Graded Semantic Vectors: An Approach to Representing Graded Quantities in Generalized Quantum Models

Dominic Widdows[1] and Trevor Cohen[2]

[1] Microsoft Bing
[2] University of Texas School of Biomedical Informatics at Houston

Appeared in 9th international conference on Quantum Interaction (QI-2015), Filzbach, Switzerland, July 14-17, 2015

Abstract. Semantic vector models are traditionally used to model concepts derived from discrete input such as tokenized text. This paper describes a technique to address continuous and graded quantities using such models. The method presented here grows out of earlier work on modelling orthography, or letter-by-letter word encoding, in which a graded vector is used to model character-positions within a word. We extend this idea to use a graded vector for a position along any scale. The technique is applied to modelling time-periods in an example dataset of Presidents of the United States. Initial examples demonstrate that encoding the time-periods using graded semantic vectors gives an improvement over modelling the dates in question as distinct strings. This work is significant because it fills a surprising technical gap: though vector spaces over a continuous ground-field seem a natural choice for representing graded quantities, this capability has been hitherto lacking, and is a necessary step towards a more complete vector space model of conceptualization and cognition.

1 Introduction and Outline

This paper proposes and demonstrates a general approach for encoding naturally graded quantities within semantic vector models. This is important in theory for understanding how discrete relationships (such as “a cheetah is a land animal”) and graded quantities (such as “a cheetah can reach speeds of 70 miles-per-hour”) might be combined in a single cognitive model. It’s important in practice for the engineering of rich search and navigation systems (for example, for finding books-for-sale on a particular topic within a given price-range). These motivations are discussed more thoroughly in Section 2.

The paper describes one approach to solving this problem. Section 3 introduces the method with a first example case, the challenge of modelling orthography. That is, instead of treating each word as atomic, it is modelled as a collection of characters bound to different positions. Vectors representing the positions of letters within a written word are thus a first example of graded semantic vectors. Section 4, at the heart of the paper, describes how the technique used to model orthography can be generalized to cover any number of other graded scales, and concepts with attributes that take values along these scales. This section also describes some of the mathematical properties of this model which make it quantum-like: some of these are well-known, some are relatively novel, and some are proposed for further investigation.

The rest of the paper is devoted to worked examples demonstrating the potential of the method. Section 5 explains how this can be extended to temporal concepts in modelling a table of information about Presidents of the USA. Section 6 shows how properties of particular items can be recovered, compared, and inferred from the semantic vector model. Finally, Section 7 proposes further work expanding these developments.

2 The Problem of Graded Representation

Any attempt at a complete account of cognition must sooner or later address the challenge of comparing graded quantities. Such notions include larger and smaller, higher and lower, before and after, nearer and farther, faster and slower, and so on. It is not enough to classify items into discrete buckets such as 'small, medium and large', or 'ancient, medieval, and modern', because while these may work for some original purpose, situations inevitably arise wherein a coarser classification must become more fine-grained to serve some new situation.

Our typical approach to this challenge is to introduce numerical scales with some agreed units, such as seconds for time, meters for distance, degrees for temperature, currency units for money, and so on. Given suitable measuring equipment, real world situations can be described using these numerical scales and compared appropriately: for example, the question of whether the temperature in a room is above or below a given target temperature can be transformed to a question of comparing numbers. This comparison can be carried out in a computing machine, for example by comparing the corresponding digits in binary representations of the measured ambient temperature and the desired target temperature. Such a system would typically not be thought of as intelligent: it can compare graded quantities that are usefully related to the concepts *too hot*, *too cold*, and the desired state of *just right*, but in a closed way that is quite removed from the way humans communicate about these concepts and that does not learn or generalize without external help.

On the other hand, recent decades have established firmly systems that do learn and generalize directly from the way humans communicate, but not in a way that supports the representation of graded quantities. Here we are referring to semantic vector models, a family of models that build representations from large corpora of natural language [1]. Models of this sort have usually been constructed from discrete data, such as tokenized text: canonical early examples of this methodology include the vector space model for search [2] and Latent Semantic Analysis [3]. The connection with generalized quantum models arises from the key observation that the Hilbert spaces and projections used quantum mechanics are from the same family as the vector spaces and similarity measurements used in semantic vector models [4, 5], albeit with some customary differences such as the use of complex coordinates in quantum mechanics and real coordinates in natural language applications [6].

In natural language processing, these methods can be used to learn quite successfully that (for example) *hot* and *cold* are related to one another, possibly even that they are antonyms, and that they are also related to *temperature*. But this is typically done by mapping *hot* and *cold* to individual points in a vector space, without any notion that there is a scale of physical temperature values, that *hot* and *cold* describe regions in this value space (subject to context and vagueness), and that other terms such as *warm*, *cool*, *frigid*, etc. also describe regions on this scale.

Many corpus-based techniques deliberately neglect this challenge, for example, by replacing all strings of digits with a single token such as NUMBER in preprocessing (see e.g., [7], though the practice is widespread and standard). Obviously this throws away a lot of information with intended meaning. This raises the challenge of combining such text-based similarity models with information from graded and continuous observations. Given that the vector space models used in distributional semantics are naturally continuous, combining such modalities should in theory not be too difficult.

In the rest of this paper we demonstrate one such combined model. The example applications include string similarity, exploring tabular data, and finding nearby terms in document search. Each of these application areas already has a well-developed scientific literature and engineering practices, and the goal of this paper is not to claim that graded vector techniques are superior to these established practices. Instead we show that graded vector techniques can at least be applied to a wide range of areas with interesting results.

3 First Example: Encoding of Orthography in Semantic Vectors

This section introduces graded semantic vectors using a motivating example. Our first application of vector representations to model graded quantities and relationships was in encoding orthography, that is, in modelling a word as an ordered list of characters in given positions. This work was presented at an earlier Quantum Interaction conference [8]. This paper also discusses previous work in this area, including alternative encoding and similarity measurement techniques such as Levenshtein distance.

Orthographic representation involves various challenges. It is motivated psychologically (trying to reproduce human behavior when reading strings of alphabetic characters) and technologically (for example for spelling correction). Such a system should recognize that *line* and *link* are written similarly, it should see similarities between *nile* and *line* but should still distinguish them, and so on. The system should have a way to give a lower weight to internal differences, because humans are particularly robust to word-internal character changes.

The solution technique presented in [8] is outlined in Figure 1, and illustrates much of the machinery that will be used more generally in this paper. We start with distinct vectors for each character in the alphabet to be used. One approach is to select these randomly. (Note that throughout this paper the notation for a concept and the vector representing that concept will often be conflated, so in vector equations w will often refer to the vector representing the character w.)

The simplest way to build a word vector out of the vectors for its constituent characters is to add these character vectors together to make word vectors, for example, given vectors for the characters *l*, *i*, *n* and *e*, the vector for the word *line* would be $l+i+n+e$. However, due to the commutativity of vector addition, this process leads to exactly the same result for the word *nile*.

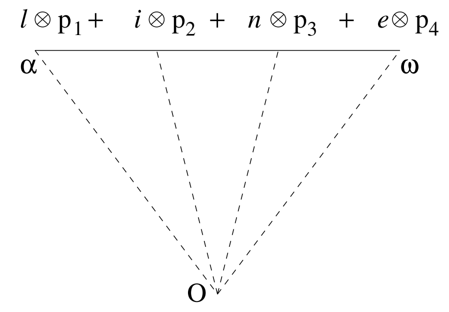
$l \otimes p_1 +$ $i \otimes p_2$ + $n \otimes p_3$ + $e \otimes p_4$
α
ω
O

Fig. 1. Interpolation to generate four demarcator vectors and encode the string *line*.

Table 1. Pairwise similarities between orthographic vectors for *line* and other words.

line	line	1.0	line	lint	0.73
line	lime	0.73	line	nile	0.76
line	file	0.62	line	curve	0.32
line	lie	0.82	line	of	-0.02

This directional encoding problem is often addressed by introducing another pairwise operator on vectors, known as a "binding" operator, such as the convolution product. That is, for two vectors a and b, there is a product $a \otimes b$ defined in such a way that it is naturally dissimilar to both a and b, but if a and a' are similar then $a \otimes b$ and $a\prime \otimes b$ are also similar. Binding operators have a natural inverse, which we shall refer to as a "release" operator.

Now suppose we define vectors for each of the positions in a word: for example, for a word with four letters, these might be written p_1, p_2, p_3, and p_4. Then the string *line* will become represented as $l \otimes p_1 + i \otimes p_2 + n \otimes p_3 + e \otimes p_4$. This leaves the question of how to define the vectors p_i. This is done by interpolation between a pair of *demarcator* vectors which we will call α and ω. These can be selected at random (which in high dimensions means they will be roughly orthogonal), or using some other method. Then $p_1 = \alpha$, $p_2 = \frac{2}{3}\alpha + \frac{1}{3}\omega$, $p_3 = \frac{1}{3}\alpha + \frac{2}{3}\omega$, and $p_4 = \omega$. This whole setup is depicted in Figure 1.

Some examples of the similarities between pairs of words obtained using this method are shown in Table 1. These pairwise similarities were obtained using 200-dimensional complex vectors, with random vectors for the vector for each letter and the α and ω vectors, circular convolution for the binding operation (which for complex vectors is conveniently given by the pairwise addition of complex arguments [9]), and cosine similarity as the similarity measure.[3]

These results were computed using the Semantic Vectors package, a freely-available open-source software package described in [6]. It's easy to see that words that 'look' similar to one another have higher similarity scores. Moving letters around reduces similarity, as does substituting one letter for another. Vectors for words with different numbers of letters are produced in the obvious way by dividing the distance between α

[3] For more on the use of complex and binary vectors in such representations, see [6, 10].

and ω vectors appropriately. Examples and results from using this method are presented in [8], which also reviews some of the observed cognitive features of lexical encoding in human word recognition experiments, and shows that the orthographic encoding technique described here parallels many of these features.

4 A General Approach to Encoding Graded Attributes

The method for orthographic encoding summarized in the previous section can readily be adapted to model all kinds of other graded concepts. The basic idea is the same throughout. Let C be a set of concepts we wish to model, and let $f : C \to \mathbb{R}$ be a mapping from C to the real numbers that describes a property of the concepts in C. For example, f may be a function that takes a data record for a particular vehicle, and returns its weight or fuel efficiency.

Our goal is to create a semantic vector model for these concepts, so that more similar concepts are closer to one another in this space, and other properties and behaviors can ideally be recovered and predicted. This is done using a Vector Symbolic Architecture or VSA [11], which is a vector space V equipped with a binding operator $\otimes : V \times V \to V$, such as as the convolution product of complex vectors introduced in Section 3. The symbol $\otimes$ is adapted from the tensor product symbol used in linear algebra. Many applications of vector composition, including quantum mechanics, and more recently artificial intelligence (e.g., [9, 10]) have been explored. Throughout this paper, the result of binding is a vector in the original space rather than a higher order tensor or any other object. The algebraic properties of VSAs in general, the computational properties of several example implementations, and related work in this rapidly developing area is discussed in several papers including [12, 10].

To encode tabular data, demarcator vectors α and ω are selected just as in Section 3. The, for each function f, its range over the whole of C is computed, giving the minimum value $f_{\min} \in \mathbb{R}$ and maximum value $f_{\max} \in \mathbb{R}$ of f over C. A value $f(c) \in \mathbb{R}$ is then modelled by interpolation just as in the orthographic example, by the vector

$$D(f(c)) = \frac{f_{\max} - f(c)}{f_{\max} - f_{\min}} \alpha + \frac{f(c) - f_{\min}}{f_{\max} - f_{\min}} \omega.$$

The subtractions in this equation are merely shorthand for distances when we know that one quantity is larger than another: the equation can perhaps be most easily understood as “a vector between α and ω representing $f(c)$ in proportion to its distance from $f_{\min}$ and $f_{\max}$”. Global information about the range of f over the whole of C is necessary for $f(c)$ to be computed, which has so far been accomplished by preprocessing the entire dataset before indexing the individual concepts.

Since each concept may have multiple attributes (there are potentially many functions f), it is important to record which attribute took which value. The binding operation features here also. An elemental vector $E(f)$ is generated for each function f. Elemental vectors are described in [10] and earlier works: they are used as building blocks for learning semantic vectors, and can be obtained in many ways. Thanks to the near-orthogonality of most vectors in high-dimensional spaces, even random allocation guarantees near-orthogonality of elemental vectors, which is enough to provide good results in many applications.

Now we can easily define a semantic vector $S(c)$ for each concept in C, using the definition

$$S(c) = \sum E(f) \otimes D(f(c)), \qquad (1)$$

where the sum is taken over all the available functions f and uses the standard vector sum operator $+$.

4.1 Generalized Quantum or Quantum-like Properties

The Quantum Interaction audience will be particularly interested to know what properties make this a generalized quantum or quantum-like representation. These include properties of semantic vector models in general:

- The use of Hilbert spaces to model concepts and the scalar product and projection operators to measure similarity.
- The geometric foundation this gives for logical and probabilistic interpretations.

These properties are well-known to the community and have been emphasized in the literature for at least a decade [4, 5]. A more explicit analysis and evaluation of the probabilistic interpretation is also in-progress in another paper by the authors (in press). Quantum-like properties of the graded representation defined in Equation 1 include:

- For discrete graded properties such as character positions in a word, the technique of quantizing a space of continuous values is similar to that used for modelling angular momentum in quantum mechanics [13].
- The superposition of bound products leads to an *entangled* representation, in the sense that the sum of these products cannot be factorized into the product of two individual vectors. This is related to the description of such systems as fully distributed or holographic [9], meaning that each concept and its contributing factors is represented over several dimensions, and each dimension is used as a feature in the representation of several different concepts and semantic properties.
- The representation can be quantized, in the sense that it can be used to categorize concepts using vague predicates such as “Which items are old?” or “Which items are heavy?”

This said, there is still much to investigate in this area. The quantization of concepts using vague predicates has yet to be implemented and tested effectively. Also, the relationship between various binding operations and quantum models should be explored further: for example, in which VSA’s is the operation “binding with a” self-adjoint, and in which cases is it part of a more general family or operators? Similar questions are currently being asked in cognitive science [14], and we anticipate that this will remain an active topic for research and discussion.

The rest of this paper is devoted to example models, which the authors hope will illustrate the practical usefulness of graded semantic vectors, and encourage deeper theoretical research with technological use-cases directly in mind.

Table 2. Nearest neighbors in a model built from tabular data, with each distinct value treated as a random elemental vector.

J. Adams		T. Roosevelt	
J. Adams	1.00	T. Roosevelt	1.00
Jefferson	0.063	Coolidge	0.072
G.H.W. Bush	0.061	Van Buren	0.067
Washington	0.050	Fillmore	0.059
G.H.W. Bush	0.056	J. Q. Adams	0.046
G. W. Bush	0.048	Taft	0.044

Table 3. Nearest neighbors with values treated as orthographic or graded numeric vectors.

J. Adams		T. Roosevelt	
J. Adams	1.00	T. Roosevelt	1.00
J. Q. Adams	0.266	Coolidge	0.340
Jackson	0.198	L. B. Johnson	0.314
Washington	0.197	Eisenhower	0.314
Buchanan	0.196	Hoover	0.309
Jefferson	0.196	Wilson	0.307

5 Tabular Data and Continuous Quantities

Tabular datasets are a very common case where rows refer to concepts and columns refer to attributes or properties of those concepts, and which often contain graded values. The example presented in this section is a summary of the case-study in [10, §6.5], which explores different methods for building a semantic vector model to represent a tabular dataset listing the Presidents of the USA (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 data values (approximating the standard approach of treating numbers as unique terms), 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 2. The nearest neighbors tend to come from exact matches: for example, Adams and Jefferson both died in the same year (1824), and Roosevelt and Coolidge both died at the same age (60). There are several erroneous similarities and these are generally poor results.

The indexing method was then improved in two ways. Firstly, orthographic vectors (in the sense of Section 3) were used for the textual columns. Secondly, for the columns involving time, graded vectors were used. Results using this method are shown in Table 3. 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 $E(\text{column}) \otimes D(year)$, where D again refers to a demarcator vector. Note the way the column is important, because it gives both the property to be

Table 4. Nearest neighbors for date-specific searches.

Random Elemental Vectors				Numeric Vectors			
Took office 1800		Born 1800		Took office 1800		Born 1800	
Taylor	0.030	Fillmore	0.17	Jefferson	0.216	Taylor	0.261
Eisenhower	0.022	F. D. Roosevelt	0.040	J. Adams	0.198	Buchanan	0.247
Jefferson	0.019	Taft	0.023	J. Q. Adams	0.190	Hayes	0.243
Carter	0.019	G.H.W. Bush	0.018	Madison	0.185	B. Harrison	0.238
Reagan	0.018	Van Buren	0.016	Monroe	0.182	Harding	0.237

searched for, and the appropriate endpoints. Results for year of birth and year of taking office near to 1800 are given in Table 4. 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.

This technique can be extended to sort rows with respect to the magnitude of the entry in a particular column by measuring the similarity between the vector representation of each row and $E(\text{column}) \otimes D(\alpha)$ and/or $E(\text{column}) \otimes D(\omega)$. Figure 2 shows the correlation between the age at inauguration of each of the presidents in the data set and scores produced by two methods of estimating their relative ages at inauguration from their row vector representations directly. For these experiments, binary vector representations generated using the Binary Spatter Code [15] were used to represent the data set. In the first method (labeled α), the similarity between the vector product $E(\text{aie\_column}) \otimes D(\alpha)$ (where aie_column is the age at inauguration column header) and $S(\text{president})$, the row vector representation of each president, was used as an estimate of relative age at inauguration, with younger ages receiving higher scores. The metric of similarity ($\text{sim}(x, y)$) in this case is $1 - \frac{2}{n}\text{HammingDistance(x, y)}$, where n is the dimensionality of the binary vectors concerned. In the second method (α - ω), the similarity to the vector product $E(\text{column}) \otimes D(\omega)$ was also taken into account, such that the score used to estimate relative age was $\text{sim}\big(S(\text{president}), E(\text{aie\_column}) \otimes D(\alpha)\big) - \text{sim}\big(S(\text{president}), E(\text{aie\_column}) \otimes D(\omega)\big)$. As illustrated in Figure 2, both approaches correlate well with the encoded age at inauguration, with the $\alpha - \omega$ method obtaining Pearson correlations above 0.9 even at relatively low dimensionalities.

6 Inferring Proximity

Binding to graded vectors permits a novel mode of inference — proximity can be inferred by comparing the results of "release" operations using different elemental vectors if a common set of demarcator vectors is applied across columns. For example, if $S(\text{washington})$ = $E(\text{took\_office})$ $\otimes D(1789)$ + $E(\text{left\_office})$ $\otimes D(1797)$ and $S(\text{harding})$ = $E(\text{took\_office})$ $\otimes D(1921)$ + $E(\text{left\_office})$ $\otimes D(1923)$, we should be able to infer that Harding had a shorter term of office as we would anticipate

$$\text{sim}\big(S(\text{harding}) \oslash E(\text{took\_office}), S(\text{harding}) \oslash E(\text{left\_office})\big) >$$
$$\text{sim}\big(S(\text{washington}) \oslash E(\text{took\_office}), S(\text{washington}) \oslash E(\text{left\_office})\big).$$

In practice, however, it is necessary to project the results of the "release" operation onto the two-dimensional plane spanned by the α and ω vectors used to construct the graded

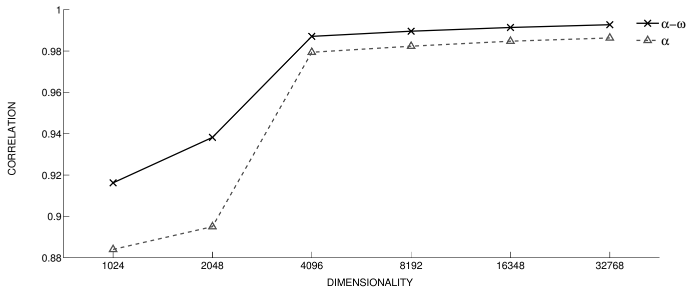

Fig. 2. Pearsons's *r* correlation at various dimensions between age at inauguration (aie) and:
Label α: Similarity of the president vector to $E(\text{aie\_col}) \otimes D(\alpha)$
Label $\alpha - \omega$: Similarity of the president vector to $E(\text{aie\_col}) \otimes D(\alpha)$ minus similarity to $E(\text{aie\_col}) \otimes D(\omega)$.

vectors of interest [4]. Proximity between two vector products (e.g. $V1 = S(\text{washington}) \oslash E(\text{took\_office})$ and $V2 = S(\text{washington}) \oslash E(\text{left\_office})$) can then be calculated by applying distance metrics to their projections on the $\alpha\omega$ plane:

$$\begin{aligned} ed(V1, V2) &= \sqrt{(\langle V1|\alpha\rangle - \langle V2|\alpha\rangle)^2 + (\langle V1|\omega\rangle - \langle V2|\omega\rangle)^2} \\ sp(V1, V2) &= \quad \langle V1|\alpha\rangle \times \langle V2|\alpha\rangle + \langle V1|\omega\rangle \times \langle V2|\omega\rangle \end{aligned}$$

The first of these metrics, $ed(V1, V2)$, estimates the euclidean distance between projections on the $\alpha\omega$ plane. The second, $sp(V1, V2)$, estimates the scalar product of these projections, providing a similarity metric. We now present evaluations of these metrics.

6.1 Proximal dates

For $ed(V1, V2)$, we evaluate the extent to which this metric can infer the relative age at which a candidate took office (this information was explicitly encoded for the evaluation illustrated in Figure 2) from the encoded demarcator vector representations of their birth and inauguration years. Binary vector representations of the data set were again generated using the Binary Spatter Code [15], and the correlation between actual age at inauguration and age as estimated by $ed\big(S(\text{president}) \oslash E(\text{birth\_year}), S(\text{president}) \oslash E(\text{inauguration\_year})\big)$. The results of these experiments are shown in Figure 3, which plots Pearson's correlation between inferred age and actual age at different dimensionalities. These results suggest that though strong correlation can be obtained, the dimensionality required to achieve this is orders of magnitude higher than that required when searching for explicitly encoded values.

[4] With binary vectors this is accomplished by comparing $S(\text{president}) \oslash E(\text{column\_header})$ to D(α) and D(ω) using $1 - \frac{2}{n}\text{HammingDistance}(\text{x}, \text{y})$. This corresponds to the scalar product if binary vectors are viewed as bipolar vectors {1,-1}.

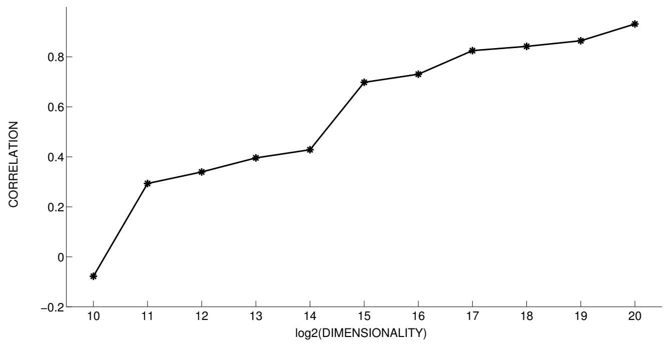

Fig. 3. Inferred correlation between age at inauguration and proximity between projections of $S(\text{president}) \oslash D(\text{year\_birth\_column})$ and $S(\text{president}) \oslash D(\text{year\_inauguration\_column})$ on the α-ω plane. Y axis = Pearson's *r*. X axis = $\log_2$(dimensionality in bits).

Table 5. Comparison between proximity-based and conventional search for phrase "chronic pain". Score gives the number of standard deviations above the mean score across all documents in the corpus.

Proximity-based search	Conventional search
80.39: Chronic pain [letter] [comment].	13.53: Pain.
70.32: Management chronic pain.	13.53: Pain control.
69.35: Chronic pain depression.	13.53: No pain, no pain.
65.12: Pain chronic pancreatitis.	13.42: Chronic pelvic pain.
61.61: Chronic pain search.	12.90: Management of chronic pain [letter]

6.2 Proximal terms

Another application for this sort of inference involves identifying documents within which two terms occur in proximity. This can be accomplished by creating document vectors as the superposition of bound products between vectors representing terms and graded vectors representing their positions within the document. If a common α and ω vector are used throughout, $sp(V1, V2)$ can be used to find documents in which the terms of interest occur in proximity. Table 5 shows the highest-ranked documents in response to a search for the phrase "chronic pain" in two 1024-dimensional Inverse Document Frequency weighted binary vector space models of the OHSUMED corpus of biomedical abstracts [16] (terms occurring more than 100,000 times were excluded from the indexing process). The first of these models (proximity-based search) estimates proximity as $sp\big(S(\text{document}) \oslash E(\text{term1}), S(\text{document}) \oslash E(\text{term2})\big)$. The second (conventional search) uses a simple Random Indexing approach, in which document vectors are constructed as weighted superpositions of elemental term vectors.

These results suggest that the proximity-based approach is more likely to retrieve documents containing the phrase in its entirety. To test this hypothesis, we extracted all two-word terms (n=176,246) from the Specialist Lexicon [17] provided by the Na-

Table 6. Comparison between proximity-based and conventional search. The mean and *median* precision at k=50 are shown, with best results at each level in boldface.

Instances of term in results	Proximity-based search		Conventional search	
$\geq$ 1 (n=24,290)	**13.02**	***4***	9.42	***4***
$\geq$ 10 (n=5,116)	**43.44**	***36***	31.18	*24*
$\geq$ 20 (n=2,519)	**62.14**	***60***	44.98	*40*
$\geq$ 30 (n=1,407)	**75.58**	***80***	56.44	*56*
$\geq$ 40 (n=801)	**83.36**	***92***	65.66	*70*

tional Library of Medicine, and searched for them in the OHSUMED set using both conventional and proximity-based procedures, retrieving the top 50 nearest neighboring documents in each case, and evaluating whether or not each of these contained an exact match (with both terms and documents converted to lowercase) for the two-word term in question. As one might anticipate, many of the phrases did not occur in the OHSUMED corpus, so we report results for those phrases that were identified by at least one of the procedures only (n=24,290). The results are shown in Table 6 which gives the mean and *median* precision at k=50 across models, stratified in accordance with the number of instances of the term retrieved by at least one of the models. Results are stratified in this way so as to reveal more pronounced differences in performance for examples in which documents containing the term are relatively frequent, which is where we would anticipate proximity search having an advantage (as if only a few documents contain the term, both proximity-based and conventional search would be expected to retrieve these within the top 50 results).

Interestingly, the applicability of these metrics is task-specific, and each of the metrics was abandoned early as an approach to the task for which it was not formally evaluated. The first metric evaluated, $ed(V1, V2)$, is not productive as a means to retrieve documents containing terms in proximity, and tends to retrieve documents containing only one of the two words in the term of interest. This may occur because it will reward instances in which only one word occurs if this word occurs close to the end of a document, as in this case the projections of the vector representations of both terms on the α axis of the $\alpha\omega$ plane will be small, leading to a small euclidean distance (suggesting close proximity). In contrast the scalar product ($sp(V1, V2)$), which multiplies the magnitudes of these projections, will be low (suggesting great distance). As such ($sp(V1, V2)$) rewards the presence of both words and is a better fit for the "proximal terms" task. However, this metric was not productive in the "proximal dates" task, where the absence of one of the elements of comparison is not an issue, and the distances of interest are small in comparison with the range that were represented.

7 Summary and Technology Potential

This paper has demonstrated that graded data can be represented in a distributional model along with discrete and relational data. 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. The potential for such combined semantic models is considerable. For example, inference in the biomedical domain often involves a combination of information derived from textual sources and quantitative measurements. As further work, we intend to apply the techniques developed in this paper to combine the narrative text and structured data (such as lab values) to generate more comprehensive representations of clinical and biomedical data sets. Similarly, the orthographic encoding technique can clearly be applied to practical language engineering tasks such as spelling correction: this work would involve comparing and evaluation the orthographic encoding method described here with other established textual similarity measures for accuracy and computational cost, to establish the relative strengths of different methods and key integration opportunities.

8 Acknowledgment

This research was supported by NIH/BD2K supplement R01LM011563-02S1.

References

1. T. Cohen and D. Widdows, “Empirical distributional semantics: methods and biomedical applications,” *Journal of biomedical informatics*, vol. 42, no. 2, pp. 390–405, 2009.
2. G. Salton and M. McGill, *Introduction to modern information retrieval*. New York, NY: McGraw-Hill, 1983.
3. T. Landauer and S. Dumais, “A solution to Plato’s problem: The latent semantic analysis theory of acquisition,” *Psychological Review*, vol. 104, no. 2, pp. 211–240, 1997.
4. K. van Rijsbergen, *The Geometry of Information Retrieval*. Cambridge University Press, 2004.
5. D. Widdows, *Geometry and Meaning*. CSLI Publications, 2004.
6. D. Widdows and T. Cohen, “Real, complex, and binary semantic vectors,” in *Sixth International Symposium on Quantum Interaction* (J. Busemeyer, F. Dubois, A. Lambert-Mogiliansky, and M. Melucci, eds.), 2012.
7. R. Lebret and R. Collobert, “Word emdeddings through Hellinger PCA,” *arXiv preprint arXiv:1312.5542*, 2013.
8. T. Cohen, D. Widdows, M. Wahle, and R. Schvaneveldt, “Orthogonality and orthography: Introducing measured distance into semantic space,” *Proceedings of the Seventh International Conference on Quantum Interaction, Leicester, UK, 2013*, 2013.
9. T. A. Plate, *Holographic Reduced Representations: Distributed Representation for Cognitive Structures*. CSLI Publications, 2003.
10. D. Widdows and T. Cohen, “Reasoning with vectors: a continuous model for fast robust inference,” *Logic Journal of IGPL*, vol. 23, no. 2, pp. 141–173, 2015.
11. R. W. Gayler, “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, 2004.
12. M. A. Kelly, D. Blostein, and D. Mewhort, “Encoding structure in holographic reduced representations.,” *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, vol. 67, no. 2, p. 79, 2013.
13. D. Bohm, *Quantum Theory*. Prentice-Hall, 1951. Republished by Dover, 1989.

14. A. Khrennikov, I. Basieva, E. N. Dzhafarov, and J. R. Busemeyer, “Quantum models for psychological measurements: An unsolved problem,” *PloS one*, vol. 9, no. 10, p. e110909, 2014.
15. P. Kanerva, “Binary spatter-coding of ordered k-tuples,” *Artificial Neural Networks—ICANN 96*, pp. 869–873, 1996.
16. W. Hersh, C. Buckley, T. J. Leone, and D. Hickam, “OHSUMED: an interactive retrieval evaluation and new large test collection for research,” *Proceedings of the 17th ACM SIGIR conference on research and development in information retrieval*, pp. 192–201, 1994.
17. N. C. f. B. Information, U. S. N. L. o. M. . R. Pike, B. MD, and . Usa, “SPECIALIST Lexicon and Lexical Tools,” Sept. 2009.