfactory is the proposed predication algorithm? It is difficult to give a strong answer to this question. If there existed a closed set of sentences corresponding to P(A) propositions, one could obtain a random sample, compute the corresponding vectors, and find some way to compare the result with our intuitions about the meaning of these sentences. Instead, all that can be done is to show for a few simple sample sentences that the predication algorithm yields intuitively sensible results, and then focus on some semantic problems that provide a more demanding test. Metaphor interpretation, causal inferences, similarity judgments, and homonym disambiguation are some domains that allow a more specific evaluation as well as some comparisons between sets of experimental data and LSA predictions.

As an example of simple predication, consider the vectors corresponding to *The horse ran* and *The color ran*. First, the closest 20 neighbors to *ran* in the LSA space are computed. This list includes *ran* itself, since a word is always part of its own neighborhood. Then, the cosines between these 20 terms and *horse* and *color*, respectively, are calculated. A net is constructed linking *horse*, respectively *color*, with the 20 neighbors of *ran*, with link strength equal to the cosine between each pair of terms and inhibitory links between each of the 20 neighbors. The absolute value of the sum of all negative links is set equal to the sum of all positive links, to insure the proper balance between facilitation and inhibition in the network. This network is then integrated, resulting in final activation values for each of the 20 neighbors. These calculations are summarized in Table 1.

The vector for *The horse ran* computed by predication is therefore the centroid of *horse*, *ran*, and the 5 most highly activated terms from the neighborhood of *ran* (column 3 Table 1), which are *ran* itself and *stopped*, *yell*, *came* and *saw*. The vector representing the meaning of *The color ran* is obtained in the same way: it is centroid of *color*, *ran*, and *down*, *shouted*, *looked*, *rushed*, and *ran*. Thus, while *ran* has different senses in these two contexts, these senses are by no means unrelated: *ran* in the *color*-sense is still strongly tied to movement verbs like *rushed* and *hurry*.

It is important to note that just which words are selected from a neighborhood by the predication algorithm does not have to be intuitively obvious, and often is not (like the choice of *yell* for the *horse*-sense of *ran* above): what needs to be intuitively meaningful is the end result of the algorithm, not the intermediate steps. In many cases, items from a neighborhood

Table 1
The 20-term neighborhood of *ran*, with cosines and activation values for *horse* and *color*

neighbors of *ran*	cosine neighbor: *horse*	activation value (*horse*)	cosine *neighbor: color*	activation value (*color*)
ran	0.21	0.46	0.08	0.24
jumped	0.17	0.09	0.06	0.00
yelled	0.09	0.00	0.04	0.00
stopped	0.21	0.46	0.06	0.00
went	0.16	0.00	0.07	0.09
shouted	0.16	0.00	0.07	0.46
running	0.17	0.09	0.04	0.00
hid	0.16	0.00	0.04	0.00
cried	0.14	0.00	0.05	0.00
grabbed	0.14	0.00	0.03	0.00
saw	0.19	0.28	0.07	0.09
screamed	0.11	0.00	0.05	0.00
hurry	0.11	0.00	0.08	0.24
looked	0.15	0.00	0.09	0.39
yell	0.20	0.37	0.05	0.00
came	0.21	0.46	0.08	0.24
raced	0.19	0.28	0.06	0.00
rushed	0.13	0.00	0.09	0.39
down	0.18	0.19	0.11	0.70
hopped	0.12	0.00	0.02	0.00

are selected that seem far from optimal to our intuitions; they achieve their intended purpose because their vectors have weights on the abstract features that are relevant in this particular context.

To interpret the meaning of these vectors, they are compared to appropriate landmarks. Landmarks need to be chosen so as to highlight the intuitively important features of the sentence. *Gallop* was chosen as a landmark that should be closer to *horse ran* than *color ran*, and *dissolve* (a synonym for this sense of *run* according to WordNet) was chosen to be closer to *color ran* than to *horse ran*. This is indeed the case, as shown in Table 2. *Ran* by itself is close to *gallop*, but is essentially unrelated to *dissolve*. For *horse ran*, the relationship to *gallop* is strengthened, but the relationship to *dissolve* remains the same. The opposite result is obtained when *ran* is put into the context *color*: the relationship to *gallop* is weakened (but it does not disappear—*the color ran* has different connotations than *the color dissolved*) and that to *dissolve* is strengthened.

Choosing different landmarks, say *race* and *smudges*, yields a qualitatively similar picture. Varying the size of the semantic neighborhood (parameter *m*) has little effect in this example,

Table 2. Cosines between *ran*, *horse ran, color ran* and two landmarks

LANDMARKS	*ran*	horse ran	color ran
gallop	.33	.75	.29
dissolve	.01	.01	.11

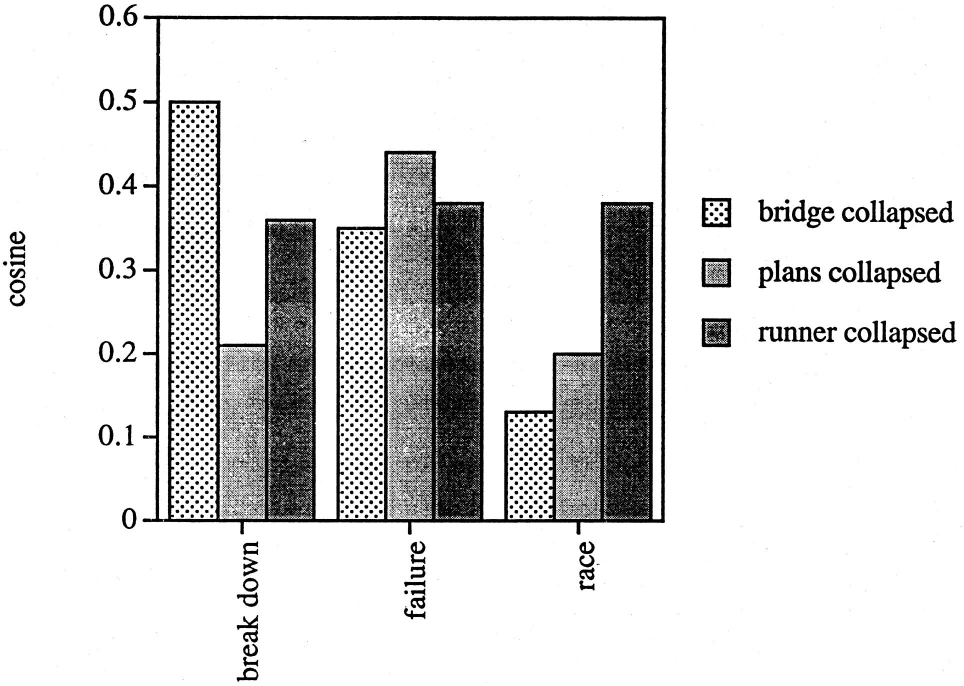


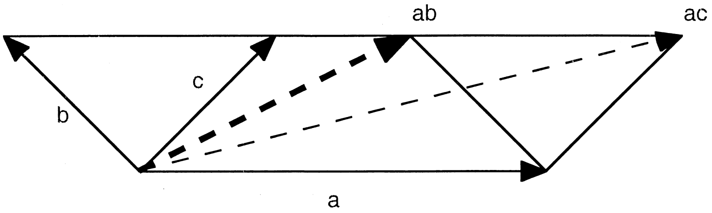
Fig. 2. Three sentences with the predicate *collapsed* compared to landmarks.

either. For $m = 50$ and $m = 100$ the cosines with the landmarks vary by a few percentage points, but the qualitative picture remains unchanged.

The horse ran is a more frequent expression than *The color ran*. In fact, in the corpus on which the semantic space used here is based, *color ran* appeared only once (in about 11 million words), whereas *horse ran* was present 12 times in the input data. Thus, LSA had little occasion to learn the meaning of *color ran*. Most of the interpretation that LSA gives to *color ran* is based on indirect evidence, rather than on direct learning. Indeed, if the semantic space is re-scaled with the single mention of *color ran* omitted, Table 2 remains basically unchanged. Thus, LSA can generate an intuitively plausible interpretation for a word sense it has never experienced: it does not need to be trained on the specific sense of *ran* in the context of *color*, it can generate the right meaning for the sentence on the basis of whatever else it knows about these words. As Landauer and Dumais (1998) have argued, vocabulary acquisition does not consist in learning about many thousands of separate word meanings, but in constructing a semantic space and embedding words and phrases into that space.

A second example of simple predication involving different senses of a word is shown in Fig. 2 where the meanings of the sentences *The bridge collapsed, The plans collapsed,* and *The runner collapsed* are compared. The landmarks were chosen in such a way that each sentence should be closest to one of the landmarks. The results confirm these expectations. The landmark *break down* is closest to *The bridge collapsed*. Appropriately, *plans collapsed* is closest to *failure*. For the *race* landmark, *runner collapsed* is closest. Thus, these results agree reasonably well with our intuitions about what these sentences mean. However, this is not the case when the sentence vectors are computed as simply the centroid of the subject and verb. In that case, for instance, *break down* is approximately equidistant to all three sentences.

The computation of sentence vectors by predication, or for that matter, by the centroid method, depends not only on the cosine between the word vectors, but also on how much information LSA has about these words. Technically speaking, a resultant vector is not only determined by the angle between its components in multi-dimensional space, but also by the length of the component vectors. Longer vectors have a greater influence on the centroid than shorter vectors. This is readily apparent in two dimensions, where it is a direct consequence of the law of parallelograms:



Scheme 1.

The direction of *ab*, the resultant of vector *a* and is not very different from that of *ac*, the vector sum of *a* and *c*, in spite of the fact that the angle between *a* and *b* is three times as large as the angle between *a* and *c*, because *a* is about three times as long as either *b* or *c*. In terms of LSA this means that if we take the centroid of two terms of unequal length, it will be weighted in favor of the term with the greater vector length. This has important consequences, as illustrated in the next example.

The vector for *bird* has length 2.04 while the vector for *pelican* has length 0.15, reflecting, in part, the fact that LSA knows a lot more about birds than about pelicans. When the two terms are combined, the longer vector completely dominates: the cosines between *bird* + *pelican* and the individual terms *bird* and *pelican* are 1.00 and .68, respectively. For comparison, the cosine between *bird* and *pelican* is .64. In other words, *pelican* doesn't make a dent in *bird*.

That result has serious consequences for predication. Of course, the centroid does not distinguish at all between *The bird is a pelican* and *A pelican is a bird*. Predication does, but with very asymmetric results. *A pelican is a bird* turns the pelican into a bird, almost totally robbing it of its individuality, as shown in Fig. 3. *Pelican is a bird* behaves like a *bird* with respect to the five landmarks in Fig. 3—closer to *sings beautifully* than to *eat fish* and *sea*! If we combine a short and a long vector, we get back basically the long vector—if the differences in vector length are as pronounced as in the case of *bird* and *pelican*, which differ by a factor of 13.

Fig. 4 illustrates what happens when the direction of predication is reversed. For LSA the meaning of *A bird is a pelican* is about the same as the meaning of *bird* by itself. Since LSA

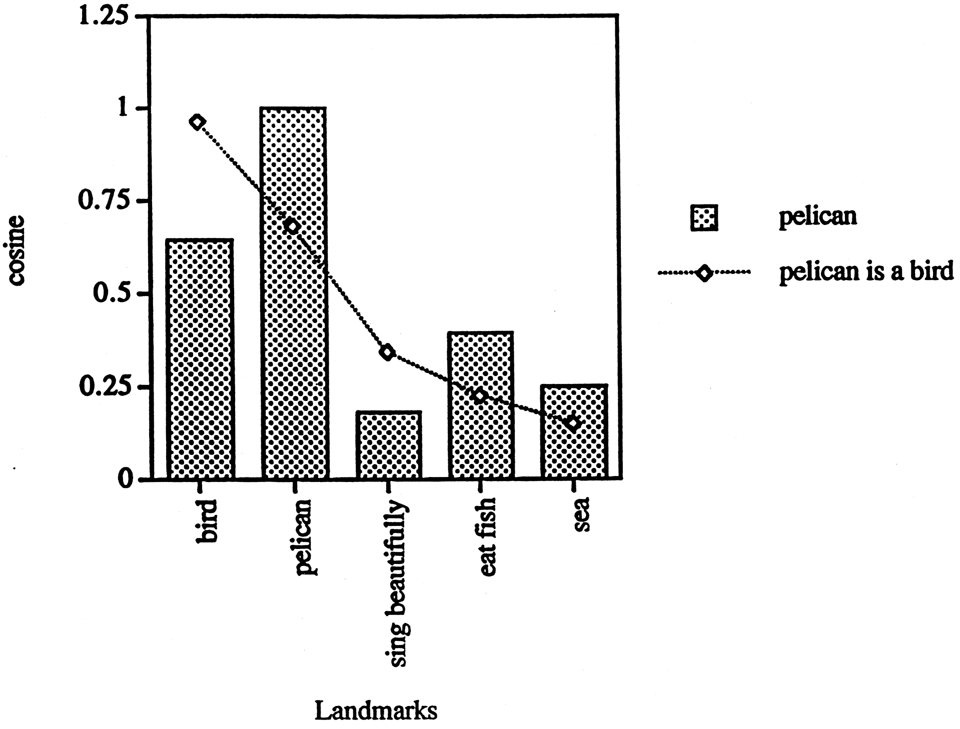


Fig. 3. Cosines between the vectors for *pelican* and *pelican is a bird* and five landmarks.

is vague about pelicans, we are not adding much meaning or knowledge to *bird* by saying it is a *pelican.*

A distinction needs to be made here between knowledge and information. LSA represents cumulative knowledge, not transient information. It measures not the new information provided by a sentence but what we already knew about its components. Predicating *pelican* about *bird* (*The bird is a pelican*) adds very little to our knowledge because we (and LSA) know very little about a *pelican,* other than that it is a kind of *bird*—it eats a little bit more fish than most birds do and sings a little bit less beautifully. The vector for *bird is pelican* is not very different from the vector for *bird.* In contrast, the sentence *The bird is a pelican* conveys information, because it excludes numerous other possibilities. On the other hand, *pelican is a bird* modifies our knowledge of *pelican* by emphasizing its general bird-features and de-emphasizing its individuality as a pelican. The language marks these distinctions. We say *The bird is a pelican,* providing information about some specific bird. Or we say *A pelican is a bird,* referring to the generic pelican. In the first case, we provide information, in the latter we provide knowledge. The informationally empty *The pelican is a bird,* and the epistomologically empty *A bird is a pelican* are not common linguistic expressions.

A similar distinction can be made between information and knowledge in a text. For each text, there exists relevant background knowledge with respect to the topic of the text. The text itself, however, usually involves information that is new and not already represented in the background knowledge. Thus, in a story unexpected things are supposed to happen,

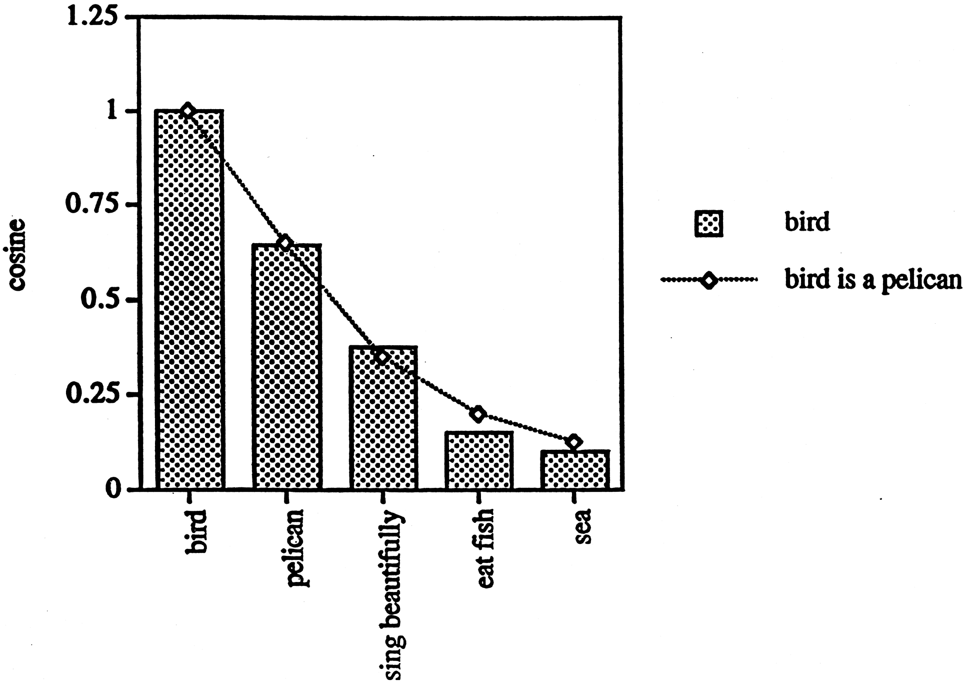


Fig. 4. Cosines between the vectors for *bird* and *bird is a pelican* and five landmarks.

building upon but different from what we already know. The textbase represents this new information in a text, while LSA provides a representation of the relevant background knowledge—what we already knew before we read the text. In comprehending the text, a representation is constructed—the situation model—that integrates the novel textual information and the pre-existing background knowledge.

5. Metaphors

As long as we are dealing with simple, familiar sentences, the results obtained with the predication algorithm often do not differ much from computations using the simpler centroid method. We need to turn to semantically more demanding cases to appreciate the full power of predication. The first of these cases is metaphor comprehension. This topic is discussed more fully in Kintsch (2000). It will be briefly summarized here, because it is crucial for an understanding of predication.

Experimental psycholinguistic evidence implies that metaphoric predication is just like any other predication in terms of the psychological processes involved (for reviews see Glucksberg & Keysar, 1994; Glucksberg, 1998; Gibbs, 1994). Thus, we need to show that applying the predication algorithm to metaphors in exactly the same way as it is applied to other sentences yields sensible interpretations of metaphors. Kintsch (2000) did just that. It showed that the interpretations of metaphors arrived at by the predication procedure agree

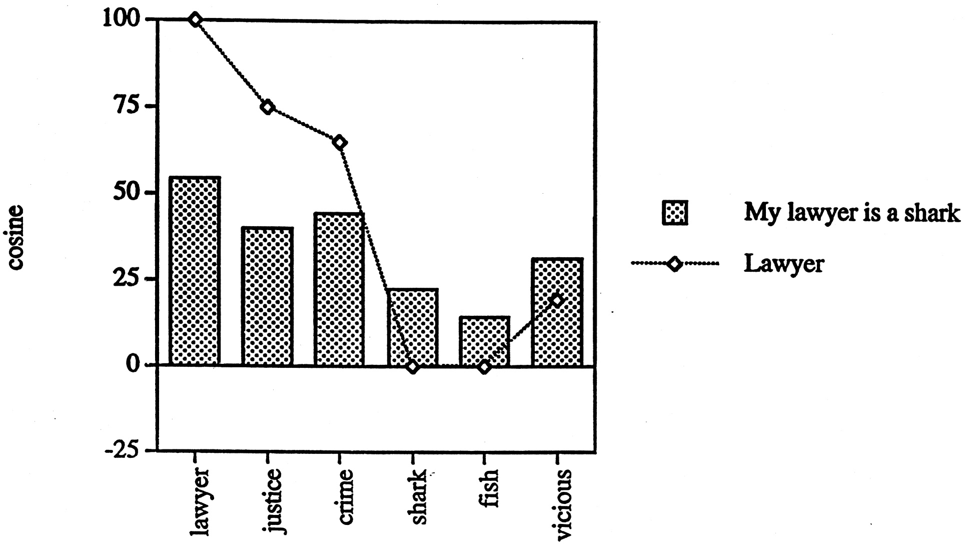


Fig. 5. *My lawyer is a shark* compared to landmarks.

with our intuitions reasonably well and, furthermore, demonstrated that some of the major phenomena in the experimental literature on metaphor comprehension can be simulated in this way, as direct consequences of the predication algorithm.

Glucksberg (1998) discusses in some detail the metaphor *My lawyer is a shark*. Fig. 5 presents the results of a comparison of the vector for *lawyer* alone and the vector computed by predication for *My lawyer is a shark* with landmarks chosen to highlight both the relevant and irrelevant features of the metaphor. By itself, *lawyer* is strongly related to concepts like *justice* and *crime*, not at all related to *shark* and *fish*, but *lawyer* is moderately related to *viciousness*. Predicating *shark* about *lawyer* changes this picture considerably. The *lawyer*-properties remain strong. The interesting thing is what happens to the *shark*-properties: *viciousness* is emphasized, in agreement with my intuitions that *My lawyer is a shark* means something like *My lawyer is vicious*. But it does not mean exactly that, otherwise we might have said so in the first place. There is also a little bit of *shark* and *fish* in it, and if we look at *bloodthirsty* or *tenacious*, we would see that elevated, too. Thus, the meaning of a metaphor is not fully captured by a literal paraphrase, but is richer, more expressive, and fuzzier than corresponding literal expressions.

Fig. 5 totally depends on the use of the predication algorithm. If the meaning of the metaphor is computed as the centroid of the words, the results do not make sense. The centroid of *lawyer* and *shark* is somewhere in semantic no man's land: more strongly related to *shark* and *fish* (cosines of .83 and .58, respectively) than to any of the *lawyer*-properties or to *viciousness*.

To compute the predication vector in Fig. 5 a semantic neighborhood of $m = 500$ was used. When a predicate and argument are semantically related, features that are relevant to both can usually be found even with lower values of m. Thus, for the calculations in the previous section m was typically set to equal 20. For metaphors, where argument and predicate can be quite unrelated in their literal senses, as in the present example, a larger semantic neighborhood must be searched to find three or five terms relevant to the predication. Furthermore, in order to insure that all terms selected are at least minimally related to both P and A, a threshold of two standard deviations above the mean for all words in the space was used. Since the mean cosine between all word pairs in the space is .02 and the standard deviation is .06, this meant that all cosines had to be at least .14. For $m < 100$, predication fails (the concept *lawyer* is not modified at all—there are no terms among the 100 closest LSA neighbors of *shark* that are semantically related to *lawyer*, that is, terms whose cosine with lawyer is at least .14, the threshold value we have chosen). For $m = 500$, 1000 or 1250, roughly equivalent results are achieved. As the neighborhood grows too large ($m = 1500$) the procedure begins to pick up random noise.

One of the salient facts about metaphors is that they are, in general not reversible. Reversed metaphors either mean something different, or they do not mean much at all. Kintsch (2000) has shown that the predication algorithm yields results in agreement with these intuitions. *Surgeon* is related semantically to *scalpel,* but not to *axe*; the reverse is true for *butcher*. *My surgeon is a butcher* has a cosine of .10 with *scalpel* and a cosine of .42 with *axe*; the reversed metaphor, *My butcher is a surgeon* has a cosine of .25 with *scalpel* and .26 with *axe*. On the other hand, reversing *My shark is a lawyer* does not yield any clear interpretation at all.

In order to assess the generality of the predication algorithm, Kintsch (2000) analyzed the first seven examples of nominal metaphors cited in Glucksberg, Gildea & Bookin (1982). Overall, the algorithm produced satisfactory results: the cosine between the metaphor and the relevant landmark was significantly higher than between the metaphor and the irrelevant (literal) landmark. The analysis failed in the case of *Her marriage is an icebox*—apparently because the LSA space used did not know enough about *iceboxes*, nor about *cold marriages*. This failure illustrates the need to distinguish between the adequacy of the underlying knowledge space and the predication algorithm itself. If LSA does not know something, it will perform badly with any algorithm; however, all one would presumably have to do in this case is to train LSA with a richer and more informative body of texts.

Two interesting phenomena about the time course of metaphor comprehension are also discussed in Kintsch (2000). First, it has been shown (Glucksberg, McGlone, & Manfredini, 1997) that the time it takes to comprehend a metaphor is increased when the literal meaning is primed. Thus, after reading *sharks can swim, My lawyer is a shark* requires more time to comprehend than after a neutral prime. The literal prime activates those features of *shark* that are related to *swim*. Hence, in the CI model, when the metaphoric sentence is being processed, the wrong features start out with a high activation value and it takes several integration cycles to deactivate the literal features and activate the metaphoric features. As the reverse of that, a metaphoric prime can slow down the comprehension of a literal sentence (Gernsbacher, Keysar, & Robertson, 1995). If *My lawyer is a shark* precedes *sharks can swim* in a sentence verification task, verification times are longer than if a neutral prime is used. The account that the predication model gives is essentially the same as in the first case. The metaphor activates features like *viciousness* and deactivates features like *fish,* so when *sharks can swim* must be verified, the wrong features are active and it requires several cycles of the integration process to deactivate these features and at the same time boost the activation of the features that are relevant to *swim*.

Thus, Kintsch (2000) goes beyond demonstrating that the predication model yields intuitively sensible interpretations of metaphors. It also shows that some of the major phenomena about metaphor comprehension in the psycholinguistic literature are readily accounted for within that framework. This is of interest, on the one hand, because it suggests that metaphor comprehension can indeed be treated in the same way as literal predication, and on the other hand, because it provides a good demonstration of how the predication algorithm extends the range of phenomena that LSA can account for.

6. Causal inferences

Many sentences imply causal consequences or causal preconditions. Thus, *The doctor drank the water* implies pragmatically (though not logically) the causal precondition that *the doctor was thirsty*, and *The student washed the table* implies the causal consequence that *the table was clean*. Usually, these are described as causal inferences, though some authors, such as Kintsch (1998), argue that the term inference is misleading in this context. When we read *The student washed the table* we do not usually, in addition, draw an inference that *the table is clean*. Rather, comprehending that sentence automatically makes available this information, without any extra processing. Long-term working memory assures that there will be a link between the sentence *The student washed the table* and its anticipated causal consequence, *the table was clean*. LSA provides a computational model of how long-term working memory functions in cases like these. Kintsch, Patel, & Ericsson (1999) have argued that the semantic space functions as the retrieval mechanism for working memory. Thus, if understanding *The student washed the table* involves computing its vector in the semantic space, closely related vectors such as *The table was clean* automatically become available in long-term working memory and may be subject to further processing (e.g., in a sentence verification task).

It remains to show that predication indeed delivers the right kind of results. Are sentence vectors in LSA, computed by predication, closer to causally related inferences than to causally unrelated but superficially similar sentences? Specifically, is the vector for *The student washed the table* closer to *The table was clean* than to *The student was clean*?

We are concerned with subject-verb-object sentences, that is, propositions of the form

$$\text{PREDICATE}[\text{ARGUMENT1}\langle\text{AGENT}\rangle, \text{ARGUMENT2}\langle\text{OBJECT}\rangle].$$

The corresponding syntactic structure is given by

$$\text{NP(N1)} + \text{VP(V} + \text{N2)}.$$

The syntax suggests that propositions of this form involve two separate predication operations: first V is predicated about N2, in the same way as discussed for simple predication above; then VP is predicated about N1.

Specifically, in Step 1 the neighborhood of size m ($m = 20$) for the predicate V is

Table 3
Cosines between four subject-verb-object sentences and causal inferences computed by centroid and predication

The student washed the table	the student was clean	**the table was clean**
centroid	.70	**.71**
predication	.62	**.83**
The student dropped the glass	the student was broken	**the glass was broken**
centroid	.76	.66
predication	.87	**.91**
The doctor drank the water	**the doctor was thirsty**	the water was thirsty
centroid	.59	*.86*
predication	**.83**	.78
The hunter shot the elk	the hunter was dead	**the elk was dead**
centroid	.66	.54
predication	.73	.70

Correct inferences are shown in boldface.

obtained. We select those terms from this neighborhood that are most relevant to N2: a network consisting of N2 and all neighbors, with link strengths equal to the cosine between N2 and each neighbor, is integrated and the k ($k = 5$) terms with the highest activation values are used to approximate the vector for (V + N2). In Step 2 the neighborhood is calculated for the complex predicate (V + N2), consisting of V, N2 and the k most relevant neighbors selected in Step 1. N1 is then used to determine the relevant terms from that neighborhood. The sentence vector, then, is approximated by the centroid of N1, V, N2, the k neighbors selected in Step 1, and the k neighbors selected in Step 2.

Thus, LSA, guided by a syntactic parse of the sentence, constructs a vector that represents the meaning of the proposition as a whole. To evaluate how well a predication vector captures the intuitive meaning of a proposition, causal inferences will be chosen as landmarks. For example,

The student washed the table—consequence—> *the table is clean*

or

The doctor drank the water—precondition—> *the doctor was thirsty*.

The vector representing the meaning of the sentence *The student washed the table*, computed by the predication procedure outlined above should be closer to the correct inference *the table is clean* than to the incorrect inference *the student is clean*.

As Table 3 shows, this is not generally the case when the meaning of the sentence is represented by the centroid of the three words. In fact, for the four examples analyzed here, the centroid makes the wrong inference in three cases. As we have seen above, the centroid is heavily influenced by vector length, so that semantically rich terms, like *hunter*, will always dominate semantically sparse terms, like *elk*. The predication procedure is able to overcome this bias in three of the four cases analyzed here. For instance, *the doctor drank the water* is strongly biased towards *the water was thirsty*, but predication manages to reverse that bias. Similarly for

The student washed the table—> the table was clean,
The student dropped the glass—> the glass was broken.

However, the wrong conclusion is reached in the case of

The hunter shot the elk—> the hunter was dead.

But even where predication fails to detect the correct inference, the cosine for *the elk was dead* increased twice as much as the cosine for *the hunter was dead* as a result of predication over a centroid based comparison. Apparently predication does something right, but may have failed for parametric reasons.

There are two parameters that need to be explored: the size of a predicate neighborhood was set at $m = 20$, and the number of most relevant terms chosen to represent the predicate vector was set at $k = 5$. Exploratory calculations suggest that these choices of parameter values are not necessarily optimal.

The calculations for *The hunter shot the elk* were repeated with the size of the predicate neighborhood $m = 100$. Increasing the neighborhood size, however, did not improve the performance of LSA in this case. Indeed, the bias in favor of *hunter dead* was slightly increased: the cosine between the sentence vector computed with $m = 100$ and *hunter dead* turned out to be .77, versus .72 for *elk dead*. What seemed to happen was that as the number of possible selections increased, the argument could select terms it liked that were, however, too distant from the predicate. For instance, when the neighborhood of *elk shot* is so large, rather distant terms like *bow* and *arrow* can be selected because they are so close to *hunter*, biasing the meaning of the sentence in inappropriate ways.

Better results were obtained by manipulating *k*, the number of terms used to approximate the predication vector. A smaller value of *k* than 5 which was used so far might work better, because in some cases the first three or four words that were selected from a neighborhood appeared to make more sense intuitively than the last ones. Hence the computations for *The hunter shot the elk* were repeated with $k = 3$. This resulted in some improvement, but not enough: the cosine between *The hunter shot the elk and hunter dead* became .69, versus .68 for *elk dead*.

Another possibility is that LSA just does not know enough about elks. If the more familiar word *deer* is substituted for *elk*, things improve. For $k = 3$, we finally get

The hunter shot the deer—> the deer is dead.

The cosine between *The hunter shot the deer* and *The deer is dead* is .75, whereas the cosine with *The hunter is dead* is now only .69.

An application of the predication algorithm to an existing set of examples of causally linked sentences uses materials developed by Singer, Halldorson, Lear, & Andrusiak (1992).

In this well-known study, Singer et al. (1992) provided evidence that causal bridging inferences were made during reading. In Experiment IV, sentence pairs in which the second sentence states a causal consequence of the first were compared with sentence pairs in which the second sentence merely follows the first temporally. An example of a causal sentence pair would be *Sarah took the aspirin. The pain went away.* An example of temporal succession would be *Sarah found the aspirin. The pain went away.* Their stimulus materials provide a further test of the ability of the predication model to explain causal inferences: the semantic relatedness between the sentences should be greater for the first than for the second pair of sentences.[3] Five of their stimuli were of the simple S-V-O form (or could be rewritten in that form with slight, inessential modifications) required by the present analysis; the other examples were syntactically more complex. The following sentence pairs could be analyzed:

Sarah took/found the aspirin. The pain went away.
(.89/.47)
Harry exploded/inflated the paper bag. He jumped in alarm.
(.33/.28)
The hiker shot/aimed-at the deer. The deer died.
(.74/.56)
Ted scrubbed/found the pot. The pot shone brightly.
(.45/.41)
The camper lost/dropped a knife. The camper was sad.
(.48/.37)

The numbers in parentheses below each line show the cosine values that were computed between the first and second sentence. In every case, causally related sentences had a higher cosine than temporally related sentences[4]. The average cosine for causally related sentence pairs was .58, versus .42 for temporally related sentence pairs.

Together with the examples presented in Table 3, the analysis of the stimulus materials from Singer et al. (1992) suggests that predication can give a satisfactory account of causal inferences in comprehension. Causally related sentence pairs appear to have generally higher cosines than appropriate control items, showing that the model is sensitive to the causal relation; the model does not yet tell us, however, that what it has found is a *causal* relation.

7. Judgments of similarity

Another domain where the predication model will be applied is that of similarity judgments. The cosines between concepts computed by LSA do not correlate highly with similarity judgments. Mervis, Rips, Rosch, Shoben, & Smith (1975; reprinted in Tversky & Hutchinson, 1986) reports similarity judgments for a 20 × 20 matrix of fruit names. The correlation between these judgments and the cosines computed from LSA is statistically significant, but low, $r = .32$. Similarly, for the data reported below in Table 4, the correlation between similarity judgments and the corresponding cosines is $r = .33$. These results appear to be representative.[5] In fact, there is no reason to believe that these correlations should be higher. It has been generally recognized for some time now that similarity judgments do not

Table 4. Rated similarity for pairs of animal names as a function of two instructional conditions (Anatomy and Behavior); after Heit and Rubenstein (1994)

			Data Anatomy	Data Behavior	cosine Anatomy	cosine Behavior
1	shark	trout	9.56	4.88	0.67	0.48
2	hawk	chicken	6.44	3.08	0.61	0.35
3	hawk	robin	7.29	4.60	0.64	0.37
4	shark	goldfish	5.75	3.60	0.46	0.38
5	mosquito	ladybug	5.43	3.53	0.22	0.15
6	bat	mouse	4.99	3.46	0.18	0.17
7	mosquito	grasshopper	4.43	3.01	0.90	0.90
8	bee	praying mantis	4.43	3.09	0.74	0.60
9	snake	turtle	4.07	3.14	0.52	0.79
10	bee	ant	5.15	4.35	0.81	0.48
11	snake	lizard	6.47	5.96	0.58	0.86
12	bat	giraffe	2.03	1.64	0.31	0.16
13	whale	bear	3.29	3.10	0.07	0.08
14	whale	tuna	5.56	5.87	0.40	0.26
15	whale	rabbit	2.51	2.83	0.03	0.12
16	snake	worm	4.90	5.25	0.41	0.58
17	bat	sparrow	4.81	5.17	0.48	0.19
18	bee	hummingbird	3.40	6.64	0.40	0.81
19	hawk	tiger	2.29	5.72	0.14	0.45
20	shark	wolf	2.32	6.08	0.14	0.25
21	mosquito	vampire bat	3.19	7.18	0.31	0.44

Cosines are computed after predicating either “anatomy” or“behavior” about each animal name.

directly reflect basic semantic relationships but are subject to task- and context-dependent influences. Each similarity judgment task needs to be modeled separately, taking into account its particular features and context.

It makes a difference how a comparison is made, what is the predicate and what is the argument. Tversky and Hutchinson (1986) point out that we say *Korea is like China*, but not *China is like Korea*, presumably because the latter is not very informative and thus violates Gricean maxims. The predication model provides an account for this observation. In *Korea is like China*, *Korea* is the argument and *China* the predicate; thus the resulting vector will be made up of *Korea* plus *China-as-relevant-to-Korea*—just as *A pelican is a bird* was made up of *pelican* plus *bird-as-relevant-to-pelican*. (Obviously, *is* and *is-like* do not mean the same, but this by no means irrelevant distinction must be neglected here). On the other hand, for *China is like Korea*, we compute a vector composed of *China* and *Korea-as-relevant-to-China*. The results are quite different. To say *China is like Korea* is, indeed, much like saying *Bird is pelican*—both statements are semantically uninformative! The cosine between *China and Korea-as-relevant-to-China* is .98, that is we are saying very little new when we predicate Korea about China in terms of the LSA semantics of the two concepts. However, to say Korea is like China, yields a cosine of only .77 between *Korea* and *China-as-relevant-to-Korea*. Our rich information about China modifies our concept of Korea successfully, whereas the little we know about Korea is so much like China anyway that it has not much of an impact on our concept of China.

The reason for the asymmetry in the previous example lies in the difference in the amount of knowledge LSA has about *China* and *Korea*: the vector length for the former is 3.22, versus 0.90 for the latter. When the vector length of the words being compared is more equal, the order of comparison may not make much of a difference. Thus, for *Buttons are like pennies* and *Pennies are like buttons*, the cosine between *buttons* and *pennies-like-buttons* is .32, which is about the same, .28, as the cosine between *pennies and buttons-like-pennies*. Even if there are differences in vector length when the words being compared are basically unrelated, order differences may be minor. For *Buttons are like credit cards* and *Credit cards are like buttons*, roughly equal cosines are obtained for the two comparisons (.04 and .07, respectively), in spite of the fact that *credit cards* has a vector length of 3.92, ten times as much as *buttons*.

The literature on similarity judgments is huge and complex and it is not at all clear at this point just which phenomena the predication model can account for and what its limits are. However, one systematic comparison with a small but interesting data set will be described here. Heit and Rubenstein (1994) report average similarity judgments for 21 comparisons with two different instructions. In one case, subjects were told to judge the similarity between a pair of animal names focusing on "anatomical and biological characteristics, such as internal organs, bones, genetics, and body chemistry". In another conditions, subjects were asked to focus on "behavioral characteristics, such as movement, eating habits, and food-gathering and hunting techniques" (p. 418). These instructions made a great deal of difference. For instance, *hawk-tiger* was judged highly similar with respect to behavior (5.72 on a 10-point scale) but not with respect to anatomy (2.29), whereas *shark-goldfish* were more similar with respect to anatomy (5.75) than with respect to behavior (3.60). Intuitively, one would expect such results—the question is whether LSA has the same intuitions or not.

In terms of the predication model, either Anatomy or Behavior were predicated about each animal name to be judged. What was compared was Animal$_1$-with-respect-to-behavior and Animal$_2$-with-respect-to-behavior on the one hand, and Animal$_1$-with-respect-to-anatomy and Animal$_2$-with-respect-to-anatomy on the other. Specifically, the semantic neighborhoods of both instruction sentences quoted above were determined, and the terms most relevant to the to-be-compared words were selected and combined with the word vector. Table 4 shows that LSA predicted the results of Heit and Rubenstein very well indeed. There are eight comparisons (rows 1–8 in Table 4) for which anatomical similarity was greater by at least one point than behavioral similarity. For these comparisons the cosines for the with-respect-to-anatomy comparisons were greater (equal in one case) than those for the behavioral comparison. There were four word pairs for which the behavioral similarity was rated at least one point higher than the anatomical similarity (rows 18–21). In all these cases the cosines for the behavioral comparisons were higher than for the anatomical comparisons. On the other hand, for the nine word pairs for which the empirical results were inconclusive (average ratings differed less than one point, rows 9–17), LSA matched the direction of the difference only in 4 cases. Average results are shown in Fig. 6, where the difference between Behavior minus Anatomy for the rating data as well as the cosines is plotted for items rated more similar in terms of behavior, neutral items, and items rated more similar in terms of anatomy.

The predictions reported here are based on computations using a semantic neighborhood of size $m = 50$ and a selection of one term from that neighborhood to be combined with the vector for each word ($k = 1$). Larger values of k yielded somewhat less satisfactory

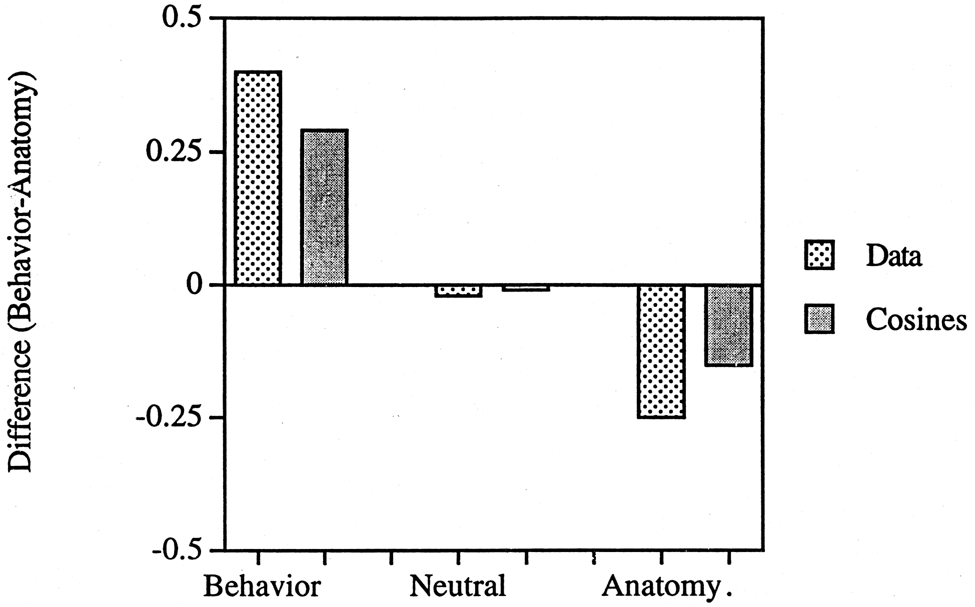


Fig. 6. Average differences between rating data and cosines for items rated more similar in terms of behavior, neutral items, and items rated more similar in terms of anatomy; data after Heit and Rubenstein (1994).

predictions. The correlation between the rating differences Anatomy-Behavior and the corresponding cosine difference were r = .62, r = .51, and r = .40 for $k = 1$, 3, or 5, respectively. Calculations based on a semantic neighborhood of $m = 20$, however, produced poor results. Only in very few cases could something relevant to the animal names be found in the anatomy neighborhood when only 20 terms were used. Thus, for this choice of parameter value behavior almost completely dominated anatomy, since even in a small neighborhood of behavior terms like *eating* were to be found, i.e., terms that are more or less relevant to all animals.

To account for the Heit & Rubenstein data, the predication model was needed; simply computing the cosine between two terms misses the context dependency of these judgments. However, similarity judgments are not always context dependent. A counterexample is given by Landauer & Dumais (1997), who were able to describe the choices people make on the TOEFL test (Test of English as a Foreign Language) simply by computing the cosine between the word to be judged and the alternatives among which a choice had to be made. For instance, what is the right response for the test word *abandoned—forsake, aberration,* or *deviance*? The cosines between the test word and the alternatives are .20, .09, and .09, respectively, so LSA chooses the right response. Indeed, LSA chooses the correct alternative 64% of the time, matching the mean percent correct choices of foreign students who are taking this test. It is easy to see why LSA alone works so well here, but why it must be used in conjunction with the predication algorithm for the Heit & Rubenstein data. The multiple choice alternatives on the TOEFL test do not provide a meaningful context with respect to which similarity can be judged because they are unrelated words (in the present example, the average cosine among the alternatives is .07), whereas in the examples discussed above, context plays a decisive role.

8. Homonyms

Predication modifies the predicate of a proposition in the context of its argument(s). However, the arguments themselves may have multiple senses or, indeed, multiple meanings. Homonyms are words that are spelled the same but have several distinct meanings—not just different senses. The vector that LSA computes for a homonym lies near both of its distinct meanings, something that is quite possible in a high-dimensional space. An example from Landauer (personal communication) will illustrate this point. Take the word *lead*. Its cosine with *metal* is .34 and its cosine with *follow* is .36; however, the cosine between *metal* and *follow* is only .06. *Lead* is related to two neighborhoods that are not related to each other. The average cosine between *lead* and ⟨*metal*, *zinc*, *tin*, *solder*, *pipe*⟩ on the one hand and ⟨*follow*, *pull*, *direct*, *guide*, *harness*⟩ is .48. But the average cosine between the words in these two distinct neighborhoods is .06.

If a homonym is used as an argument in one of its meanings in a sentence, do we need to adjust its meaning contextually similarly to the way it was done for predicates? Or does the predication procedure, which combines the vectors for the predicate and argument, automatically accomplish the meaning selection for arguments with multiple unrelated meanings? The latter appears to be the case. The LSA vectors for homonymous nouns contain all possible meanings (with biases for the more frequent ones), and appropriate predicates select fitting meanings from this complex. Some examples will illustrate this claim.

According to WordNet, *mint* has four meanings as a noun, one as a verb, and one as an adjective. Fig. 7 compares three of these senses with suitable landmarks. The sentences use the *candy* sense, the *plant* sense, and the verb sense of the homonym. The vector for *mint* is roughly equally related to the three landmarks that capture the different meanings of the homonym, with a slight bias in favor of the *candy* meaning. Vectors for these sentences were computed according to the predication procedure, and these vectors were compared with landmarks emphasizing one of these meanings: *chocolate* for the *candy* sense, *stem* for the *plant* sense, and *money* for the verb sense. Of course, once an argument is embedded in its sentence context, it is not possible to extract a separate vector for the argument; rather, the resulting vector represents the whole sentence. Fig. 7 shows that these sentence vectors have become very specific: they are strongly related to the appropriate landmarks, but only a little or not at all related to the inappropriate landmarks. However, the vectors for all three sentences remains related to the word *mint*. That is, *mint* still plays an important role in the sentence, not just the other two context words (cosines between *mint* and the three sentences are .40, .27, and .38, respectively, for the *candy*, *plant* and *coins* sentence).

The *mint*-example comes, in somewhat different form, from a priming experiment by Till, Mross, & Kintsch (1988), where it was the first item in their list of experimental materials. We also analyzed the next five of their examples with the predication procedure. For each homonym noun, two different disambiguating phrases were constructed, using as predicates

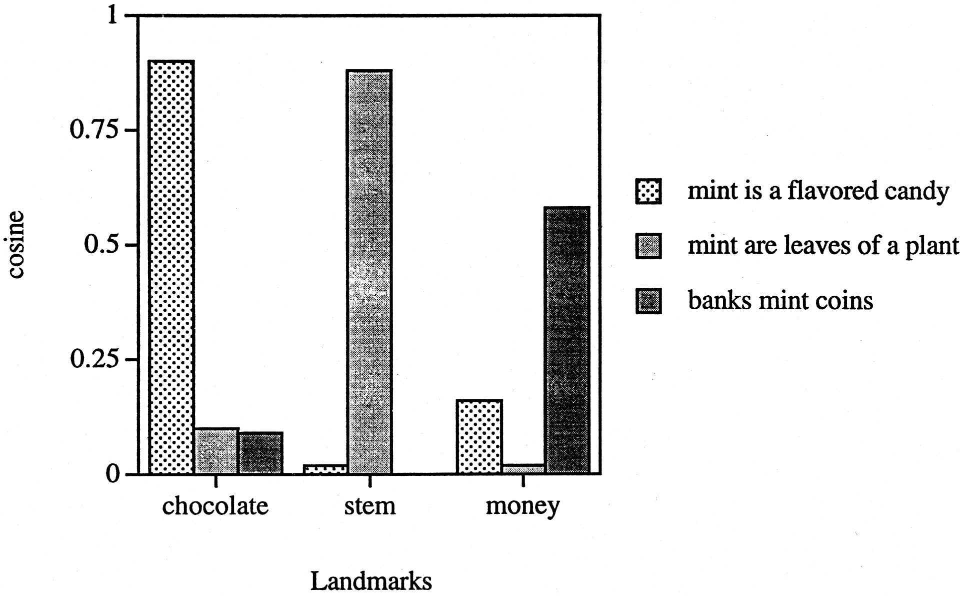


Fig. 7. The relationship between three landmarks and the vectors for three sentences expressing different senses of *mint*.

words from the definitions given in WordNet. Thus, for the homonym *pupil*, the phrases used were *pupil in school* and *pupil of the eye*. These brief contexts were clearly sufficient to determine the meaning of the homonym for educated adults. Would they suffice also for LSA? For each phrase a vector was computed with the predication procedure as explained above ($m = 50$, $k = 3$). This vector was then compared to landmarks also selected from WordNet—the category names to which each meaning was assigned in WordNet. The following test list of homonyms and landmarks was thus obtained: *ball/game-shot*; *bit/stable gear-pieces*; *pupil/learner-opening*; *dates/day-fruit*; *foil/sheet-sword*. Thus, 10 comparisons could be made. In 9 cases, the cosine between the phrase vector and the appropriate landmark was higher than the cosine between the phrase and the inappropriate landmark. The average cosine value for appropriate landmarks was .37, compared with .16 for inappropriate landmarks. The one failure that was observed was for the phrase *fencing foil*—the General Reading Space does not know the *fencing* meaning of *foil* (the two words have a cosine of $-.04$). Note that this indicates a lack of knowledge—not necessarily a failure of the predication algorithm.

Thus, it appears that the predication procedure is sufficient to contextualize words that have different meanings, in the same way as it handles words that have different senses. At least that is a reasonable hypothesis, pending further research. Gentner and France (1988) performed a series of experiments to investigate the comprehension of sentences in which the noun and verb were mismatched to make the interpretation of the sentence difficult. They concluded that under these conditions “verbs adjust to the nouns rather than the other way around.” Their results provide some support for the kind of model considered here.

9. Discussion

9.1. LSA-semantics

LSA is a new theory of word meaning. It is a theory that has considerable advantages over other approaches to lexical semantics, starting with the fact that it is a completely explicit mathematical formalism that does not depend on human intervention. It has also been strikingly successful in practical applications and has provided a solution to one of the toughest previously unresolved puzzles in the psychology of language—to explain the astonishing rate of vocabulary acquisition in children (Landauer & Dumais, 1997). Nevertheless, not everyone has been willing to take LSA seriously as the basis for a semantic theory. Too many of the traditional concerns of semantics have been outside the scope of LSA. The predication algorithm that is proposed in the present paper rectifies this situation to some extent. By combining LSA with the construction-integration model LSA can be made to account for the way in which syntax modifies meaning, at least for some simple, basic cases. At this point, it is not clear where the limits of the predication model are in this respect. However, even if the proposed approach eventually yields a full and satisfactory account of predication, other fundamental semantic problems remain for LSA, for example, concerning the classification and distinction among such semantic relations as hypernymy and hyponymy, meronymy, antonymy and so on.

Even though LSA is still only incomplete as a semantic theory, it nevertheless provides an interesting and promising alternative to the dominant conceptions of lexical semantics. Providing a computational model of how the syntactic and semantic context can modify and shape word meanings makes it possible to think about a lexicon in which word senses do not have to be distinguished. Words in LSA are represented by a single vector in a high-dimensional semantic space, however many meanings or senses they might have. The separate meanings and senses emerge as a result of processing a word in its syntactic and semantic context. They are therefore infinitely sensitive to the nuances of that context—unlike predetermined definitions, that will never quite do justice to the demands of complex linguistic contexts. Kintsch (1988, 1998) has argued that such a theory is required for discourse understanding in general; here, this argument is extended to the mental lexicon, and made precise through the computational power of LSA.

9.2. Centroid and predication

Centroid and Predication are two different composition rules for an LSA semantics. The analyses reported here indicate that in some cases predication gives intuitively more adequate results than centroid. This is clearly so for metaphoric predicates, causal inferences, and contextually based similarity judgments, and probably so for simple predication. But if predication is the better rule, why has the centroid rule been so successful in many applications of LSA, such as essay grading? It may be the case that the only time really important differences arise between these rules are in simple sentences out of a larger context, where specific semantic interpretations are at issue, as with metaphoric predication or causal inference. In the context of longer sentences or paragraphs, centroid and predication probably yield very similar results. The more predicates that appear in a text, the more neighborhood terms are introduced, so that their effects very likely would cancel each other. Enriching semantically a brief sentence can make an appreciable difference, as was demonstrated above, but enriching every phrase and sentence in a long text probably has very little effect and may get us right back to the centroid of the terms involved.

For the fine detail, predication seems superior to centroid. But the fine detail may not weigh very much when it comes to the meaning of a longer passage, such as an essay. Even for short sentences, centroid and predication often give very similar results. The vector for *The hunter shot the deer* computed by centroid and predication have a cosine of .97. Nevertheless, when we compare the centroid vector with *the hunter was dead* and *the deer was dead*, the centroid vector is much closer to the hunter being dead (cosine = .65) than the deer being dead (cosine = .35); when the vector computed by predication is compared with these inferences, on the other hand, it is closer to *the deer was dead* (cosine = .75) than to *the hunter was dead* (cosine = .69). Centroid and predication are almost the same for most purposes, except when we need to make certain subtle but crucial semantic distinctions.

9.3. The role of syntax

The predication algorithm presupposes a syntactic analysis of the sentence: one must know what is the predicate and what is the argument. People obviously use syntactic information in comprehension, but LSA does not. One could imagine how an existing or future syntactic parser could be combined with LSA to compute the necessary syntactic information. Ideally, one would like a mechanism that learns about syntactic structure in the same way as LSA learns about semantic structure, namely, through unsupervised learning. Such a system does not currently exist. There are of course many efficient syntactic parsers, either hand coded or based on supervised learning algorithms, that could be used in conjunction with LSA. However, since only very simple sentences are being analyzed here, little would be gained thereby at present.

9.4. Parameter estimation

The predication algorithm has two parameters, m, the size of the semantic neighborhood, and k, the number of items selected from the semantic neighborhood. (The parameter m is required only because of the calculational approximations used here—in principle one could always deal with the complete semantic neighborhood, though not very conveniently). For similarity judgments, especially when not very similar words are to be compared, such as *bat* and *giraffe*, a fairly large semantic neighborhood must be considered in predication ($m = 50$), but not too much from that neighborhood becomes integrated into the judgment ($k = 1$). For familiar subject-verb-object sentences, on the other hand, there is no need to work with such a large neighborhood since relevant terms could reliably be found within a much smaller neighborhood ($m = 20$). But predication had a much greater effect there than with similarity judgments—much more information from that neighborhood appeared to be integrated into the resulting sentence vector (the most convincing results were obtained for $k = 3$ or 5). Metaphors were different again, in that a much larger neighborhood had to be considered ($m = 500$), because the kind of argument relevant terms that predication selected from the predicate neighborhood tended not to be as strongly related to the predicate as in familiar sentences. For instance, for *My Lawyer is a shark*, most of the close neighbors of *shark* were irrelevant to *lawyer*, and one had to go way down the list to terms only moderately related to *shark* before finding *lawyer*-relevant terms for the integration. Furthermore, for metaphors, a threshold value (two standard deviations above the mean for random word pairs) was used to avoid selecting noise items, a precaution usually not necessary otherwise. However, further work will be needed to determine whether the use of a threshold is justified. When predication fails, should it fail because it cannot construct an interpretation (the threshold model) or because it constructs an off-the-wall interpretation (without a threshold)?

One may speculate that the way predication is instantiated in the brain is as a parallel activation process in which all neighbors sufficiently strongly related to the predicate are activated and tested for their relevance to the argument. All items are compared to both A and P, and the ones most strongly related to both are selected. How much of that information is then actually used in constructing the integrated sentence vector appears to be task dependent. When fairly deep understanding is required, as in causal inferences or metaphor understanding, quite a bit of the most relevant information from the predicate becomes integrated with the argument. On the other hand, in a more superficial task such as similarity judgment, less information from the predicate neighborhood is being used.

10. Conclusions

Assume that humans acquire knowledge in much the same way as LSA does: by keeping track of the events in their environment (their external as well as internal environment, and certainly not restricted to digitalized text) and deriving from it a high-dimensional semantic space by an operation like dimension reduction. This semantic space serves them as the basis for all cognitive processing. Often cognition directly reflects the properties of this semantic space, as in the many cases where LSA alone has provided good simulations of human cognitive processes. But often cognitive processes operate on this knowledge base, thus transforming it in new ways. One such case was explored in the present paper. Predication uses the LSA space to represent static word knowledge, but by putting a spreading activation net on top of it, it introduces an element of contextual modification that is characteristic of comprehension processes. Thus, by combining a comprehension model with an LSA knowledge base, a new and more powerful model was obtained. What we have, however, is still not a complete model of cognition. We may conjecture that a model of analytic thinking also uses an LSA knowledge base, but in ways as yet unknown. LSA by itself does not account for metaphor comprehension. But LSA in combination with the construction-integration model of comprehension, does. On the other hand, analogical reasoning, for example, is still beyond the scope of the LSA+comprehension model. To understand an analogy apparently requires more than finding some features in one domain that illuminate the other domain, as in metaphor comprehension, but requires systematic mapping and translation processes that require additional computational mechanisms than the constraint satisfaction process underlying the construction-integration model. Given the promise of the predication algorithm introduced here, it seems reasonable to keep looking for new ways to expand the use of an LSA knowledge base in modeling cognitive processes.

Notes

1. The term feature is used here in a non-technical sense.
2. A matrix can be decomposed into factors involving its eigenvalues (singular values) and its eigenvectors. The semantic space is constructed from the components of the matrix corresponding to its 300–500 largest eigenvalues. The method is described in detail and an illustrative example is given in Landauer & Dumais (1997).
3. I thank Murray Singer for providing me with these materials.
4. Parameter values used were k = 3, m = 50.
5. Similar low correlations are obtained for free-association matrices. For instance, r = .38 for the frequency of responses to a list of words related to *butterfly* (Deese, 1961) and the cosines between the respective words.
6. In one case the category name in WordNet was unknown to LSA and in another case it was barely familiar to LSA; the next word listed in the WordNet entry was used as a substitute.

Acknowledgment

This research was supported by the Army Research Institute and the J. S. McDonnell Foundation. I thank Tom Landauer, Eileen Kintsch, Dave Steinhart and the other members of the LSA Research Group for their help, comments, and stimulating discussions.

References

Chomsky (1987). Language in a psychological setting. *Sophia Linguistica, 22,* 1–73.

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.

Deese, J. (1961). From the isolated verbal unit to connected discourse. In: C. N. Cofer, Verbal learning and verbal behavior. New York, NY: McGraw-Hill.

Collins, A., & Quillian, M. (1969). Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior, 9,* 240–247.

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

Foltz, P. W., Kintsch, W., & Landauer, T. K. (1998). The measurement of textual coherence/with Latent Semantic Analysis. *Discourse Processes, 25,* 285–307.

Gentner, D., & France, I. M. (1988). The verb mutability effect: Studies of the combinatorial semantics of nouns and verbs. In: S. L. Small, G. W. Cottrell, & M. K. Tanenhaus, *Lexical ambiguity resolution: Perspectives from psycholinguistics, neuropsychology, and artificial intelligence*. San Mateo, CA: Kaufman. Pp. 343–382.

Gernsbacher, M. A., Keysar, B., & Robertson, R. W. (1995). *The role of suppression in metaphor interpretation*. Paper presented at the annual meeting of the Psychonomic Society, Los Angeles.

Gibbs, R. W. Jr. (1994). Figurative thought and figurative language. In: M. A. Gernsbacher, *Handbook of psycholinguistics* (pp. 411–446). San Diego: Academic Press.

Glucksberg, S. (1998). Understanding metaphors. *Current Directions in Psychological Science, 7,* 49–43.

Glucksberg, S., Gildea, P., & Bookin, H. B. (1982). On understanding nonliteral speech: Can people ignore metaphors? *Journal of Verbal Learning and Verbal Behavior, 21,* 85–98.

Glucksberg, S., & Keysar, B. (1990). Understanding metaphorical comparisons: Beyond similarity. *Psychological Review, 97,* 3–18.

Glucksberg, S., McGlone, M. S., & Manfredini, D. A. (1997). Property attribution in metaphor comprehension. *Journal of Memory and Language, 36,* 50–67.

Heit, E., & Rubenstein, J. (1994). Similarity and property effects in inductive reasoning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 20,* 411–422.

Katz, J. (1972). *Semantic theory*. New York: Harper & Row.

Kintsch, E., Steinhart, D., Stahl, G., Matthews, C., Lamb, R., & the LSA Research Group (2000). Developing summarization skills through the use of LSA-based feedback. *Interactive Learning Environments, 8,* 87–109.

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

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

Kintsch, W. (2000). A computational theory of metaphor comprehension. *Psychonomic Bulletin & Review, 7,* 257–266.

Kintsch, W., Patel, V. L., & Ericsson, K. A. (1999). The role of Long-Term Working Memory in text comprehension. *Psychologia, 42,* 186–198.

Laham, D. (1997). Latent semantic analysis approaches to categorization. In: M. G. Shafto, & M. K. Johnson, Proceedings of the 19th Annual Conference of the Cognitive Science Society (p. 979) Mahwah, NJ: Erlbaum.

Landauer, T. K. (1999). Learning and representing verbal meaning: The latent semantic analysis theory. *Current Directions in Psychological Science, 7,* 161–164.

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., & Laham, D. (1998). An introduction to Latent Semantic Analysis. *Discourse Processes, 25,* 259–284.

Landauer, T. K., Laham, D., Rehder, R., & 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, Proceedings of the 19th annual meeting of the Cognitive Science Society (pp. 412–417). Mahwah, NJ: Erlbaum.

Lenat, D., & Guha, R. (1990). *Building large knowledge-based systems*. Reading, MA: Addison-Wesley.

Mervis, C. B., Rips, L., Rosch, E., Shoben, E. J., & Smith, E. E. (1975). Relatedness of concepts. Unpublished data.

Miller, G. A. (1996). *The science of words*. NY: Freeman.

Pustejovsky, J. (1995). *The generative lexicon*. Cambridge, MA: MIT Press.

Schank, R. (1975). *Conceptual information processing*. Amsterdam: North Holland.

Singer, M., Halldorson, M., Lear, J. C., & Andrusiak, P. (1992). Validation of causal bridging inferences in discourse understanding. *Journal of Memory and Language, 31,* 507–524.

Till, R. E., Mross, E. F., & Kintsch, W. (1988). Time course of priming for associate and inference words in a discourse context. *Memory & Cognition, 16,* 283–298.

Tversky, A., & Hutchinson, J. W. (1986). Nearest neighbor analysis of psychological spaces. *Psychological Review, 93,* 3–22.

Wierzbicka, A. (1988). *The semantics of grammar*. Amsterdam: John Benjamins.

Wittgenstein, L. (1953). *Philosophical investigations*. New York: Macmillan.

Wolfe, M. B., Schreiner, M. E., Rehder, R., Laham, D., Foltz, P. W., Landauer, T. K., & Kintsch, W. (1998). Learning from text: Matching reader and text by Latent Semantic Analysis. *Discourse Processes, 25,* 309–336.