Reasoning with Vectors: A Continuous Model for Fast Robust Inference

Preprint of paper to appear in the Logic Journal of the IGPL.
Please visit and cite the canonical version at
http://jigpal.oxfordjournals.org/content/early/2014/12/03/jigpal.jzu028.short

Dominic Widdows
Microsoft Bing
dwiddows@microsoft.com

Trevor Cohen
University of Texas School of Biomedical Informatics at Houston
trevor.cohen@uth.tmc.edu

December 4, 2014

Abstract

This paper describes the use of continuous vector space models for reasoning with a formal knowledge base. The practical significance of these models is that they support fast, approximate but robust inference and hypothesis generation, which is complementary to the slow, exact, but sometimes brittle behavior of more traditional deduction engines such as theorem provers.

The paper explains the way logical connectives can be used in semantic vector models, and summarizes the development of Predication-based Semantic Indexing, which involves the use of Vector Symbolic Architectures to represent the concepts and relationships from a knowledge base of subject-predicate-object triples. Experiments show that the use of continuous models for formal reasoning is not only possible, but already demonstrably effective for some recognized informatics tasks, and showing promise in other traditional problem areas. Examples described in this paper include: predicting new uses for existing drugs in biomedical informatics; removing unwanted meanings from search results in information retrieval and concept navigation; type-inference from attributes; comparing words based on their orthography; and representing tabular data, including modelling numerical values.

The algorithms and techniques described in this paper are all publicly released and freely available in the Semantic Vectors open-source software

package.[1]

1 Introduction

Logic traditionally relies on discrete mathematical systems, rather than the continuous geometric representations that are foundational in mechanics and physics. These particular associations between branches of mathematics and their application domains are already well-developed in the writings of Aristotle,[2] and became *de facto* paradigms for centuries.

More recently, these paradigms have been challenged with many useful and startling results. Early in the twentieth century, discrete structures reappeared in physics partly due to the development of quantum mechanics (see e.g., von Neumann (1932)). Conversely, new sciences such as information retrieval have incorporated continuous methods into analyzing language (Salton and McGill, 1983; van Rijsbergen, 2004), a traditional province of discrete symbolic reasoning; and continuous-valued logics such as fuzzy logic have been developed and applied to many areas (Zadeh, 1988). As part of this boundary-crossing development, this paper demonstrates that continuous methods, more traditionally associated with linear algebra, geometry, and mechanics, can be used for logic and reasoning in semantic vector models.

The uncompromising nature of discrete symbolic logic is of course vitally important to many applications. Most obvious perhaps is mathematics itself, where a theorem must be proved to be true without doubt: mathematical proof does not embrace a "partially true" state! But in many cases, such demonstrable certainty in results is unattainable or undesirable. In our criminal justice system, "beyond reasonable doubt" is the benchmark. Psychological experiments have demonstrated that humans judge belonging to a category relatively, not absolutely: for example, people are quick to judge that a robin is a bird, but take longer to make the same judgment for a chicken or a penguin (Rosch, 1975; Aitchison, 2002), and the question "Was the archaeopteryx a bird?" is open to reasonable discussion. A variety of algebraic operators on vector spaces can be used to model such graded reasoning, with conceptual representations whose relationships and tendencies are learned from large amounts of training examples. Graded reasoning modelled in this way need not be probabilistic (in the sense of estimating the chance that an event will or will not take place), but it does need to quantify the notions of nearer and

[1] `http://semanticvectors.googlecode.com`

[2] For canonical examples, compare Aristotle's introduction of discrete symbols in logic for things that are either affirmed or denied (*Prior Analytics*, Bk I), as against the continuous treatment of physical magnitudes, locomotion, and time (*Physics*, Bk VI).

farther, stronger and weaker, and so on. As in quantum mechanics, probabilities of different outcomes can sometimes be derived from these measures of association.

This paper describes some of these methods and their applications, particularly Predication-based Semantic Indexing (PSI), which represents traditional "subject–predicate–object" relationships (such as "aspirin TREATS headache") using points and operators in a vector space. Section 2 introduces the traditional vector model for search engines, and how the vector sum and orthogonal complement can be used to add logical disjunction and negation to such systems. Section 3 explains why the robustness of such models stems directly from the mathematical properties of vectors in high dimensions: in particular, the way that in higher dimensions, the chances of accidentally confusing two known vectors become very small, even in the presence of considerable noise. These properties are crucial for the recovery of basic elements from compositional structures in Vector Symbolic Architectures (VSAs), which enrich the standard additive operations on vectors with a multiplicative binding operator, and are described in Section 4.

Section 5, the heart of the paper, illustrates a significant application of VSAs to logical deduction, a method we call Predication-based Semantic Indexing (PSI). For complex inferences with many pathways, PSI is sometimes especially robust (because reasonable doubt in one step is propagated throughout a chain of deductions), and computationally simple (because many logical combinations can be searched simultaneously using the same query, an example of entangled superposition whose mathematics is identical to that of quantum theory). This section also summarizes some experimental results-to-date that demonstrate that PSI is already successful as a large-scale working system.

Section 6 goes on to introduce several related applications, including the discovery of reasoning patterns, orthogonal negation in PSI models, type-systems, orthographic encoding, and the representation of structured tabular data and continuous quantities. Finally, Section 7 describes related work, in the hope of giving the reader a small taste of the wealth of research activity ongoing in this area, and Section 8 concludes the paper.

2 Vector Models for Search, and their Logical Connectives

In the early decades of information retrieval (roughly the 1950s to the 1980s), the challenge was modelled as "finding all the documents in a collection relevant to a given query". As document collections grew, this swiftly became

Table 1: A term–document matrix

	Doc_1	Doc_2	Doc_3	...	Doc_m
Term_1	M_{11}	M_{12}	M_{13}	...	M_{1m}
Term_2	M_{21}	M_{22}	M_{23}	...	M_{2m}
Term_3	M_{31}	M_{32}	M_{33}	...	M_{3m}
...	...	...	...	...	...
Term_n	M_{n1}	M_{n2}	M_{n3}	...	M_{nm}

untenable, especially since the rise of the World Wide Web: there is no use in returning to the user every webpage that is relevant in some way to the query *New York*. This observation naturally led to the conclusion that search results need to be ranked: not just divided into relevant and non-relevant, but presented to the user starting with the *most* relevant. Thus, unlike mathematical truth, relevance has to be treated as a graded quantity.

The vector model for search (Salton and McGill, 1983, Ch. 3) is a famous response to this problem, other solutions including the probabilistic and fuzzy Boolean models (Baeza-Yates and Ribiero-Neto, 1999). A vector model search engine is created by counting the number of times each term occurs in each document. (A 'term' is a word in the vocabulary, often subject to normalization or tokenization rules such as "disregard very frequent words like *the* and *of* ", "normalize to lower-case", "strip off endings so that *John's* becomes *John*", etc.) This counting process creates what is known as a *term-document matrix* (Table 1).

In the King James Bible corpus used as an example below, this gives 12818 terms and 1189 documents (treating each chapter as a document). Typically, modern corpora are much larger: representations need to scale to several million terms and sometimes billions of documents, which leads to a variety of challenges in sparse representations and distributed systems.

In this representation it's easy to see that each term can be treated as a row-vector and each document as a column-vector. Multiplying each element of such a vector by an appropriate weight is called scalar multiplication, the operation of adding two vectors together coordinate-by-coordinate is called the vector sum, and these operations are crucial in the standard definition of a vector space (Widdows, 2004, Ch 5).

However, since the search engine needs to be able to assess similarity between terms and documents, a new set of document vectors D is created with the same number of coordinates (same dimension) as the term vectors: each such document vector $d \in D$ is usually constructed as a weighted sum of the

rows corresponding to the terms in the document, using appropriate weights. Using the symbols from Table 1, the coordinates d_j of such a vector can be expressed as $d_j = \sum_{\mathrm{Term}_i \in \mathrm{D}oc_j} w_{ij} M_{ij}$, where w is a weighting function and the sum is over the set of terms occurring in the given document.

At search time, the system looks up the row for each known term in the query, and again adds these rows together with appropriate weights to construct a query vector q. The query vector is then compared with the set of document vectors, and returns the closest matches. The similarity between query vectors and document vectors is scored using a suitable overlap measure. In geometric models, the similarity measure used most typically is the cosine of the angle between a pair of vectors, which varies between $+1$ and -1 and is commutative. Rows of the term-document matrix are often normalized before the similarity-comparison step, using the Euclidean or L_2 norm.

The practice of using vector spaces to represent and retrieve documents goes back at least to the 1960's (Switzer, 1965), one of the most famous working implementations being the SMART system (Salton and McGill, 1983). The model has many modern descendants, including (for example) Apache Lucene, a widely-used open-source search engine.[3] Though simple and precise, the term-document matrices tend to be very sparse. To concentrate or distill the information in such matrices, dimension-reduction techniques are sometimes used, the best-known of which is singular-value decomposition. That is, an $n \times m$ matrix A can be factorized as the product of three matrices $\hat{U}\hat{\Sigma}V^*$, where the columns of $\hat{U}$ are orthonormal, V^* is an orthonormal matrix, and $\hat{\Sigma}$ is a diagonal matrix which can be ordered so that the values along the diagonal (called the singular values) are non-increasing. Taking the first k singular values and leaving the rest zero leads to a simplification of the original matrix which retains the most important variations in the original vectors in a vector space where each row has k coordinates, thus projecting the original space onto a lower-dimensional subspace. The application of this technique to a term-document matrix is called Latent Semantic Indexing or Latent Semantic Analysis (LSA) (Deerwester et al, 1990; Landauer and Dumais, 1997). LSA has been widely studied and several alternatives have been implemented.

Term vectors can be compared with one another, and the space can be searched for the nearest neighbors of any given term. An example is shown in Table 2, which shows the terms with the highest similarity with *fire* and *water* in a model built from the chapters in the King James Bible, using the Semantic Vectors package, a freely available open-source software package maintained

[3] See `http://lucene.apache.org`.

Table 2: The nearest words to *fire* and *water* in an LSA model built from 1189 chapters in the King James Bible. Real vectors, dimension 200.

fire		water	
fire	1.000000	water	1.000000
offer	0.600808	toucheth	0.566327
offering	0.569526	bathe	0.539766
sweet	0.468413	issue	0.517784
flour	0.466211	wash	0.513593
savour	0.465213	copulation	0.504446
burn	0.460434	clothes	0.479129
burnt	0.459722	rinsed	0.461367
meat	0.448330	separation	0.439121
shall	0.443111	unclean	0.422366

by the authors (Widdows and Cohen, 2010).[4]

As we come to apply vector models to more complicated reasoning tasks in the rest of this paper, it is important to bear in mind that vector model search can be done quite quickly even with large vocabularies. For example, with a vector of 200 dimensions whose coordinates are represented by 4-byte floating point numbers, the memory requirement is on the order of 1 kilobyte for each term or document vector (that is, 200 * 4 for the vector plus a generous 200 bytes for the name / identifier of the item being indexed). So approximately 1 million terms and their vectors can be represented in 1 gigabyte of main memory, which is well within the capabilities of standard hardware today. This can be searched linearly in comfortably under 3 seconds in experiments on one of the authors' laptops. This is not dramatically fast when compared with other keyword-based search engines, but when applied to tasks that involve chains of reasoning, it is dramatically faster than many symbolic alternatives.

For all of the above, there are corresponding probabilistic formulations. Rows in the term-document matrix are normalized using an L_1 norm (that is, the coordinates are rescaled so that they sum to 1, whereas with the Euclidean or L_2 norm, the sum of the squares of the coordinates is 1). Rows are compared using a probabilistic measure such as the Kullbeck–Liebler divergence, which measures the amount of information lost when approximating one distribution with another (Manning and Schütze, 1999, §2.2.5). Matrix decomposition is likely to use a suitable non-negative matrix factorization, so that the coordinates can be interpreted as probabilities (with some geometric methods such as LSA, negative coordinates arise naturally through projection onto a subspace, even if all the initial coordinates are non-negative). We emphasize

[4]See `http://semanticvectors.googlecode.com`.

this point here so that the reader is aware that there is a large body of work on probabilistic models for text in the statistical machine learning literature (see e.g., Blei (2012)), and that these models have some differences but also much in common with the geometric vector models.

2.1 Logical Operators in Vector Space Models

As well as adding term vectors together to form simple vector representations for queries or documents, such spaces can be explored to a limited extent using vector versions of the traditional logical NOT and OR operators. An appropriate NOT operator for two vectors a and b is the projection of a onto the subset orthogonal to b, that is:

$$a \text{ NOT } b = a - \frac{a \cdot b}{|a \cdot b|}\, b \tag{1}$$

For example, the term vector for *pharaoh* in the King James Bible has neighbors that are associated with two different Pharaohs in two different stories (that is, the story of Joseph in Genesis and the story of Moses in Exodus). The term *magicians* is only pertinent to the Exodus story, and projecting the vector for *pharaoh* so that it is orthogonal to the vector for *magicians* brings the vector for *joseph* from position 20 to the top of the list. A more systematic examination of such behavior demonstrated that removing unwanted terms using such projection techniques does a much better job at also removing neighbors and synonyms of the unwanted terms than just removing documents containing an unwanted term from search results (Widdows, 2003). The effect is particularly marked when multiple terms are removed by projecting the original query so that is is orthogonal to *all* of these related terms. This involves creating a subspace spanned by all the unwanted terms, so in vector logic, the disjunction a OR b becomes modelled by the plane spanned by a and b. The query vector is then projected onto the subspace orthogonal to the plane spanned by a and b, by subtracting appropriate linear multiples of the a and b vectors.

This leads to a logic of projection operators in vector spaces. Each projection operator projects onto a (linear) subspace; the conjunction of two operators projects onto the intersection if these subspaces; their disjunction projects onto the linear sum of these subspaces; and the negation is the projection onto the orthogonal complement. Such a logic has in fact been known since the 1930's, when it was introduced by Garrett Birkhoff and John von Neumann to model the logical relations between observables in quantum mechanics (Birkhoff and von Neumann, 1936). The relation between the Hilbert-space model for quantum mechanics and the vector models used in search engines is explored in

Widdows (2004, Ch 7), and much more fully in van Rijsbergen (2004), part of a growing list of applications of mathematics originally invented in quantum theory to problems in other disciplines including economics (Khrennikov, 2010) and cognitive science (Busemeyer and Bruza, 2012). A discussion of whether quantum mechanics itself is related to any of these human activities is beyond the scope of this paper: what is clearer is that the mathematics of high-dimensional vector spaces and their associated lattices and projections, pioneered long before quantum theory by Hermann Grassmann (1862), has considerable applications beyond its adoption as a model for the Hilbert-space formulation of quantum mechanics. An introduction to Grassmann's work in this area, its influence on the foundations of lattice theory, the relevance of these concepts to Boolean and quantum logic, and their impact on the design of search engines in recent years, is presented in Widdows (2004, Ch. 7,8).

3 Special Features of High Dimensional Vector Spaces

The robust retrieval of documents that contain many terms, using vectors that are linear combinations of the appropriate term vectors, depends on some properties of high-dimensional spaces that are somewhat counter-intuitive if we generalize too naively from experience in one, two and three dimensions. Most importantly, high dimensional spaces are sparse, in the sense that vectors chosen at random are likely to be almost orthogonal, and so the cosine of the angle between them is likely to be close to zero. The distributions of the similarity scores between randomly-chosen unit vectors in high dimensions is shown in Figure 1.

A mathematical explanation for these distributions, and the contrast with the low dimensions of our immediate physical experience, is given in Appendix A. Similar conclusions are reached by Kanerva (1988, Ch. 1) with binary-valued vectors, and by Sandin et al (2011) with ternary-valued vectors (that is, vectors whose coordinates are taken from the set $\{-1, 0, +1\}$). As Kanerva puts it "In terms of the sphere analogy, with any point and its complement taken as poles, almost all the space lies at or near the equator" (Kanerva, 1988, p. 19).

The observation that two randomly-chosen vectors are likely to be almost orthogonal has significant consequences for robust engineering and cognitive plausibility. Firstly, it enables vectors to "recognize themselves" from a set of randomly-assigned vectors, even in the presence of significant inaccuracy or noise. That is, suppose that X is a distinct set of (unit) vectors. Select an element $x \in X$ and add a random vector y of similar length. Assuming that y is almost orthogonal to x, normalization to unit length gives a vector $\frac{\sqrt{2}}{2}(x+y)$,

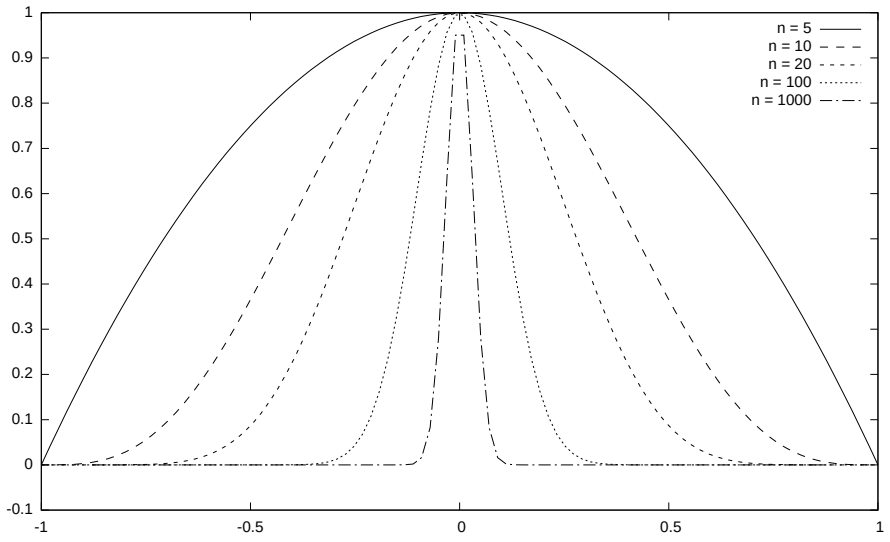

Figure 1: Distributions of cosine similarities between randomly-chosen vectors in high dimensions

whose similarity to x is roughly $\frac{\sqrt{2}}{2} \approx 0.7$, which in high dimensions is far above the region of similarities expected by chance (again, see Figure 1). It follows that out of all the elements in the set X, a noisy vector $x + y$ will still easily be recognized as a distorted copy of x (or y) rather than any other vector. As Kanerva puts it, "if 200 features (20%) can be determined with certainty, the remaining 800 features can be assigned zeros and ones at random ... and the test item can still be recognized" (Kanerva, 1988, p. 26).

This argument holds for any superpositions, not just noisy ones. So if x and y are two term vectors summed to create a document vector $x + y$, then either x or y as query vectors will recognize themselves as constituents of this document vector. The extent to which this behavior can be relied upon in practice is examined in Wahle et al (2012): with an element set of 1000 vectors, over 100 real vectors (dimension 500) can be added before a non-constituent is accidentally recognized as a constituent; and for binary vectors, this number can be increased to over 1000 constituents without increasing the physical memory used. This superposition property is also discussed in detail by Plate (2003, §3.2) and by Gallant and Okaywe (2013).

Finally, it is important to note that finding almost-orthogonal vectors in high dimensions is *easy* and *incremental*. This is very unlike the problem of finding exactly orthogonal vectors using an eigenvalue decomposition such

as the Singular-Value Decomposition, which is computationally intensive and requires prior knowledge of the system to be orthogonalized. In high dimensions, new elements can be created automatically with almost no effort, and almost no danger that they will clash with preexisting elements. This is crucial to the robustness of VSA's, because it means that we almost never encounter accidental similarities between elements that have nothing in common. Back in 1739, David Hume, often acknowledged as the greatest empiricist philosopher of modern times, postulated that our sensations and perceptions lead to mental representations called *impressions*, and that:

> every simple idea has a simple impression, which resembles it, and every simple impression a correspondent idea. (Hume, 1739, Bk I, Part I, §1)

Initially it may seem a daunting task to create such impressions without carefully organizing and curating the available representations, or alternatively running out of memory. Instead, the investigation of similarities in high dimensions teaches us that there are systems where such representations are cheap to create and robust to all kinds of noise and comingling.

4 Vector Symbolic Architectures

Thus far we have explained some of the uses of vector spaces, similarity measures, and the associated projection operators in creating relatively simple and clearly useful search engines. These behaviors depend on the properties of vectors in high dimensions, and as we saw in the previous section, the challenge in using these spaces does not lie in creating enough diversity for different objects to be easily distinguished and recognized: this is easy. Instead, the challenge is to represent complex relationships between impressions and ideas in a mathematical model where the natural tendency of a randomly chosen element is to be unrelated to almost all other elements.

Most obviously, the basic way of composing term vectors in vector-model search engines is to use the linear sum of vectors. This is commutative, in the sense that $a + b$ is always the same as $b + a$, and similarity-preserving, in the sense that the vector dot product $a \cdot (a + b)$ is large. These are sometimes desirable properties: for example, they enable a vector-model search engine to find matching documents from just a handful of query terms, without being disrupted by word order. But they are limited: one natural drawback is that if we keep adding vectors constructed in this fashion, they eventually all converge. For modelling the meaning of concepts and their combinations, this is

untenable in many situations: for example, there are many negational descriptors such as "former president", "fake ten-dollar bill", "red herring", whose effect is to combine into a concept that is clearly different from the unmodified original. Empirically, even when composition is not negational in this sense, when meanings depend on one another, cycles of repeated addition are only useful for a few iterations before semantic distinctions become eroded (Cohen et al, 2010a).

To model more general operations, we introduce at least one more operation, called *binding*, which will be written as $\otimes$. Good binding operators obey at least the following rules:

- For two vectors a and b, $a \otimes b$ is a new vector, that is usually not similar to either a or b, in the sense that $a \cdot (a \otimes b)$ and $b \cdot (a \otimes b)$ should be approximately zero.
- Binding has an inverse or at least an approximate inverse operator $\oslash$, called the *release* operator, such that $(a \oslash (a \otimes b)) \cdot b \approx 1$.

The release operator ensures that, given a bound product and one of its factors, we can recover the other factor with reasonable accuracy. The introduction of the binding and release operators turns a standard vector space (with the axioms described in Widdows (2004, Ch. 5)) into a structure increasingly known as a *Vector Symbolic Architecture* or VSA (Gayler, 2004; Levy and Gayler, 2008). The core operations in a VSA are summarized in Table 3. Note that several of these rules are currently somewhat vague: for example, superposition and measure overlap need to behave in such a way that $A \cdot (A + B)$ is 'large', but what do we mean by large? In practice, we mean that the result must be much larger than similarities we would find by chance. In this and many other cases, VSAs depend very fundamentally on the high-dimensional properties discussed in Section 3: in more dimensions, it is less likely that a large similarity between randomly-chosen vectors would occur by chance, and the required size of $A \cdot (A + B)$ would be correspondingly smaller. This example emphasizes the general point that the rules in Table 3 are mathematically a work-in-progress. They are not, properly speaking, axioms, because axioms should not contain vague terms like 'near to' and 'relatively large'. Instead, they are extremely useful guidelines, which may become (with suitable mathematical development) an axiomatic definition for a VSA as an algebraic structure.

Note that the definition of *Measure Overlap* is different from the standard definition of a scalar product in vector spaces: it is required that the outcome be a real number, even if the vector space itself uses a different ground field from the real numbers. (The 'ground field' of a vector space is in practice the

Table 3: The core algebraic operations in a Vector Symbolic Architecture

- **Generate Random Vector**. Creates a random vector that can be used to represent an elemental concept (that is, a concept used as an ingredient for assembling derived concepts). Random elemental vectors can be thought of as arbitrary names or labels for concepts. Random vectors in high dimensions are typically almost-orthogonal to one another.
- **Measure Overlap**. Measures the similarity between two vectors x and y, giving a real number, $x \cdot y$, typically between -1 (opposite), 0 (unrelated) and 1 (identical). The overlap between two randomly generated elemental vectors should be near to zero (or some other value that means 'no significant overlap' or geometrically orthogonal).
- **Superpose**. Takes two vectors x and y and generates a third vector $x + y$, such that $x \cdot (x+y)$ and $y \cdot (x+y)$ are relatively large. Superposition is sometimes called *bundling* in the literature. Superpositions can be weighted by any real number.

  Measure Overlap distributes over Superposition in the sense that $a \cdot \ (b + c) = (a \cdot b) + (a \cdot c)$ for all vectors a, b, c.
- **Bind**. Takes two vectors x and y and generates a third vector $x \otimes y$, such that $x \cdot (x \otimes y)$ and $y \cdot (x \otimes y)$ are usually near to zero. However, if y and $y\prime$ are close to each other, $x \otimes y$ should be close to $x \otimes y\prime$.

  Binding distributes over Superposition in the sense that $a \otimes (b + c) = (a \otimes b) + (a \otimes c)$ for all vectors a, b, c.
- **Release**. Inverse of bind, written as $x \oslash y$. Should behave in such a way that $(x \oslash (x \otimes y)) \cdot y \approx 1$.

number field from which coordinates for the vectors are drawn.) Note also that the symbol $\otimes$ is not intended to denote the tensor product itself, but (like the use of the addition symbol $+$ in group theory), denotes an operation which in some ways resembles multiplication, and must be defined in each particular VSA.

Now we give some examples of VSAs over the ground fields of real, complex, and binary numbers. These (and some other options) are all available in the Semantic Vectors package: for more details, consult Widdows and Cohen (2012).

4.1 Real Vectors

Real vectors (that is, vectors whose coordinates are taken from the real numbers $\mathbb{R}$) are used throughout machine learning and computational linguistics. The most standard similarity measure between real vectors is cosine similarity, which is simple, cheap, and works nicely with the natural definition of superposition as the sum of two vectors.

Binding, however, is less obvious. A simple and cheap suggestion is to use permutation of coordinates before superposing two vectors, since this has worked well for modelling word-order effects in semantic vector spaces (Sahlgren et al, 2008). That is, let $P_n(x)$ be the vector obtained by permuting each coordinate in x through n places, so that for example $P_1(x_1, x_2, x_3) = (x_3, x_1, x_2)$. Then $x \otimes y$ can be defined as $P_{-1}(x) + P_1(y)$. An exact inverse is easy to define, and the rules in Table 3 are satisfied. However, it also follows from this definition that $a \otimes b + c \otimes d = a \otimes d + b \otimes c$, so the way in which the vectors were combined becomes confused in the output, which can have unintended lossy consequences. (Rules of operator precedence are standard, so that $\otimes$ is applies before $+$.)

Other popular alternatives can be derived from the standard tensor product operation (in coordinates, this is the product of x as column vector with y as a row vector, giving an n-squared matrix). Tensor products have been investigated as a useful product operator in artificial intelligence (for background, see Plate (2003, Ch 1)), with renewed interest in recent years. The quadratic increase in dimensions can be a practical problem, one solution to which is the *circular convolution*, which sums the coordinates of the tensor product along each diagonal (from left to right, wrapping round at the left-hand edge), recovering a product that is itself a vector in dimension n. (This process is described in detail with diagrams in Plate (2003) and Levy and Gayler (2008).) The naive method for constructing the circular convolution of two vectors is thus to construct the tensor product and then sum along the diagonals, which takes quadratic or n^2 time: however, this can be improved by using Fast Fourier Transforms, which enable the convolution to be computed in $n \log(n)$ time. An approximate inverse (which becomes more exact in more dimensions) can be defined relatively easily. This method has also been used to model word-order effects and language n-grams, as a component of the BEAGLE model (Jones and Mewhort, 2007).

4.2 Complex Vectors

The use of complex numbers is ubiquitous in physics, but applications in informatics-related disciplines have been uncommon to date (often because we don't know intuitively what the imaginary coordinates might be useful for). However, there is at least one clear computational benefit to using complex numbers for a VSA: the circular convolution is much simpler. The circular convolution involves summing vectors in the 'frequency domain', and Fast Fourier Transforms can transform between the spatial domain and the frequency domain in $n \log(n)$ time. But for complex vectors this is already done: the vector in the frequency domain is effectively given by the arguments (angles) of each complex coordinate in modulus-argument form (Plate, 2003; De Vine and Bruza, 2010), so the circular convolution binding operator is as simple as summing the arguments of each pair of coordinates (which therefore has a simple and exact inverse).

This easy success motivates other questions, including whether the overlap measure should be the standard Hermitian metric, or the circular metric computed by comparing the angles of each coordinate (Plate, 2003, Ch. 4): and this question leads to further questions including the nature of normalization and the status of zero coordinates in sparse vectors. This is philosophically and mathematically a healthy situation, because it challenges us to consider the basic question of whether complex numbers should be treated as circular or rectilinear.

Practically, complex vectors have been successful in some reasoning experiments: due to the exact inverse and computational optimization, they can give models that are more accurate and faster than those created using real numbers; and when binding is involved, they are much faster to generate than models that use binary numbers.

4.3 Binary Vectors

The use of binary-valued vectors for representing concepts in a VSA has been developed and championed by Kanerva (1988, 2009). A binary vector of dimension n is simply a string of n binary digits or bits. The overlap between two of these bit-vectors is computed by counting the number of binary digits in common between the two vectors (which is n minus the Hamming distance, Hamming distance being the number of bits that differ). This can easily be normalized to a suitable overlap measure using the definition

$$x \cdot y = 1 - \frac{2}{n} \operatorname{HammingDistance}(x, y))$$

where x and y are binary vectors in n-dimensions. This gives a number between -1 (all bits are opposite) and 1 (all bits are the same), with 0 if exactly half the bits are the same and half are different.

Superposition is trickier: Kanerva (2009) recommends taking a majority vote between constituents to choose a 0 or a 1 for each position, and breaking ties at random: but when summing just two vectors, ties are very frequent, and the outcome is highly nondeterministic. This becomes a practical problem when we try to reconstruct chains of reasoning. We partly solve this problem by making the tie-breaking process deterministic, by making the seed for the random number generator depend in some procedural way on the vectors themselves. Such a process can also be used to define partial sums of binary vectors: so, for example, with suitably-weighted deterministic tie-breaking, we can define a binary vector $\lambda a + \mu b$ such that $a \cdot (\lambda a + \mu b) = \lambda$ and $b \cdot (\lambda a + \mu b) = \mu$. This process also enables us to define orthogonalization in the same way as for real and complex vectors.

The binding operation predominantly used is the bitwise XOR of coordinates, again following Kanerva (1988). This has the property of being self-inverse, so in this particular example of a VSA, the bind and release operations are the same.

Because of the need for bitwise operations such as pseudorandom tie-breaking, the use of binary vectors can be computationally costlier than real or complex vectors. However, these costs can be isolated to the appropriate indexing phases: querying or searching models can be made at least as fast than with real and complex numbers (or faster, because no floating-point arithmetic is necessary). Binary vectors have been shown in experiments to be even more robust to noise than real or complex vectors with the same storage requirements (Wahle et al, 2012).

4.4 Summary and Implementation Details

The VSAs described above are implemented, tested, and freely available as part of the Semantic Vectors open source package. As well as the choice between real, complex, and binary numbers, there are many other details and options available that can significantly change behavior. These include:

- The availability of permutation (as introduced by Sahlgren et al (2008)) as a complementary, non-commutative binding operation.
- The use of sparse vectors (especially for real, and sometimes for complex numbers) to represent elemental vectors. This enables models to scale

from hundreds-of-thousands to tens-of-millions of inputs (such as large corpora of documents) on a typical contemporary desktop machine.

- Deterministic generation of elemental vectors for any input. For example, for a given term, one can take a deterministic hash of its string representation, and use this as the random seed to generate pseudorandom elemental vectors. One immediate benefit of such deterministic approaches is that they reduce storage and synchronization requirements in distributed systems.

- Reuse of semantic vectors from one modelling process as elemental vectors for a subsequent modelling process. Reflective Random Indexing (Cohen et al, 2010a) applies this method to the term-document scenario in information retrieval, effectively predicting term cooccurrence in hitherto unseen documents.

For these and other reasons, the use of a particular VSA in practice always includes a considerable number of options and choices. These must be tailored to the requirements and resources available for a particular system. For example, for larger datasets binary VSAs preserve information more effectively, which can improve accuracy on reasoning tasks. However, the training phase is typically more time-consuming, which may be problematic for rapid prototyping, or for comparisons between several sets of models. In such situations the computational performance of complex vectors with the circular convolution as a binding operator is sometimes more desirable. VSAs are an important mathematical abstraction that describe the commonalities shared by many varied systems that can be used for distributional semantic computing.

5 Predication-based Semantic Indexing

In this section, we describe Predication-based Semantic Indexing (PSI), a technique we have developed that uses VSAs to represent structured data from a formal knowledge base. This is the main development described in this paper, because it is at the heart of using continuous mathematical methods to model formal systems traditionally manipulated using purely discrete mathematical approaches. It is a significant development in the field of distributional semantics, because since the early vector model search engines, most research in semantic vector models and distributional semantics generally has focussed on learning from free 'unstructured' natural language text, rather than formal 'structured' knowledge representations.

5.1 PSI fundamentals

Let X be a set of objects and let $\mathcal{R}$ be a set of relations on X, that is, each $R \in \mathcal{R}$ is a relation on X in the sense that $R \subseteq X \times X$. A semantic vector for each concept x is generated by summing the bound products of the elemental vectors for each relation xRy that involves x. In symbols,

$$S(x) = \sum_{R_j \in \mathcal{R}} \sum_{y \in X} W(R_j, x, y) E(R_j) \otimes E(y) \text{ for all } R_j, y \text{ such that } xR_jy.$$

Here $W(R_j, x, y)$ is some weighting function depending on (for example) the frequencies of the concepts x, y and the relation R_j. In practice, the inverse relation R-INV, defined by yR-INV x if and only if xRy, is always included in the set $\mathcal{R}$.

For readers less familiar with the algebra of relations and its application to semantics and knowledge representation, the rest of this section explains this process in a step-by-step fashion.

Predication-based Semantic Indexing takes as input a collection of concept–relation–concept triplets, sometimes called subject–predicate–object triples. Examples might be:

"Zeus PARENT_OF Hercules", "Insulin TREATS Diabetes".

Such relations have natural inverses, such as

"Hercules CHILD_OF Zeus", "Diabetes TREATED_BY Insulin".

In general, for a relation R we will write its inverse as R-INV.

Relations of this nature as a category of meaning were introduced by Aristotle (*Categories*, Ch. 7), and have come to the fore again in recent decades to support computational and electronic representations of meaning. Perhaps most famously, such triples are at the heart of the Semantic Web initiative (Berners-Lee et al, 2001). In the cognitive and biomedical literature they are sometimes referred to as "propositions" or "predications": they are thought to represent the atomic unit of meaning in memory in cognitive theories of text comprehension (Kintsch, 1998) on the basis of evidence from recall studies (for example, Kintsch and Keenan (1973)).

Algebraically, a subject–relation–object predication is often written xRy, where x is the subject, y is the object, and R is the relation. PSI takes as input a collection of such predications. For each such term x or relation R, its elemental vector will be written as $E(x)$ or $E(R)$. Elemental vectors are generated for each concept using one of the processes outlined in the previous section. (Alternatively, vectors from a previously-learned model can be reused

as elemental vectors.) Elemental vectors are also generated for each type of relation. No syntactic distinction between concept and relation types is made when selecting elemental vectors: this distinction comes later in the training phase depending on how these vectors are used in binding and superposition operations.

Semantic vectors for concepts are learned gradually by binding with the elemental vectors of related items: thus, if we have a predication xRy, the semantic vector for x, written $S(x)$, is incremented by the bound product $E(R) \otimes E(y)$. The same process is applied in reverse to $S(y)$ using the inverse relation R-INV, for which a different elemental vector $E(R\text{-INV})$ is generated (the elemental vector $E(R\text{-INV})$ need not be mathematically derived from $E(R)$ in any special way).

For example, encoding a single instance of the predication "Insulin TREATS Diabetes Mellitus" is accomplished as follows:

$$\begin{aligned} S(\text{insulin}) &\mathrel{+}= E(\text{TREATS}) \otimes E(\text{diabetes mellitus}) \\ S(\text{diabetes mellitus}) &\mathrel{+}= E(\text{TREATS-INV}) \otimes E(\text{insulin}) \end{aligned}$$

(The symbol "+=" is used here in the computing sense of "add the right hand side to the left hand side".) Thus, the semantic vector for diabetes mellitus encodes the assertion that it is treated by insulin, and the semantic vector for insulin encodes the assertion that it treats diabetes. Statistical weighting metrics may be applied at this point to temper the effect of repeated mentions of the same predication, and increase the influence of infrequently occurring concepts and predicates. The net result is a set of semantic vectors derived from the set of predications in which each concept occurs. On account of the reversible nature of the binding operator, this information can be retrieved. One would anticipate, for example:

$$S(\text{diabetes mellitus}) \oslash E(\text{TREATS-INV}) \approx E(\text{insulin})$$

If insulin occurs in many other predications, this retrieval will be approximate, but on account of the sparse and noise-robust properties of high-dimensional spaces explained in Section 3 we would still anticipate the vector

$$S(\text{diabetes mellitus}) \oslash E(\text{TREATS-INV})$$

being much closer to $E(\text{insulin})$ than to other unrelated elemental vectors in the space.

This explains the fundamentals of how PSI models are built, and how information from these models can be searched and recognized.

5.2 PSI Examples Using SemMedDB

Much of the completed work described in this paper uses PSI models built from SemMedDB (Kilicoglu et al, 2012). SemMedDB contains predications extracted by SemRep, a biomedical Natural Language Processing system that draws on both knowledge of grammatical structure and domain knowledge of the ways in which types of biomedical entities relate to one another (such as a drug can have side effects) to extract predications from the biomedical literature (Rindflesch and Fiszman, 2003). For example, SemRep extracts the predication "Insulin TREATS Diabetes Mellitus" from the phrase "Insulin lispro (Humalog), a novel fast-acting insulin analogue for the treatment of diabetes mellitus". To date, SemRep has extracted more than sixty million predications from the biomedical literature, which have been publicly released as the part of the SemMedDB database.

For example, Table 4 shows the results of nearest-neighbor searches for two composite query vectors in a PSI space derived from the SemMedDB database (specifically the June 2013 release). The PSI space is a 32,000-dimensional binary vector space that includes all predicates in SemMedDB, and all concepts occurring 500,000 or fewer times in the database. In addition, $\log(1 + \mathit{predication\ count})$ and the inverse document frequency of the concept concerned were applied as local and global weighting metrics respectively during training.

The diabetes-related search retrieves many standard treatments for diabetes, as well as a few tangentially related concepts in the ten nearest neighbors. The food-related query retrieves many types of food.

As VSAs use the same representational unit to represent both concepts and the relationships between them, the same decoding process that is used to retrieve concepts related in a particular way can determine the way in which a pair of concepts are related. So we would anticipate:

$$S(\text{food}) \oslash E(\text{vegetables}) \approx E(\text{ISA-INV})$$

This is indeed the case in the PSI space that generated the results in Table 4, where the nearest neighboring elemental vector representing a predicate to the vector product $S(\text{food}) \oslash E(\text{vegetables})$ is $E(\text{ISA-INV})$. So PSI is able to recall the fact that vegetables are a type of food, which follows naturally from the underlying mathematics.

Table 4: Nearest-neighbor Searches in a PSI Space derived from SemMedDB. Score = $x \cdot y = 1 - \frac{2}{n}\,\mathrm{HammingDistance}(x, y)$.

$S(\text{diabetes mellitus}) \oslash E(\text{TREATS-INV})$		$S(\text{food}) \oslash E(\text{ISA-INV})$	
score	**concept**	**score**	**concept**
0.039	biguanides	0.078	vegetables
0.035	insulin, regular	0.076	flavoring
0.034	acetohexamide	0.076	wheat
0.033	pancreas, artificial	0.074	rice
0.033	medical therapy	0.072	cow's_milk
0.032	islets of langerhans	0.072	cereals
0.031	drug eluting stent	0.071	soybeans
0.030	insulin, glargine, human	0.70	meat
0.030	insulin	0.67	peanuts dietary
0.029	tolbutamide	0.66	fruit

5.3 Analogical Inference with PSI

Perhaps more surprising, however, is that the noisy approximation of E(ISA-INV) recalled in this manner is sufficient to solve a proportional analogy problem of the form *a* is to *b* as *c* is to *?*. The nearest neighboring semantic concept vector to the cue vector $S(\text{food}) \oslash E(\text{vegetables}) \otimes E(\text{bourbon})$ is $S(\text{whiskey})$. This capacity for analogical reasoning, in which concepts are identified on the basis of shared structural relations, underlies many of the applications we will subsequently discuss.

In practice, accurate inference with a single cue is not always possible, but the signal preserved in a noisy approximation of a predicate can be amplified. One way to accomplish this is to explicitly retrieve the vector representation for the predicate concerned. However, superposing additional cues greatly amplifies the strength of this signal. For example,

$$S(\text{food}) \otimes E(\text{vegetables}) \oslash E(\text{bourbon}) + S(\text{food}) \otimes E(\text{flavoring}) \oslash E(\text{bourbon})$$

is closer to S(whiskey) than either of the above summands is individually. A more general analysis of this phenomenon is illustrated in Figure 2, which shows the similarity between $S(\text{whiskey})$ and individual cue vectors derived from the food types in Table 4, as well as the superposition of these individual cue vectors. As more cues are added, the similarity between the superposed product of these cues and $S(\text{whiskey})$ rapidly moves from the realm of the merely improbable to more than five standard deviations above the mean

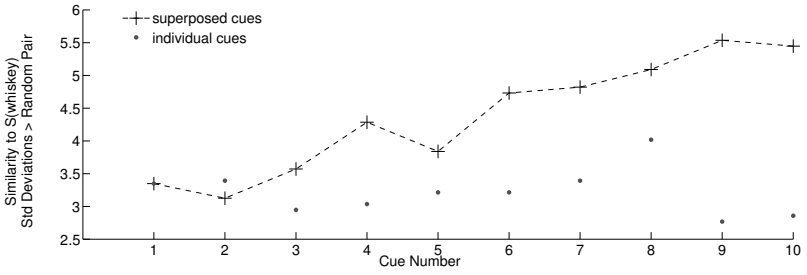

Figure 2: Cue Superposition Amplifies Analogical Retrieval

anticipated between random vectors of this dimensionality. Of note, this degree of similarity is higher than that produced by any individual cue.

Analogical inference can extend across longer predicate pathways also. Consider the case of Major Depressive Disorder (MDD). The product of $S(\mathrm{MDD})$ $\oslash$ $E(\text{TREATS-INV})$ approximates the elemental vectors representing instances of things that treat MDD, so we would anticipate that the vector for $S(\mathrm{MDD})$ $\oslash$ $E(\text{TREATS-INV})$ $\otimes$ $E(\text{ISA-INV})$ would approximate the semantic vectors representing classes of treatments for depression (as the operation $S(\mathrm{CLASS})$ $+= E(\text{ISA-INV}) \otimes E(\mathrm{INSTANCE})$ will have occurred during training).

For example, the five nearest neighboring semantic vectors to the vector product $S(\mathrm{MDD}) \oslash E(\text{TREATS-INV}) \otimes E(\text{ISA-INV})$ in the previously used PSI space represent the concepts "reuptake inhibitors", "antidepressive agents, second-generation", "antidepressive agents", "psychotropic drugs" and "selective serotonin re-uptake inhibitor", all of which are categories of agents used to treat depression.

Furthermore, it is possible to infer the dual predicate path connecting two concepts from an example pair. The vector representation of $E(\text{TREATS-INV}) \otimes E(\text{ISA-INV})$ is more similar to the vector product $S(\mathrm{MDD}) \oslash S(\mathrm{reuptake\_inhibitors})$ than the vector representation of any other pair of predicate vectors. As is the case with individual predicates, analogical retrieval can be accomplished without explicitly retrieving the predicates concerned. Instead of using an explicit predicate vector, a cue pair of vectors can be selected whose bound product closely approximates the vector representation of the predicate or predicate pathway concerned. Several such products of cue pairs can be superposed, leading to a combined 'holistic' or 'entangled' representation that cannot be decomposed into the product of any two individual cues (Cohen et al, 2011).

6 Applications of VSAs and PSI

This section presents a variety of new topic areas to which PSI has been applied more recently.

6.1 Discovering Discovery Patterns

This section explores further the use of analogical reasoning in PSI space as a means to infer previously unseen relationships between biological entities, with a focus on the recovery and discovery of potentially therapeutic relationships.

We have applied PSI to knowledge extracted by SemRep to infer therapeutic relationships between pharmaceutical agents and human diseases (Cohen et al, 2012a,b,c), using an approach we call *discovery-by-analogy*. The idea underlying this approach is to constrain the search for potential treatments to those that are connected to the disease in question along reasoning pathways suggesting therapeutic relationships. The idea of using reasoning pathways consisting of predicates, called *discovery patterns*, was developed by researchers in the field of literature-based discovery (Hristovski et al, 2006). Rather than manually constructing these pathways as had been attempted previously, we were interested to see whether we could both infer and apply them using PSI-mediated analogical reasoning.

For example, in the PSI space we have utilized for our examples so far, the vector product $S(\text{insulin}) \oslash S(\text{diabetes mellitus})$ generates a vector with relatively high similarity to the vector product $S(\text{INTERACTS WITH-INV}) \oslash S(\text{ASSOCIATED WITH-INV})$, suggesting that a set of biological entities exists that is both ASSOCIATED WITH diabetes, and INTERACTS WITH insulin. When applying this pattern to the semantic vector for asthma the closest semantic vector for a pharmaceutical agent to the resulting vector represents dexamethasone, a commonly employed asthma therapy.

In our discovery experiments, the best results were obtained by evaluating multiple reasoning pathways simultaneously, with around one third of the total held-out therapeutic relations ranked in the top one percent of predictions for each of the types of cancer evaluated. The five most popular reasoning pathways across large numbers of known treatment pairs were inferred, and used together to 'rediscover' a held-out set of treatments. These reasoning pathways were combined using the quantum disjunction operator to create a compound search expression which was used to retrieve treatments connected to other diseases across one, or several of these pathways (Cohen et al, 2012c,b). Further improvements in performance were obtained by extending the length

of the predicate pathways concerned to include popular triple-predicate pathways also (Cohen et al, 2012a), allowing for the recovery of around ten percent more of the held-out set within the top one percent of predictions across all cancer types. This was accomplished by creating *second-order semantic vectors* for diseases, as the superposition of the semantic vectors of concepts that occurred in a predication of predicate type ASSOCIATED WITH with the disease in question, and using these as the starting point for inference instead of the disease in question.

In our most recent work, we have evaluated the ability of these models to predict the results of a high-throughput screening experiment in which over a thousand pharmaceutical agents (that is, active pharmaceutical substances) were evaluated for their activity against prostate cancer cells that are resistant to commonly used hormonal therapies (Cohen et al, 2014). Of these agents, only a small number actively inhibited the growth of these particular cancer cells: of the 1398 evaluated agents that were represented in our PSI space, only 68 slowed the growth of cancer cells to 1.5 standard deviations less than the average across all agents tested (this average value was indistinguishable from negative controls). Table 5 presents the 20 highest ranked predictions generated by applying discovery-by-analogy to a PSI space that was deprived of knowledge of any direct relationships between pharmaceutical agents and types of cancer. Specifically, any predication involving a direct relationship between a pharmaceutical agent and a cancer type was withheld, as were any TREATS relationships, to simulate discovery. In addition, only a subset of predicates were encoded (see Cohen et al (2014) for further details). Aside from these restrictions, parameters were identical to those used to generate the space used in Table 4.

As shown in the table, the vast majority of the top 20 predictions occurred in TREATS relationships with hormone-refractory prostate cancer in the predication database (most likely indicating they had an inhibitory effect on a prostate cancer cell line, or showed efficacy in an animal model or clinical trial — though these may also be due to natural language processing errors, or an effect observed in combination with other drugs), once again illustrating the ability of discovery-by-analogy to recover held-out TREATS relationships. In addition, seven of the top 20 predictions were amongst the small number of agents that were effective against prostate cancer cells in our empirical experiments, a yield of active agents approximately seven times higher than would be anticipated if 20 agents were selected at random.

These experiments also illustrate the efficient (albeit approximate) reasoning that PSI mediates. For example, with relevant vector stores retained in

Table 5: Twenty Top Predicted Therapeutic Relationships for Hormone-refractory Prostate Cancer

rank	agent	TREATS	Active
1	gefitinib	✓	
2	paclitaxel	✓	✓
3	gemcitabine	✓	✓
4	resveratrol	✓	
5	methotrexate	✓	
6	sorafenib	✓	
7	tretinoin	✓	
8	cyclophosphamide	✓	
9	cyclosporine		✓
10	epigallocatechin gallate		
11	sirolimus	✓	
12	fluorouracil	✓	
13	troglitazone		
14	dactinomycin		✓
15	estradiol		
16	topotecan	✓	✓
17	cycloheximide		✓
18	docetaxel	✓	✓
19	dehydroepiandrosterone		
20	celecoxib	✓	

main memory, inferring the most strongly associated dual-predicate path connecting two concepts can be accomplished in milliseconds, and determining which pharmaceutical substance is most strongly associated across a multitude of such paths can be accomplished within microseconds. Strongly ranked predictions are often connected across more than 10,000 unique reasoning pathways (such as *docetaxel INHIBITS prostate_specific_antigen ASSOCIATED_WITH hormone-refractory_prostate_cancer*). However, PSI is able to evaluate the extent through which two concepts are related across large numbers of pathways simultaneously, by converting the task of exploring multiple possible reasoning pathways into the task of measuring the similarity between vector representations. Furthermore, unlike a discrete system in which the time required to explore such pathways increases exponentially with pathway length, the time required for search in PSI is identical for single-, double- and triple-predicate search once the relevant vectors have been constructed.

6.2 Negation in PSI Models

Negation can be carried out in PSI models using orthogonal projection in just the same way as in semantic vector models built from free text (see Section 2.1). The basic principle is that for any vector x, "not x" is modelled as the subspace orthogonal to x, and so the operation of negating x is performed by projection onto this orthogonal subspace. In the particular case of a desired concept x and and unwanted concept y, the compound concept x NOT y is represented using Equation 1.

The example in this section (and below) uses a simple database of national facts extracted from Wikipedia, which is included in the Semantic Vectors package, and lists the currency, capital city, and national animal of 277 world countries.[5] The example query we consider is "Find countries other than the USA that use the US Dollar." Since the semantic vector $S(\text{united states dollar})$ is learned by summing vectors of the form $E(\text{HAS\_CURRENCY}) \otimes E(\text{country})$, and binding distributes over addition (in the sense that $a \otimes (b + c) = (a \otimes b) + (a \otimes c)$ for all vectors a, b, c), we can decode this relationship using the release operator, so that $S(\text{united states dollar}) \oslash E(\text{HAS\_CURRENCY})$ gives a vector which, in the PSI model, could be interpreted as a prototypical vector for representing the concept "countries that use the US dollar". Using this vector as a query and searching for nearest neighbors correctly recovers the list of countries that use the US dollar as currency. Projecting this query vector so that it is orthogonal to $E(\text{united states})$ and repeating the search gives results without the United States, and with other similarity scores slightly readjusted according to how similar or different the other countries are to the United States if only the other attributes are considered. Results are presented in Table 6.

The exact scores depend on the vector type, the choice of VSA operations, and the number of dimensions used. For comparison, the next highest-ranked result that is not a country using the US dollar is included. Note that while all the result sets correctly rank countries that use the US dollar over all other results, the complex and binary result scores make a much clearer categorical distinction than the results obtained using real vectors.

The key points to note include:

- PSI correctly finds the intended set of results.
- Projecting orthogonally to the USA vector removes this result without removing others unnecessarily.

[5] This includes several regions such as England and the Isle of Man that are countries in an informal sense but not sovereign states in an official sense, hence the large number of countries.

Table 6: PSI results for countries that use the US dollar as currency. Columns marked "Any" are results close to the query vector $S(\text{united states dollar}) \oslash E(\text{HAS\_CURRENCY})$. Columns marked "∼USA" are results close to the projection of this vector onto the subspace orthogonal to $E(\text{united states})$. For each experiment, the next highest result is included below for comparison.

Country	Real, dim 1000		Complex, dim 500		Binary, dim 4096	
	Any	∼USA	Any	∼USA	Any	∼USA
bonaire	0.228	0.229	0.337	0.308	0.261	0.212
british indian ocean terr.	0.203	0.207	0.324	0.295	0.274	0.218
east timor	0.257	0.255	0.445	0.392	0.268	0.191
ecuador	0.224	0.243	0.290	0.248	0.266	0.219
saba	0.211	0.217	0.286	0.267	0.287	0.201
marshall islands	0.207	0.210	0.300	0.302	0.289	0.230
sint eustatius	0.239	0.248	0.247	0.228	0.270	0.229
turks and caicos islands	0.256	0.259	0.372	0.290	0.271	0.204
united states	0.233	**0**	0.492	**0**	0.265	**0**
...						
hungary	0.163	0.169	...	...	...	...
freetown	...	...	0.087	...	...	...
peregrine falcon	...	...	...	0.098	...	...
gambian dalasi	...	...	...	...	0.051	...
guernsey	...	...	...	...	...	0.051

- If dimensions are reduced, eventually the recovery becomes noisy and polluted with other results. Experiments like this can be used to tune models by choosing appropriate numbers of dimensions.

This example demonstrates that the orthogonal negation operator used with free text models works in this particular PSI model. Several topics remain to be investigated, such as the behavior of orthogonal negation in PSI models built using recursive hierarchies, the relationship between negation and normalization, and the use of negation in more complex statements and chains of reasoning.

6.3 Semantic Types

A long-time criticism of distributional models is that they do not have a type-system or taxonomic structure. For example, the words related to 'water' in Table 2 tend to be objects and actions related to water such as 'clothes' and 'wash', rather than other physical substances, and there is nothing that tells us explicitly that water is a physical substance.

In PSI models, each predication xRy that goes into the encoding of a concept x tells us that "x has some attribute R". In many situations, the list of attributes of a concept can be used to assign a category or type for that con-

cept (a process pioneered again by Aristotle, c.f. *Posterior Analytics*, Bk I Ch. 4, though the notion that this defines "Aristotle's theory of concepts" is a simplification). The type information for a concept x can therefore in some cases be recovered by examining the possible relations of its semantic vector, and seeing which of these relations leads clearly to another vector, rather than an empty region of semantic space. Thus for each possible relation R, we check to see if there is an elemental vector $E(y)$ which is close to the vector produced by releasing the elemental relation vector $E(R)$ from the semantic vector $S(x)$, so that $S(x) \oslash E(R) \approx E(y)$. If so, this indicates that the relation xRy was used in the semantic indexing for the vector $S(x)$, so the concept x has some relation R. Then the collection of populated relations is compared with a list of attributes appropriate for each type. An example result is as follows:[6]

```
0.478    lesotho : HAS_NATIONAL_ANIMAL : black_rhinoceros
0.557    lesotho : CAPITAL_OF-inv : maseru
0.535    lesotho : HAS_CURRENCY : lesotho_loti
'lesotho' is therefore a 'COUNTRY'
```

This numerical example could be described informally in words as "*Lesotho* has a capital city, a currency, and a national animal: it sounds as if *Lesotho* is a country."

Using a list of known attributes to infer a type in this way is a standard technique, introduced formally at least as early as Aristotle (*Posterior Analytics*, Bk I, Ch 4), and often known to computer scientists as "duck typing" (swims like a duck, quacks like a duck, so it's a duck). This so-called Aristotelian approach is sometimes criticized because it is too brittle and does not account for graded or probabilistic classification. The most famous critic was Aristotle himself, who is strict in applying the technique to triangles in mathematics, but often states that biological properties such as "cows have four stomachs" are true "for the most part". More recently, psychological experiments suggest that belonging to a category is perceived in a graded fashion (Aitchison, 2002, Ch 5), and Hofstadter and Sander (2013) propose that such fine gradations in concepts are at the core of our ability to reason by analogy.

The type-recovery process for PSI is similarly 'graded', in that the score for each relation is continuous, and the overall category-belonging score can easily be made continuous as a result. This is of course standard in the construction of any probabilistic classifier which works by combining indicative features. The

[6]This example uses the example National Facts dataset (available with the SemanticVectors package), real vectors of dimension 200, and a score-cutoff value of 0.25 so that the relation $a \approx b$ is defined by the inequality $a \cdot b > 0.25$.

interesting thing about type-recovery in PSI particularly is that all the features are simultaneously encoded in a single semantic vector, and recovered using VSA operations.

There are other ways to approach this challenge. One would be to build classifiers for entire regions of semantic space, in much the way that "concepts" are modelled as convex regions in the cognitive theory of Conceptual Spaces (Gärdenfors, 2000). Another would be to infer a type $T(x)$ deterministically from the relations listed in the initial triple store, and to add the relation $S(x) \mathrel{+}= E(\mathrm{IS\_A}) \otimes E(T(x))$ during the indexing phase. (That is, each type T would be assigned an elemental vector $E(T)$, and then a binding of the $\mathrm{IS\_A}$ relationship with this type would be added into the semantic vector $S(x)$ and recovered as robustly as other contributing relationships.) Another would be to model the category T as a sum of the predicates involved in T in the PSI space itself. This last would be the most innovative approach, and introduces the more general question of which metadata *about* the PSI model can be modelled *within* the PSI model.

6.4 Representing Orthography

In most VSA applications, the binding operator is applied to near orthogonal elemental vectors in order to generate a third vector that is dissimilar from either of its component vectors. From a geometric perspective, the vector product $E(\mathrm{A}) \otimes E(\mathrm{B})$ is likely to lead to a point in space that is far from the bound product of any other pair of elemental vectors, or any other individual elemental vector in the space. So these vector products have the same desirable property of robustness that characterizes elemental vectors, which have deliberately been constructed to be orthogonal, or close-to-orthogonal in space. While this is useful for many applications, it is also the case for all the well-behaved binding operators that, given a vector $E(\hat{\mathrm{B}})$ that is similar to $E(\mathrm{B})$, the bound product $E(\mathrm{A}) \otimes E(\hat{\mathrm{B}})$ will lead to a point in space close to $E(\mathrm{A}) \otimes E(\mathrm{B})$. (See Table 3.)

In recent work (Cohen et al, 2012d), we have exploited this property in order to estimate *orthographic similarity*, the similarity between the surface features of a word. Previous attempts to model orthographic similarity in vector space depended upon using a binding operator to generate near-orthogonal vector representations of sequences of characters within a word, including gapped sequences to allow for flexibility (Cox et al, 2011; Kachergis et al, 2011; Hannagan et al, 2011). However, encoding in this way requires the generation of a large number of vector products. As an alternative, we used interpolation between a pair of elemental vectors to generate a set of *demarcator vectors*

a predetermined distance apart from one another in high-dimensional space. A demarcator vector, $D(\text{position})$, is used to encode the position of a character within a word. So, the orthographic vector for the term "bard" is constructed as follows:

$$O(\text{bard}) = E(\text{b}) \otimes D(1) + E(\text{a}) \otimes D(2) + E(\text{r}) \otimes D(3) + E(\text{d}) \otimes D(4).$$

The similarity between $E(\text{a}) \otimes D(2)$ and $E(\text{a}) \otimes D(1)$ is simply the similarity between $D(2)$ and $D(1)$). Demarcator vectors are constructed such that $D(1) \cdot D(2) > D(1) \cdot D(3)$, so the similarity between a pair of orthographic vectors reflects the distance between the position of the characters they have in common. The method for doing this is simple. First, orthogonal endpoint vectors are generated for the start and the end position in the word. Then the other demarcator vectors are generated by linear interpolation between these endpoints.

Table 7 provides examples of nearest-neighbor search based on orthographic similarity in a 32,000 dimensional binary vector space derived from the widely-used Touchstone Applied Science Associates, Inc. (TASA) corpus (only terms with at least two characters that occurred between 5 and 15,000 times in the corpus were considered, and terms containing non-alphabet characters were excluded). In each case, the cue consists of a misspelled word, and the nearest neighboring terms in the corpus to this misspelling are retrieved.

As illustrated in the table, this simple encoding results in a vector representation of terms that preserves similarity in the face of insertion, deletion, substitution and change in the position of characters. Such representations are of interest from a psychological perspective, as humans are also able to recognize terms despite changes of this sort.

6.5 Tabular Data and Continuous Quantities

In this example, we show that the ideas used for indexing predications and encoding orthography can also be applied to modelling tabular data. The prin-

Table 7: Orthographic Similarity in a 32,000 dimensional binary vector space. Each query term in **bold** is a misspelling / out-of-vocabulary word. The results are the orthographically closest vocabulary words in the TASA corpus.

accomodated		**carribean**		**glamourous**		**assasination**	
0.394	accommodated	0.374	caribbean	0.350	glamorous	0.471	assassination
0.385	accommodate	0.335	barbarian	0.306	humorous	0.458	assassinations
0.367	accommodates	0.331	carrier	0.300	glomerulus	0.403	assassinated
0.363	accommodation	0.331	carriages	0.298	allosaurus	0.383	assistant
0.346	accommodating	0.331	carbine	0.292	homologous	0.382	assimilation

Table 8: Nearest neighbors in a model built from tabular data, with each distinct value treated as a random elemental vector. Real vectors, dimension 200.

J. Adams		T. Roosevelt	
J. Adams	1.000	T. Roosevelt	1.000
Jefferson	0.257	Coolidge	0.322
Kennedy	0.255	B. Harrison	0.217
Ford	0.219	Eisenhower	0.209
Hoover	0.195	Garfield	0.197
Taft	0.191	Hayes	0.181

cipal is similarly simple: any column-value pair in a table can be modelled as a bound product of the vector representing the column with the vector representing the value, and then these vectors can be superposed to give a combined vector for each row. (This is equivalent to transforming a table into a set of triples and performing PSI on the resulting triples.)

Results can, however, be initially disappointing. As an example, we used a test dataset listing the Presidents of the USA (available with the Semantic Vectors package, columns including name, party, state of birth, religion, years of birth, death, taking and leaving office, and age at these times). Using random elemental vectors for the data values, the combined vectors for the rows tend to share features only if they have an exactly equal value in at least one column. Example results for the queries *J. Adams* and *T. Roosevelt* are shown in Table 8. The nearest neighbors tend to come from exact matches: for example, John Adams and Thomas Jefferson died in the same year (1826), while Theodore Roosevelt and Calvin Coolidge shared a party (Republican) and an age of death (60). With such a small dataset, there is also a lot of variation between different experimental runs when using random elemental vectors.

Results are improved by using orthographic vectors for the values, as described in the previous section (Table 9). The orthographic similarity between names has, for example, raised the similarity of the other Adams and Roosevelt presidents as nearest neighbors.

However, orthographic similarity remains a poor way of comparing dates or numbers. There is some similarity between (say) 1800 and 1801, but much less similarity between (say) 1799 and 1800. This challenge can be addressed using the same technique as used for generating demarcator vectors for orthographic encoding. That is, for each column that is recognized as a numerical quantity, orthogonal endpoint vectors are created for the minimum and maximum numbers in the range of values in this column. Vectors for intermediate

Table 9: Nearest neighbors with values treated as orthographic vectors. Real vectors, dimension 200.

J. Adams		T. Roosevelt	
J. Adams	1.000	T. Roosevelt	1.000
Ford	0.808	Coolidge	0.830
Buchanan	0.794	F. D. Roosevelt	0.812
J. Q. Adams	0.788	B. Harrison	0.807
Garfield	0.788	Kennedy	0.796
Van Buren	0.788	Carter	0.791

Table 10: Nearest neighbors with dates and ages treated as numerical quantity vectors, other string values treated as orthographic vectors. Real vectors, dimension 200.

J. Adams		T. Roosevelt	
J. Adams	1.000	T. Roosevelt	1.000
Jefferson	0.994	Coolidge	0.969
Madison	0.992	Grant	0.964
Jackson	0.982	Taft	0.957
Monroe	0.980	F. D. Roosevelt	0.956
J. Q. Adams	0.979	Arthur	0.953

values are then generated by weighted interpolation between these endpoints. Results using this method are shown in Table 10. Note that several spurious results have disappeared, and historically closer presidents are now preferred.

This technique of generating vectors to represent numeric quantities can also be used to create queries for particular columns. For example, we can now search for items whose year of taking office or whose year of birth are close to a particular value, by generating the vector $E(\mathrm{column}) \otimes D(\mathrm{year})$, where D again refers to a demarcator vector. Note the way the column is important, because it gives both the property to be searched for, and the appropriate endpoints. Results for year of taking office near to 1800 and year of birth near to 1900 are given in Table 11. The method using raw elemental vectors is more or less random, whereas the use of numeric vectors gives results that are all in the right periods. (Results vary considerably between real, complex, and binary vectors, the reasons for which are at present poorly understood.)

This case-study demonstrates the following points:

- Tabular data can be represented in a distributional model.
- Quantitative numeric attributes can be treated appropriately and recov-

Table 11: Nearest neighbors for date-specific searches. Binary vectors, dimension 2048.

Orthographic vectors for years				Numeric vectors for years			
Took office 1800		Born 1900		Took office 1800		Born 1900	
Cleveland	0.118	Fillmore	0.123	Madison	0.238	Reagan	0.751
Obama	0.116	Ford	0.119	Washington	0.223	G.H.W. Bush	0.711
G.H.W. Bush	0.106	Hoover	0.117	Monroe	0.211	Ford	0.703
Pierce	0.104	Van Buren	0.114	Jefferson	0.204	Nixon	0.687
Garfield	0.104	Tyler	0.111	J. Adams	0.195	Eisenhower	0.680
Wilson	0.104	G. W. Bush	0.109	Jackson	0.194	Truman	0.675

ered easily (this overcomes something that has been considered a major theoretical shortcoming of distributional semantic models).

- This can be done by reusing the VSA operations and demarcator vector techniques introduced already: no special new mathematical operators need to be used.

- The representation stays holistic throughout: we do not have to attach any special semantics to particular dimensions.

Of course, the set-theoretic approach to representing structured tabular data, and its instantiation in relational databases, is well-established and highly successful: the goal of this new work is not to supplant such technologies, but to explore alternatives that may support complementary and more contextual features.

7 Previous and Related Work

Using spatial models for reasoning tasks is not a new idea: the explicit correspondence between geometric inclusion and logical implication goes back at least as far as Aristotle's introduction to logic itself, with the definition:

> That one term should be included in another as in a whole is the same as for the other to be predicated of all of the first. (*Prior Analytics*, Book I, Ch. 1).

Recent years have seen a significant growth of new mathematical applications and empirical successes in this area, of which Predication-based Semantic Analysis is just one.

The introduction of tensor products to conceptual modelling in artificial intelligence is often ascribed to Smolensky (1990). The tensor product concept

can be extended mathematically to products of any number of constituents, the mathematical links with quantum models for cognition being perhaps most thoroughly explored in the physics literature in the works of Aerts et al (Aerts and Czachor, 2003; Aerts, 2009).

Within the cognitive science community, vector symbolic approaches to reasoning have been developed and evaluated for their ability to perform tasks traditionally accomplished by discrete symbolic models, often motivated by Fodor and Pylyshn's critique concerning the limited ability of existing connectionist models of cognition to address such issues (Fodor and Pylyshyn, 1988). One prominent area of application concerns analogical reasoning (Plate, 1994, 2000; Kanerva et al, 2001; Eliasmith and Thagard, 2001). This work demonstrates that analogical mapping can be accomplished by a VSA trained on a small set of propositions that have been deliberately constructed to represent a well-defined analogical reasoning problem. A further motivating argument for this work has to do with the relative biological plausibility of distributed representations, that do not require a one-to-one mapping between their representational units and units of information such as discrete symbols. Further support for this argument is provided by recent work by Crawford and his colleagues, who demonstrate that a VSA that encodes and retrieves propositions can be implemented using a network of simulated neurones (Crawford et al, 2013). This work involved the encoding of a fairly large number of propositions, and is similar in this respect to our work with PSI, though the goals of these endeavors are markedly different. Some of this work explores the biological plausibility of such networks as "simulated brains" (see e.g., Eliasmith (2013)), and investigations in this area have become mature enough to progress beyond purely informatics questions (such as "Does the system produce good outputs?") to physical questions (such as "Would such a system in a real brain consume a reasonable amount of energy?").

In computational linguistics, some of the most innovative recent works have been in semantic composition, to the extent that 'compositional distributional semantics' has become a recognized area of study in its own right. This work was initially motivated by several long-time problems with classical (in the sense of Boolean or set-theoretic) models for compositionality, such as the natural observations that a *tiger moth* is not a tiger and a *stone lion* is not a lion (Gärdenfors, 2000; Widdows, 2008). Partly to address the fact that adjectives change their meaning depending on the noun being modified, Baroni and Zamparelli (2010) developed the representation of nouns as vectors and adjectives as matrices (equivalent to rank-2 tensors) that operate on nouns. This idea is applied to more general syntactic binding operations by Socher et al (2012),

who use a recursive neural network to learn an appropriate vector and matrix to every node in a parse tree. Grefenstette and Sadrzadeh (2011) also use matrices for composition of vectors, based on a general model for composition of meaning using category theory, and Grefenstette (2013) has generalized this approach to describe a full-blown predicate calculus using tensors. One feature of several of these works is the continued use of matrices and tensors: so instead of projecting the n^2 representation of a product-state back into n dimensions using an operator such as convolution, some researchers rely on numerical optimizations such sparse-matrix representations to maintain tractability at larger scales. For more of the mathematical and linguistic basis for some of these theories, see Coecke et al (2010) and Clark and Pulman (2007): for further experiments on composition, explorations of the relationship with quantum structures, see Blacoe and Lapata (2012) and Blacoe et al (2013).

Vector representations still have many mathematical properties that are well-known in other applications whose use in computational linguistics and artificial intelligence is in its early stages. Examples of such developments include the use of dual-spaces by Turney (2012) and lattices by Widdows (2004, Ch. 8), and Clarke (2012). In general, the mathematical sophistication and experimental successes of compositional distributional semantics have been making considerable strides over the past decade, sometimes hand-in-hand and sometimes independently, and we expect developments in this field to keep accelerating over the next few years.

7.1 Summary of Previous Results using PSI

In this section we provide a brief summary of results achieved using PSI over the past five years. The bullet points below provide key results only, and we encourage the interested reader to refer to the source material for a more detailed description of these and other related findings. The list is provided roughly in the sequence in which these results were obtained. As such it generally demonstrates a progression from the foundational to the applied. Earlier work established with the capacity of PSI to retrieve encoded information, explored the application of quantum logical operators in PSI space, and established a basis for large-scale analogical reasoning experiments. Recent work has applied these capabilities predict helpful and harmful effects of drugs.

- Predication-based Semantic Indexing was first implemented using real vectors and with permutation of coordinates used to encode predicate type. It was trained on a set of over ten million semantic predications extracted by SemRep, and tested by comparing the retrieved predica-

tions for 1000 concepts with those in the original set. When evaluated for its ability to recover these predications, the PSI system achieved best results of 99.7% precision and 65.8% recall at 1500 dimensions (Cohen et al, 2009).

- This work was extended to chains of reasoning over several predications, recovering 2-step conclusions from 1000 pairs of linked predications as premises, with 95.9% precision (Cohen et al, 2010c). Orthogonal negation was also used to obtain new middle-terms in such pathways, resulting in new reasoning pathways being found in 94.1% of cases, maintaining a precision of 92.3%.

- Along with other distributional models, the initial implementation of PSI was embedded in a system called EpiphaNet, providing biomedical scientists with the means to explore the relations between concepts for research purposes Cohen et al (2010b); Schvaneveldt et al (2013). Qualitative evaluation of the use of the system by biomedical domain experts revealed frequent discovery of surprising yet informative associations between concepts within their domain of expertise, as one might anticipate given the breadth of literature upon which the models concerned were trained.

- PSI for analogical reasoning with binary vectors was implemented following an example from Kanerva (2010). This permitted analogical retrieval (of the form *A is to B as C is to ?*), with precision over 60% with an individual cue, and in the high 90%'s if many cues are superposed (Cohen et al, 2011). This supports the claim that analogy works best with 'entangled' concepts, where entangled is used to mean 'a compound object that cannot be factored into a single pair of elements', as in quantum mechanics.

- Reasoning paths of using a variety of different relation types can be inferred by analogy from known examples, and combined to recover held-out sets of therapeutic relations (Cohen et al, 2012a,b,c). For both complex and binary vectors, this work also demonstrated gains when using quantum disjunction (subspace) over superposition for combining several near-orthogonal reasoning pathways. In addition, superposing vectors to extend the search over longer (triple-predicate) pathways further improved performance.

- Similarly, reasoning pathways indicating adverse effects of drugs can be inferred from known examples and used to recover held-out side-effects.

Predications populating these reasoning pathways can be linked back to the literature from which they were extracted, revealing plausible physiological explanations for these effects (Shang et al, 2014).

- Most recently, therapeutic substances that may inhibit prostate cancer growth have been predicted. Laboratory experiments have shown that one-third of positive results are in the top 10% of the ranked predictions (Cohen et al, 2014).

8 Conclusion

What we have discovered in recent years is that continuous models can address some of the gaps between formal deductive reasoning and human-like intelligent inference. This is primarily because continuous models can support the notion of 'near enough' in such a way that we can *analyse several reasoning pathways simultaneously* and generate likely hypotheses of relationships between concepts: not in the way humans do when we write a mathematical proof; but more akin to the way humans, including domain-experts, have hypotheses or hunches that may progress from vague associations to confirmed pieces of knowledge when put to the test.

Predication-based Semantic Indexing or PSI is one of the successful models that has been developed in this area. PSI represents the concepts and relationships from a knowledge-base in such a way that new relationships can be inferred, not just by following individual chains of predications, but also by examining many reinforcing pieces of information simultaneously, using superposition and quantum disjunction. PSI has so far been applied particularly to biomedical informatics challenges such as predicting TREATS relationships for drug discovery and drug repurposing: however, the mathematical models are more general than these particular examples, and several other researchers have applied related techniques to other concept-representation, composition, and inference challenges, with increasing and notable successes. As well as achieving good semantic performance, the computational performance of these methods is typically very favorable when compared with traditional logic programming.

Some well-known challenges remain, including representing and inferring the semantic type of an object (such as *DISEASE* or *COUNTRY*), and (somewhat surprisingly) using continuous models to represent continuous as well as discrete concepts. As demonstrated in this paper, these topics can be addressed and these problems can be solved in vector models. We expect that

at least some of these initial successes will take root firmly in computational linguistics and informatics more generally. For many researchers and practitioners, the accustomed gulf between symbolic and distributional, rule-based and statistical, rationalist and empiricist methods is quickly disappearing, to be replaced by the much more interesting challenge of making these methods work together to produce systems that can apply human-like sophistication to bodies of information far beyond the individual human scale.

9 Funding

This research was supported in part by US National Library of Medicine [grant numbers R21 LM010826, R01 LM011563], and Cancer Prevention and Research Institute of Texas [grant number RP110532-AC].

10 Acknowledgments

The authors would like to acknowledge Tom Rindflesch for extracting and sharing the predication database, and the Intramural Research Program of the US National Institutes of Health, National Library of Medicine for supporting his work in this area.

The authors would like to thank the reviewers who have contributed considerably to making the paper more exact and more readable. We also thank Eric Veach for particular discussions and guidance in analysing the distribution of angles over high-dimensional spheres.

References

Aerts D (2009) Quantum structure in cognition. Journal of Mathematical Psychology 53(5):314–348

Aerts D, Czachor M (2003) Quantum aspects of semantic analysis and symbolic artificial intelligence. arXiv preprint quant-ph/0309022

Aitchison J (2002) Words in the Mind: An Introduction to the Mental Lexicon, 3rd edn. Blackwell

Baeza-Yates R, Ribiero-Neto B (1999) Modern Information Retrieval. Addison Wesley / ACM Press

Baroni M, Zamparelli R (2010) Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In: Proceedings of

the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)

Berners-Lee T, Hendler J, Lassila O (2001) The Semantic Web. Scientific American 284(5):28–37

Birkhoff G, von Neumann J (1936) The logic of quantum mechanics. Annals of Mathematics 37:823–843

Blacoe W, Lapata M (2012) A comparison of vector-based representations for semantic composition. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics, EMNLP-CoNLL '12, pp 546–556

Blacoe W, Kashefi E, Lapata M (2013) A quantum-theoretic approach to distributional semantics. In: Proceedings of NAACL-HLT, pp 847–857

Blei DM (2012) Probabilistic topic models. Communications of the ACM 55(4):77–84

Busemeyer JR, Bruza PD (2012) Quantum models of cognition and decision. Cambridge University Press

Clark S, Pulman S (2007) Combining symbolic and distributional models of meaning. In: AAAI Spring Symposium: Quantum Interaction, pp 52–55

Clarke D (2012) A context-theoretic framework for compositionality in distributional semantics. Computational Linguistics 38(1):41–71

Coecke B, Sadrzadeh M, Clark S (2010) Mathematical foundations for a compositional distributional model of meaning. CoRR abs/1003.4394

Cohen T, Schvaneveldt R, Rindflesch T (2009) Predication-based semantic indexing: Permutations as a means to encode predications in semantic space. AMIA Annu Symp Proc pp 114–8

Cohen T, Schvaneveldt R, Widdows D (2010a) Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections. Journal of Biomedical Informatics 43(2):240–256

Cohen T, Whitfield GK, Schvaneveldt RW, Mukund K, Rindflesch T (2010b) EpiphaNet: An interactive tool to support biomedical discoveries. Journal of Biomedical Discovery and Collaboration 5:21–49

Cohen T, Widdows D, Schvaneveldt, R Rindflesch T (2010c) Logical leaps and quantum connectives: Forging paths through predication space. In: Proceedings of the AAAI Fall Symposium on Quantum Informatics for Cognitive, Social, and Semantic Processes (QI 2010)

Cohen T, Widdows D, Schvaneveldt R, Rindflesch T (2011) Finding schizophrenia's prozac: Emergent relational similarity in predication space. QI'11 Proceedings of the 5th International Symposium on Quantum Interactions Aberdeen, Scotland Springer-Verlag Berlin, Heidelberg

Cohen T, Widdows D, De Vine L, Schvaneveldt R, Rindflesch TC (2012a) Many Paths Lead to Discovery: Analogical Retrieval of Cancer Therapies. In: Busemeyer JR, Dubois F, Lambert-Mogiliansky A, Melucci M (eds) Quantum Interaction, no. 7620 in Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp 90–101

Cohen T, Widdows D, Schvaneveldt R, Rindflesch T (2012b) Discovery at a distance: Farther journey's in predication space. In: Proceedings of the First International Workshop on the role of Semantic Web in Literature-Based Discovery (SWLBD2012), Philadelphia, PA

Cohen T, Widdows D, Schvaneveldt RW, Davies P, Rindflesch TC (2012c) Discovering discovery patterns with predication-based semantic indexing. Journal of biomedical informatics 45(6):1049–1065

Cohen T, Widdows D, Wahle M, Schvaneveldt R (2012d) Orthogonality and orthography: Introducing measured distance into semantic space. Proceedings of the Seventh International Conference on Quantum Interaction, Leicester, UK, 2013

Cohen T, Widdows D, Stephan C, Zinner R, Kim J, Rindflesch T, Davies P (2014) Predicting high-throughput screening results with scalable literature-based discovery methods. CPT: Pharmacometrics and Systems Pharmacology (in press)

Cox GE, Kachergis G, Recchia G, Jones MN (2011) Toward a scalable holographic word-form representation. Behavior Research Methods 43(3):602–615

Crawford E, Gingerich M, Eliasmith C (2013) Biologically plausible, human-scale knowledge representation. In: Proceedings of the 35th annual conference of the cognitive science society

De Vine L, Bruza P (2010) Semantic oscillations: Encoding context and structure in complex valued holographic vectors. Quantum Informatics for Cognitive, Social, and Semantic Processes (QI 2010)

Deerwester S, Dumais S, Furnas G, Landauer T, Harshman R (1990) Indexing by latent semantic analysis. Journal of the American Society for Information Science 41(6):391–407

Eliasmith C (2013) How to Build a Brian: A Neural Architecture for Biological Cognition. Oxford University Press

Eliasmith C, Thagard P (2001) Integrating structure and meaning: A distributed model of analogical mapping. Cognitive Science 25(2):245–286

Fodor JA, Pylyshyn ZW (1988) Connectionism and cognitive architecture: A critical analysis. Cognition 28(1–2):3–71

Gallant SI, Okaywe TW (2013) Representing objects, relations, and sequences. Neural computation 25(8):2038–2078

Gärdenfors P (2000) Conceptual Spaces: The Geometry of Thought. Bradford Books MIT Press

Gayler RW (2004) Vector symbolic architectures answer jackendoff's challenges for cognitive neuroscience. In: In Peter Slezak (Ed.), ICCS/ASCS International Conference on Cognitive Science, Sydney, Australia. University of New South Wales., pp 133–138

Grassmann H (1862) Extension Theory. History of Mathematics Sources, American Mathematical Society, London Mathematical Society, translated by Lloyd C. Kannenberg (2000)

Grefenstette E (2013) Towards a formal distributional semantics: Simulating logical calculi with tensors. arXiv preprint arXiv:13045823

Grefenstette E, Sadrzadeh M (2011) Experimental support for a categorical compositional distributional model of meaning. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)

Hannagan T, Dupoux E, Christophe A (2011) Holographic string encoding. Cognitive science 35(1):79–118

Hofstadter D, Sander E (2013) Surfaces and Essences. Basic Books

Hristovski D, Friedman C, Rindflesch TC, Peterlin B (2006) Exploiting semantic relations for literature-based discovery. AMIA Annual Symposium Proceedings / AMIA Symposium AMIA Symposium pp 349–53

Hume D (1739) A treatise of human nature. Courier Dover Publications, 2003

Jones MN, Mewhort DJK (2007) Representing word meaning and order information in a composite holographic lexicon. Psychological Review 114:1–37

Kachergis G, Cox G, Jones M (2011) OrBEAGLE: integrating orthography into a holographic model of the lexicon. Artificial Neural Networks and Machine Learning–ICANN 2011 pp 307–314

Kanerva P (1988) Sparse distributed memory. The MIT Press

Kanerva P (2009) Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. Cognitive Computation 1(2):139–159

Kanerva P (2010) What we mean when we say "What's the dollar of mexico?": Prototypes and mapping in concept space. In: 2010 AAAI Fall Symposium Series

Kanerva P, Sjödin G, Kristoferson J, Karlsson R, Levin B, Holst A, Karlgren J, Sahlgren M (2001) Computing with large random patterns. In: Uesaka, Y., Kanerva, P., and Asoh, H. (eds.) Foundations of Real-World Intelligence., CSLI Publications., Stanford, California.

Khrennikov A (2010) Ubiquitous Quantum Structure: From Psychology to Finance. Springer

Kilicoglu H, Shin D, Fiszman M, Rosemblat G, Rindflesch TC (2012) Semmeddb: a pubmed-scale repository of biomedical semantic predications. Bioinformatics 28(23):3158–3160

Kintsch W (1998) Comprehension : a paradigm for cognition. Cambridge University Press

Kintsch W, Keenan J (1973) Reading rate and retention as a function of the number of propositions in the base structure of sentences. Cognitive Psychology 5(3):257–274, DOI 10.1016/0010-0285(73)90036-4, URL http://www.sciencedirect.com/science/article/pii/0010028573900364

Landauer T, Dumais S (1997) A solution to Plato's problem: The latent semantic analysis theory of acquisition. Psychological Review 104(2):211–240

Levy SD, Gayler R (2008) Vector symbolic architectures: A new building material for artificial general intelligence. In: Proceedings of the 2008 conference on Artificial General Intelligence 2008: Proceedings of the First AGI Conference, IOS Press, pp 414–418

Manning CD, Schütze H (1999) Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge, Massachusetts

von Neumann J (1932) Mathematical Foundations of Quantum Mechanics. Princeton Univ Press, reprinted 1996

Plate TA (1994) Estimating analogical similarity by dot-products of holographic reduced representations. Advances in neural information processing systems pp 1109–1109

Plate TA (2000) Analogy retrieval and processing with distributed vector representations. Expert systems 17(1):29–40

Plate TA (2003) Holographic Reduced Representations: Distributed Representation for Cognitive Structures. CSLI Publications

van Rijsbergen K (2004) The Geometry of Information Retrieval. Cambridge University Press

Rindflesch TC, Fiszman M (2003) The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. Journal of Biomedical Informatics 36:462–477

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

Sahlgren M, Holst A, Kanerva P (2008) Permutations as a means to encode order in word space. Proceedings of the 30th Annual Meeting of the Cognitive Science Society (CogSci'08), July 23–26, Washington DC, USA

Salton G, McGill M (1983) Introduction to modern information retrieval. McGraw-Hill, New York, NY

Sandin F, Emruli B, Sahlgren M (2011) Incremental dimension reduction of tensors with random index. CoRR abs/1103.3585

Schvaneveldt R, Cohen T, Whitfield GK (2013) Paths to discovery. Expertise and Skills Acquisition: The Impact of William G Chase (Carnegie Mellon Symposia on Cognition Series) James J Staszewski (ed) pp 147–177

Shang N, Xu H, Rindflesch TC, Cohen T (2014) Identifying plausible adverse drug reactions using knowledge extracted from the literature. Journal of Biomedical Informatics epub ahead of print, DOI 10.1016/j.jbi.2014.07.011

Smolensky P (1990) Tensor product variable binding and the representation of symbolic structures in connectionist systems. Artificial intelligence 46(1):159–216

Socher R, Huval B, Manning CD, Ng AY (2012) Semantic compositionality through recursive matrix-vector spaces. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics, EMNLP-CoNLL '12, pp 1201–1211

Switzer P (1965) Vector images in document retrieval. Statistical association methods for mechanized documentation pp 163–171

Turney PD (2012) Domain and function: A dual-space model of semantic relations and compositions. J Artif Intell Res(JAIR) 44:533–585

Wahle M, Widdows D, Herskovic JR, Bernstam EV, Cohen T (2012) Deterministic binary vectors for efficient automated indexing of medline/pubmed abstracts. In: AMIA Annual Symposium Proceedings, American Medical Informatics Association, vol 2012, p 940

Widdows D (2003) Orthogonal negation in vector spaces for modelling word-meanings and document retrieval. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL), Sapporo, Japan

Widdows D (2004) Geometry and Meaning. CSLI Publications

Widdows D (2008) Semantic vector products: Some initial investigations. In: Proceedings of the Second International Symposium on Quantum Interaction

Widdows D, Cohen T (2010) The semantic vectors package: New algorithms and public tools for distributional semantics. In: Fourth IEEE International Conference on Semantic Computing (ICSC)

Widdows D, Cohen T (2012) Real, complex, and binary semantic vectors. In: Busemeyer J, Dubois F, Lambert-Mogiliansky A, Melucci M (eds) Sixth International Symposium on Quantum Interaction

Zadeh LA (1988) Fuzzy logic. Computer 21(4):83–93

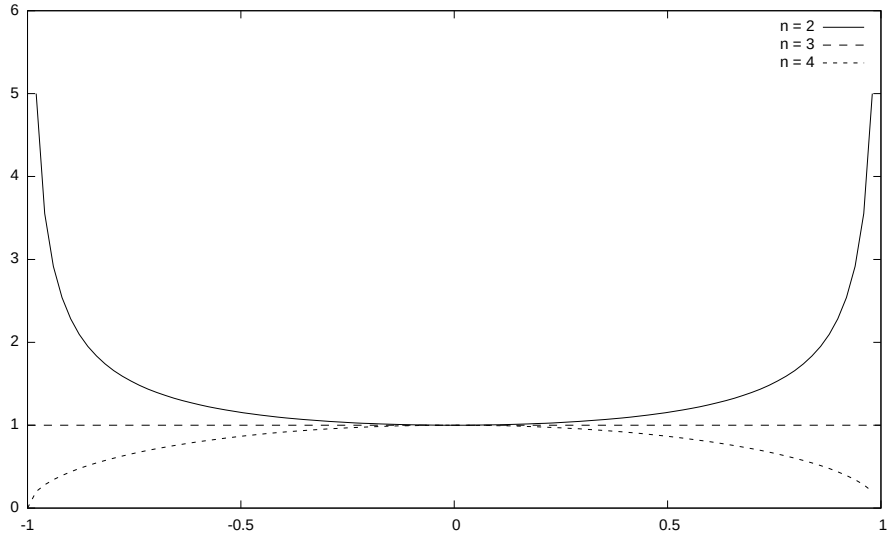

Figure 3: Distributions of cosine similarities between randomly-chosen vectors in low dimensions

A Appendix: Mathematical Explanation of Similarity Distributions in High Dimensions

This appendix explains the mathematics behind the distributions of similarities between random vectors shown in Figure 1 from Section 3, and why these are so different from the distributions that we experience intuitively in one, two, and three dimensions.

In one dimension, the unit vectors are just the set $\{-1, 1\}$ and the cosine similarities available are exactly plus and minus one, which mirrors the situation in traditional two-valued logic. In two dimensions, the unit vectors fall on the unit circle, and the cosine of the angle between two randomly-chosen vectors follows the arcsin distribution, which is concentrated towards the extremes. In three dimensions, the unit vectors fall on the standard unit sphere, and the extra concentration of cosine values 'near the poles' is exactly cancelled out by the larger size of the parallels 'near the equator', and the distribution of angles between randomly-chosen angles is uniform. In four dimensions, the distribution starts to prefer the middle-region near the equator. These distributions are shown in Figure 3, which should be contrasted with Figure 1.

The key mathematical simplification that leads to these distributions comes from noting that without loss of generality, we can use symmetry of the unit

sphere in any dimension to fix one of the vectors to $(1, 0, \ldots, 0)$, and then the cosine similarity of this vector for any other unit vector $(x_1, ..., x_n)$ is simply x_1. It follows that the distribution of cosine similarities of *pairs* of unit vectors is the same as the distribution of any chosen coordinate x_i in the unit sphere $S^{n-1} = \{(x_1, ..., x_n) \in \mathbb{R}^n | \sum_{i=1}^n x_i^2 = 1\}$.

Geometrically speaking, for any fixed $x_1 \in [-1, 1]$, the locus of points in S^{n-1} with the first-coordinate x_1 is a lower-dimensional sphere, given by the constraint $\sum_{i=2}^n x_i^2 = 1 - x_1^2$, which describes a sphere isomorphic to S^{n-2} with radius $\sqrt{1 - x_1^2}$, as shown in Figure 4. The surface area of this sphere is proportional to the $(n-2)^{th}$ power of its radius (modulo constants involving π and the total number of dimensions), so for each cosine similarity x, the size of the set of unit vectors with this similarity score is proportional to $(1 - x^2)^{\frac{n-2}{2}}$. Finally we adjust for the fact that cosine similarity values closer to ± 1 correspond to proportionally more angles, by a similar factor of $\frac{1}{\sqrt{1-x^2}}$ (the arcsin distribution which we already saw is the dominant factor in 2 dimensions). Thus the distribution of similarity scores between randomly-chosen unit vectors in $\mathbb{R}^n$ is proportional to $(1 - x^2)^{\frac{n-3}{2}}$. These curves are depicted in Figures 3 and 1.

These curves are shown non-normalized for the sake of comparison: to represent them as probability distributions, they need to be normalized so that the area under each curve is equal to 1, a general solution of which requires hypergeometric functions.

The mean and variance of the distribution in each dimension can be deduced using relatively elementary methods. Making continued use of the simplification that the distribution of similarities is the same as the distribution of the first coordinate x_1, it follows that the mean similarity is zero in each dimension, because each coordinate on the unit sphere is distributed symmetrically. The variance of the distribution is therefore given by $E[x_1^2]$, the expected value of the square of the first coordinate. Since all the x_i have the same distribution,

$$E(x_1^2) = \frac{1}{n} \sum E[x_i^2] = \frac{1}{n} E\left[\sum x_i^2\right] = \frac{1}{n},$$

since for unit vectors, $\sum x_i^2 = 1$.

Thus the distribution of cosine similarity scores between randomly-chosen unit vectors in $\mathbb{R}^n$ has a mean of 0 and a standard deviation of $\frac{1}{\sqrt{n}}$.

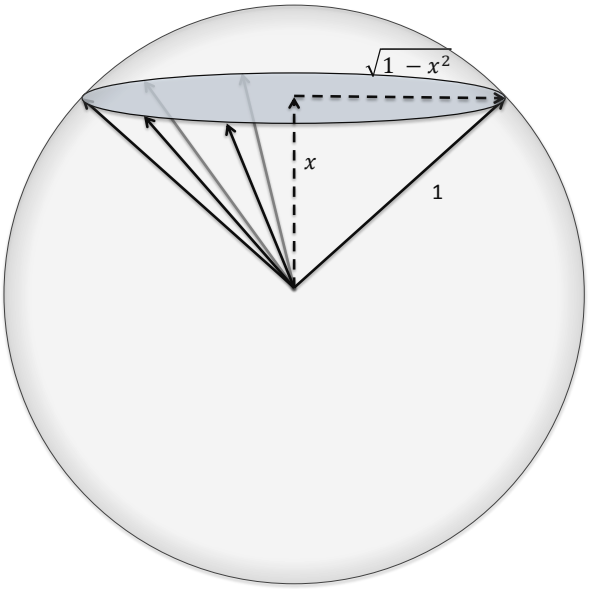

Figure 4: The set of points on the unit sphere S^2 with first coordinate equal to x is a circle (isomorphic to the sphere S^1) with radius $\sqrt{1-x^2}$.