Orthogonality and Orthography: Introducing Measured Distance into Semantic Space

Trevor Cohen[1], Dominic Widdows[2], Manuel Wahle[1], and Roger Schvaneveldt[3]

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

Abstract. This paper explores a new technique for encoding structured information into a semantic model, for the construction of vector representations of words and sentences. As an illustrative application, we use this technique to compose robust representations of words based on sequences of letters, that are tolerant to changes such as transposition, insertion and deletion of characters. Since these vectors are generated from the written form or orthography of a word, we call them 'orthographic vectors'. The representation of discrete letters in a continuous vector space is an interesting example of a Generalized Quantum model, and the process of generating semantic vectors for letters in a word is mathematically similar to the derivation of orbital angular momentum in quantum mechanics. The importance (and sometimes, the violation) of orthogonality is discussed in both mathematical settings.This work is grounded in psychological literature on word representation and recognition, and is also motivated by potential technological applications such as genre-appropriate spelling correction. The mathematical method, examples and experiments, and the implementation and availability of the technique in the Semantic Vectors package are also discussed.

Keywords: Distributional Semantics, Orthographic Similarity, Vector Symbolic Architectures

1 Introduction

The relationships between words, their representation in text, concepts in the mind, and objects in the real world, has been the source of inquiry over many centuries. Empirical, distributional paradigms have been shown to successfully derive human-like estimates of semantic distance from large text corpora, and recent developments in this area have mediated the enrichment of distributional models with structural information, such as the relative position of terms [1, 2], and orthographic information describing the configuration of characters from which words are composed [3, 4]. This paper extends these beginnings in three principal ways. First, we propose a new and very simple method for encoding structure into semantic vectors, using a quantization of the space between two extreme 'demarcator vectors'. This vector generation method performs well in experiments and has some key computational advantages. Second, the method is general enough to be applied within a wide range of Vector Symbolic Architectures (VSAs), including Circular Holographic Reduced Representations (CHHRs) that use complex vectors [5]. Thirdly, we investigate some higher-level compositions as well, demonstrating some early results with compositional representations for sentences.

Each of these developments is related to quantum interaction as follows. Demarcator vector generation is similar in a sense to the derivation of orbital angular momentum values in quantum mechanics, in that they use the same mathematics. The application within many VSAs, including those over complex vector space, makes the new methods available within algebras that are particularly related to generalized quantum models. Finally, compositional methods are important in research on generalized quantum structures, and the way many levels of representation are combined seamlessly in the compositional models presented here should be of interest to researchers in the field.

2 Orthogonality in Distributional and Orthographic Models

Geometric methods of distributional semantics derive vector representations of terms from electronic text, such that terms that occur in similar contexts will have similar vectors [6, 7]. While such models have been shown to approximate human performance on a number of cognitive tasks [8, 9], they generally do not take into account structural elements of language, and as consequently have been referred to, at times critically, as "bags of words" models. Emerging approaches to semantic space models have leveraged reversible vector transformations to encode additional layers of meaning into vector representations of terms and concepts. Examples include the encoding of the relative position of terms [1, 10], syntactic information [11], and orthographic information [4].

The general approach used depends upon the generation of vector representations for terms, and associating reversible vector transformations with properties of interest. In accordance with terminology developed in [12], we will refer to the vector representations of atomic components such as terms as *elemental vectors*. Elemental vectors are constructed using a randomization procedure such that they have a high probability of being mutually orthogonal, or close-to-orthogonal. This adds robustness to the model, by making it highly improbable that elemental vectors would be confused with one another, despite the distortion that occurs during training. However, it also introduces the implicit assumption that elemental vectors are unrelated to one another, which means that models generated in this way must be composed of discrete elements.

This limitation notwithstanding, this approach has allowed for the integration of structured information into distributional models of meaning. From the perspective of cognitive psychology, this is desirable as it presents the possibility of a unified term representation that can account for a broad range of experimental phenomena. Recent work in this area has leveraged circular convolution to generate vectors representing the orthographic form of words [3], and integrate these with a geometric model of distributional semantics [4]. Vector representations of orthographic word form are generated by using circular convolution to generate bound products representing the component bigrams of the term concerned, including non-contiguous bigrams. Karchergis *et al* give the following example ($\circledast$ indicates binding using circular convolution) [4]:

$$\begin{aligned} word =& w + o + r + d + w \circledast o + o \circledast r + r \circledast d \\ & + w \circledast o + (w \circledast \_) \circledast r + (w \circledast \_) \circledast d + (w \circledast o) \circledast r + ((w \circledast \_) \circledast r) \circledast d \\ & + (w \circledast \_) \circledast d + (o \circledast \_) \circledast d + r \circledast d + ((w \circledast o)\_) \circledast d + (o \circledast r) \circledast d \end{aligned}$$

The vector representation for the term “word” is generated by combining a set of vectors representing unigrams, bigrams, and trigrams of characters. It is a characteristic of the model employed that each of these vectors have a high probability of being mutually orthogonal, or close-to-orthogonal. So, for example, the vectors representing the trigrams $((w \circledast \_) \circledast r)$ and $((w \circledast o) \circledast r)$ will be dissimilar from one another. Consequently it is necessary to explicitly encode all of the n-grams of interest, including gapped trigrams (such as “w_ r”) to provide flexibility. From the perspective of computational complexity, this is less than ideal, as the number of representational units that must be generated and encoded is at least quadratic to the length of the sequence.

Rather than explicitly encoding character position precisely (with respect to some other character, or the term itself), alternative models of orthographic representation allow for a degree of uncertainty with respect to letter position. These approaches measure the relatedness between terms on the basis of the similarity between probability distributions assigned to the positions of each matching character [13, 14], providing a more flexible measure of similarity. However, on account of the constraints we have discussed, only orthographic representations based on discrete bigrams or the exact position of characters have been combined with other sorts of distributional information in an attempt to generate a holistic representation to date [3, 15]. Imposing near-orthogonality adds robustness, but also necessitates ignoring potentially useful information to do with structure, namely the proximity between character positions within a word. Consequently, we have selected the generation of orthographic representations as an example application through which to illustrate the utility of our approach.

The paper proceeds as follows. First, we present the mathematical language we will use to describe the operators provided by VSAs, a family of representational approaches based on reversible vector transformations [16]. Next, we will describe an approach we have developed through which the distance between elemental vectors, and hence bound products, can be predetermined. In the context of an illustrative application for orthographic modeling, we show that this approach permits the encoding of structural information to do with proximity, rather than absolute position, into a distributional model. We then discuss a relationship between this approach and quantum mechanics, and conclude with some experimental results and example applications.

3 Mathematical Structure and Methods

3.1 Vector Symbolic Architectures (VSAs)

The reversible vector transformations we have discussed are a distinguishing feature of a family of representational approaches collectively known as VSAs [16]. In our experiments the VSAs we will use are Kanerva’s Binary Spatter Code (BSC), which uses binary vectors [17], and Plate's CHRR [5], which uses complex vectors where each dimension represents an angle between $-\pi$ and π, using the implementation developed in [10]. In addition, we will use an approach based on permutation of real vectors [2].

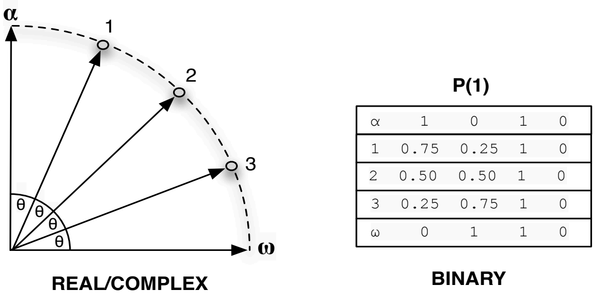
Binding is the primary operation facilitated by VSAs (in addition to standard operators for vector superposition and vector comparison). Binding is a multiplication-like operator through which two vectors are combined to form a third vector C that is *dissimilar from* either of its component vectors A and B. We will use the symbol "$\otimes$" for binding, and the symbol "$\oslash$" for the inverse of binding for the remainder of this paper. It is important that this operator be invertible: if C = A $\otimes$ B, then A $\oslash$ C = A $\oslash$ (A $\otimes$ B) = B. In some models, this recovery may be approximate, but the robust nature of the representation guarantees that A $\oslash$ C is similar enough to B that B can easily be recognized as the best candidate for A $\oslash$ C in the original set of concepts. Thus the invertible nature of the bind operator facilitates the retrieval of the information it encodes.

In the case of the BSC, elemental vectors are initialized by randomly assigning 0 or 1 to each dimension with equal probability. Pairwise exclusive or (XOR) is used as a binding operator: X$\otimes$Y $=$ X XOR Y. As it is its own inverse, the binding and decoding processes are identical ($\otimes$=$\oslash$). For superposition, the BSC employs a majority vote: if the component vectors of the bundle have more ones than zeros in a dimension, this dimension will have a value of one, with ties broken at random.

In CHRR, binding through circular convolution is accomplished by pairwise multiplication: $X \otimes Y = \{X_1Y_1, X_2Y_2, ...X_{n-1}Y_{n-1}, X_nY_n\}$, which is equivalent to addition of the phase angles of the circular vectors concerned. Binding is inverted by binding to the inverse of the vector concerned: $X \oslash Y = X \otimes Y^{-1}$, where the inverse of a vector is its complex conjugate. Elemental vectors are initialized by randomly assigning a phase angle to each dimension. Superposition is accomplished by pairwise addition of the unit circle vectors, and normalization of the result for each circular component. In the implementation used in our experiments, normalization occurs after training concludes, so the sequence in which superposition occurs is not relevant.

Our real vector implementation follows the approach developed by Sahlgren and his colleagues [2], and differs from the binary and complex implementations, in that elemental vectors are "bound" to permutations, rather than to other vectors. Elemental vectors are constructed by creating a high-dimensional zero vector (on the order of 1000 dimensions), and setting a small number of the dimensions of this vector (on the order of 10) to either +1 or -1 at random. The permutations utilized consist of shifting all of the elements of a given vector n positions to the right, where each value n is assigned to, or derived from, the information it is intended to encode. In the case of our orthographic model, this information consists of the character occurring in a particular position, so we have used the ASCII value of the character concerned as n. Binding is reversed by permuting all of the elements of the vector n positions to the left. Superposition is accomplished by adding the vectors concerned, and normalizing the result.

In all models, the "random" initiation of elemental vectors is rendered deterministic by seeding the random number generator with a hash value derived from a string or character of interest following the approach developed in [18]. This retains the property of near-orthogonality where desired, while ensuring that random overlap between elemental vectors is consist across experiments.



α	1	0	1	0
1	0.75	0.25	1	0
2	0.50	0.50	1	0
3	0.25	0.75	1	0
ω	0	1	1	0

Fig. 1. Interpolation to Generate Five Demarcator Vectors. P(1) = probability of 1.

3.2 Measured Similarity

The first step in our approach involves generating a set of vectors that are a fixed distance apart, which we will refer to as *demarcator vectors* ($D(\mathrm{position})$), as illustrated in Figure 1. The first pair of demarcator vectors are conventional elemental vectors $D(\alpha)$ and $D(\omega)$, constructed randomly such that they have a high probability of being mutually orthogonal or close-to-orthogonal. To ensure this with certainty, we render $D(\omega)$ orthogonal to $D(\alpha)$ using the quantum negation procedure, or its binary approximation, described in [19] and [20] respectively. The remaining demarcator vectors are generated by interpolation. In the continuous vector spaces, this is accomplished by subdividing the 90^{o} angle between $D(\alpha)$ and $D(\omega)$ and generating the corresponding unit vectors. In binary vector space, this is accomplished by weighting the probability of assigning a 1 when $D(\alpha)$ and $D(\omega)$ disagree in accordance with the desired distance between the new demarcator vector and each of these extremes[4]. As the vectors representing adjacent numbers are approximately equidistant, the distances between vectors pairs representing numbers the same distance apart should also be approximately equal (e.g. $sim(D(1),D(3)) \approx sim(D(2),\, D(4))$.

Table 1 illustrates the pairwise similarities between a set of five demarcator vectors constructed in this manner. In the binary case vectors of dimensionality 32,000 are used, in the complex case vectors of dimensionality 500 are used, and in the real case, vectors of dimensionality 1,000 are used. In both cases, the relatedness between demarcator vectors a fixed distance apart is approximately equal. For example, the similarity between all pairs of demarcator vectors two positions apart (e.g. 1 and 3) is approximately 0.5 in the binary implementation, and 0.71 in the complex and real vector implementations. In the binary implementation, the difference in relatedness is proportional to the difference in demarcator position. This is not the case in the complex or real implementations, where the drop in similarity between demarcator vectors becomes progressively steeper. This is an artifact of the metric used to measure similarity in each case. With binary vectors, $2 \times (0.5 - \mathrm{normalized\ Hamming\ distance})$ is used, but with continuous vectors the cosine distance metric is used. While a proportional decrement could be

[4] The same random number sequence must be used for all vectors in a demarcator set, so that a consistent random value for each bit position is compared to the relevant thresholds.

Table 1. Pairwise Similarity between Demarcator Vectors

	BINARY						COMPLEX						REAL				
	α	**1**	**2**	**3**	ω		α	**1**	**2**	**3**	ω		α	**1**	**2**	**3**	ω
α	1.00	0.75	0.49	0.25	0.00	α	1.00	0.92	0.71	0.38	0.00	α	1.00	0.92	0.71	0.38	0.00
1	0.75	1.00	0.74	0.50	0.25	**1**	0.92	1.00	0.92	0.71	0.38	**1**	0.92	1.00	0.92	0.71	0.38
2	0.49	0.74	1.00	0.75	0.51	**2**	0.71	0.92	1.00	0.92	0.71	**2**	0.71	0.92	1.00	0.92	0.71
3	0.25	0.50	0.75	1.00	0.75	**3**	0.38	0.71	0.92	1.00	0.92	**3**	0.38	0.71	0.92	1.00	0.92
ω	0.00	0.25	0.51	0.75	1.00	ω	0.00	0.38	0.71	0.92	1.00	ω	0.00	0.38	0.71	0.92	1.00

obtained by measuring the angle between complex vectors directly (or taking the arccos of this cosine value), we will retain the use of the cosine metric for our experiments.

3.3 Encoding Orthography

Using controlled degrees of non-orthogonality, we can encode information about the positions of letters in words into their vector representations. Like spatial encoding [14] and the overlap model [13], our approach is based upon measuring the difference in position between matching characters. This is accomplished by creating elemental vectors for characters, and binding them to demarcator vectors representing positions. For example, the orthographic vector for the term "word" is constructed as follows:

$$S(\text{word}) \mathrel{+}= E(\text{w}) \otimes D(1) + E(\text{o}) \otimes D(2) + E(\text{r}) \otimes D(3) + E(\text{d}) \otimes D(4)$$

As the vectors for characters are mutually orthogonal or near-orthogonal, bound products derived from different characters will be orthogonal or near-orthogonal also. For example, in the complex vector space used to generate Table 1, $sim(E(\text{w}) \otimes D(1), E(\text{q}) \otimes D(1)) = 0$. Furthermore, the distance between bound products containing the same character will approximate the distance between their demarcator vectors. For example, in the real vector space used to generate Table 1, $sim(E(\text{w}) \otimes D(1), E(\text{w}) \otimes D(2)) = 0.92 = sim(D(1), D(2))$. Ultimately, the similarity between a pair of terms is derived from the distance between their matching characters. If this distance is generally low, the orthographic similarity between these terms will be high[5]. Thus, the models so generated are innately tolerant to variations such as transposition, insertion and deletion of sequence elements.

[5] For terms of different lengths, we elected to construct a set of demarcator vectors for each term. So while $D(\alpha)$ and $D(\omega)$ will be identical, the demarcator for a particular position may differ. It would also be possible to use identical demarcator vectors (by generating a set large enough to accommodate the longest term), which may be advantageous for some tasks.

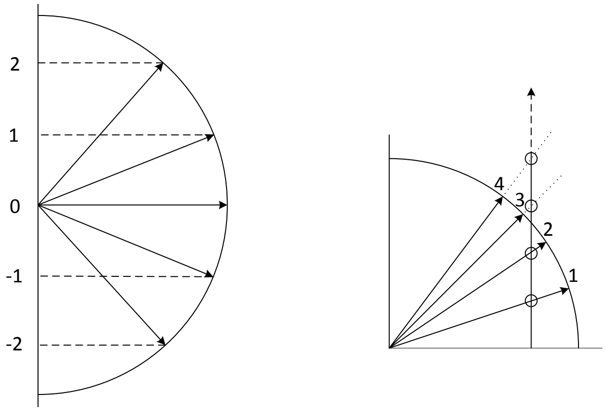


Fig. 2. Orbital angular momentum vectors, as derived from quantized states along a given axis (left), and a related strategy for encoding positions as vectors

4 Demarcator Vectors and Orbital Angular Momentum

Evenly distributing normalized vectors between two orthogonal vectors is one of many strategies we could adopt to generate demarcator vectors. To generalize this process, we can describe it as follows:

1. Construct a line in the vector space with a given starting point and direction.
2. Place demarcators along this line using some dividing strategy.
3. (Optional) Project demarcator vectors onto the unit circle to normalize them.

Two examples of this strategy are illustrated in Figure 2. In the example on the left, the generating line is the vertical axis, the dividing strategy is to mark points at even intervals along this axis, and the projection strategy is to project orthogonally from the vertical axis onto the unit sphere[6]. In the example on the right, we have chosen a generating line parallel to the vertical axis, marked points at even intervals, and projected onto the unit sphere using standard vector normalization. The left-hand strategy will be familiar to some readers: this is precisely the way *orbital angular momentum* states are generated in quantum mechanics. We refer the reader to a text on quantum mechanics for this derivation, e.g., [21, Ch 14]: the process includes solving a wave equation in three dimensions, exploring the angular momentum operator and the commutator relations between its components, and noting that each point on the axis can be mapped to many on the outer sphere (this ambiguity corresponding to the fact that measuring the component of angular momentum along one axis must leave the component along the other axes undetermined according to the Uncertainy Principle). The important point for this discussion is that it is quite standard to derive non-orthogonal vectors for states

[6] In this example we have drawn negative and positive positions, though in practice we have only experimented with nonnegative positions so far.

in this fashion, not only in generalized quantum structures but in quantum mechanics itself. The underlying spherical harmonic functions involved in angular momentum are orthogonal to one another under pairwise integration, but lead to several possible non-orthogonal angular momentum vectors.

The ambiguity of the projection onto the unit sphere in the angular momentum model in more than two dimensions is problematic, and we expect the strategy on the right to be simpler in practice. Note also that both strategies do not distribute vectors evenly around the unit sphere: for example, in the right-hand strategy, the vectors representing positions 3 and 4 are closer to each other than the vectors representing positions 1 and 2. Such flexibility to vary these pairwise similarities between positions along a string is a desirable property, because changes at the beginning of a word may be more significant than changes in the middle [13]. We note also that in our current implementation in the Semantic Vectors package, this generalized strategy works well for real-valued and complex-valued vectors, but is as yet underspecified for binary-valued vectors because (for example) ‘the point half-way between A and B’ is multiply defined using Hamming distance [22]. We are currently investigating appropriate strategies to bring binary vectors into this generalized description.

5 Applications: Orthography, Morphology, Sentence Similarity

Table 2 provides examples of nearest-neighbor search based on orthographic similarity in real, complex and binary vector spaces derived from the widely-used Touchstone Applied Science Associates, Inc. (TASA) corpus. Terms in the corpus that occurred between 5 and 15,000 times were represented as candidates for retrieval. The dimensionality of the real, complex and binary vector spaces concerned was 1000, 500 and 32,000 respectively. These dimensions were chosen so as to normalize the space requirements of the stored vectors across models, and were retained in our subsequent experiments. Our approach successfully recovers orthographically related terms, including terms containing substrings of the original term (“**domini**c” vs. “con**domini**um”); insertions (“**orthography**” vs. “**ortho**photo**graphy**”); substitutions (“**an**g**ular**” vs. “**an**n**ular**”) and transposition of characters (“**w**ah**le**” vs. “**w**ha**le**”).

While we would be hesitant to propose the simple model of orthographic representation we have developed as a cognitive model of lexical coding, it is interesting to

Table 2. Orthographic Similarity

REAL				COMPLEX		BINARY			
CUE	**dominic**	**CUE**	**orthography**	**CUE**	**orthogonality**	**CUE**	**angular**	**CUE**	**wahle**
0.92	dominican	0.92	orthophotography	0.86	orthogonally	0.73	agranular	0.94	whale
0.89	dominion	0.93	photography	0.85	orthogonal	0.70	annular	0.61	awhile
0.88	demonic	0.91	chromatography	0.84	orthodontia	0.67	angularly	0.60	while
0.85	dominions	0.90	orthographic	0.82	ornithology	0.66	gabular	0.60	whales
0.85	condominium	0.90	choreography	0.82	ornithologist	0.66	inaugural	0.60	whaley

Table 3. Comparison with benchmark conditions from [15]. 1X = original dimensionality (used for Table 2). 2X = twice original dimensionality. nb = no binding. ED = $1 - \frac{\text{edit distance}}{\text{combined length}}$.

	BINARY			COMPLEX			REAL		
	1X	2X	2Xnb	1X	2X	2Xnb	1X	2X	ED
Stability	✓	✓	✓	✓	✓	✓	✓	✓	✓
Edge Effects							✓		
Local TL	✓	✓	✓	✓	✓	✓	✓	✓	✓
Global TL									✓
Distal TL									
Compound TL	✓	✓		✓	✓		✓	✓	
Distinct RP	✓	✓	✓	✓	✓	✓	✓	✓	✓
Repeated RP	✓	✓		✓	✓	✓	✓	✓	✓

note that it does conform to the majority of a set of constraints abstracted from lexical priming data by Hannagan and his colleagues [15]. We will describe these constraints in brief here, but refer the interested reader to [15] for further details. The constraints are as follows: (1) **Stability**: a string should be most similar to itself ($\mathrm{sim} >= 0.95$); (2) **Edge Effects**: substitutions at the edges of strings should be more disruptive; (3) **Local Translocations (TL)**: transposing adjacent characters should be less disruptive than substituting both of them; (4) **Global TL**: transposing *all* adjacent characters should be maximally disruptive (5) **Distal TL**: transposing non-adjacent characters should be more disruptive than substituting one, but less than substituting both; (6) **Compound TL**: TL and substitution should be more disruptive than substitution alone; (7) **Distinct Relative Position (RP)**: removing some characters should preserve some similarity; and (8) **Repeated RP**: removing a repeated or non-repeated letter should be equally disruptive[7]. Each constraint is accompanied by a set of test cases, consisting of paired strings, and the degree to which a model meets the constraint is determined from the estimated similarities between these pairs, and the relationships between them.

The extent to which the our models meet these constraints is shown in Table 3. Estimates based on all models consistently meet all constraints aside from those related to Edge Effects, and the Global and Distal TL constraints (in the latter case this is due to the fact that translocation of characters one position apart is *less* disruptive than substituting one of these characters). This represents a better fit to these constraints than comparable models based on letter distribution only (labeled "nb", or not bound). It also represents a better fit than the majority of approaches evaluated against this benchmark previously[15, 23], providing motivation for the further evaluation of a more developed model in the future. While the real model appeared to meet the edge effect related constraint at its original dimensionality, this finding did not hold at higher dimensionality, and was most likely produced by random overlap. This is not surprising given that our

[7] As the randomization procedure makes it very unlikely that the estimates of similarity between any two pairs will be identical, we have considered a difference of $<= 0.05$ to be approximately equal. This mirrors the relaxed constraint that $>= 0.95$ is approximately identical used by Hannagan and his colleagues for the stability constraint.

Table 4. Combining Orthographic and Semantic Similarity

REAL				COMPLEX		BINARY			
CUE	think	**CUE**	**bring**	**CUE**	**eat**	**CUE**	**catch**	**CUE**	**write**
0.57	intend	0.68	bringing	0.44	eats	0.191	catching	0.180	writes
0.56	know	0.62	brings	0.43	ate	0.184	caught	0.173	writer
0.51	thinks	0.53	brought	0.39	meat	0.181	catches	0.172	rewritten
0.51	thinking	0.52	ring	0.36	eaten	0.176	watch	0.172	wrote
0.51	want	0.51	burning	0.36	restate	0.176	teach	0.171	reread

current model does not address edge effects. As suggested earlier, one way to address this issue would be to increase the distance between peripheral demarcator vectors, a customization we plan to evaluate in future work.

The results in Table 4 were obtained by superposing the orthographic vector for each term from the TASA corpus with a semantic vector for the same term generated using the permutation-based approach described in [2], with a 2+2 sliding window. As anticipated by previous work combining orthographic and semantic relatedness [24, 25], the examples suggest that this model is able to find associations between morphologically related terms, including those between English verb roots and past tense forms are related by non-affixal patterns such as “**br**i**ng**:**br**ou**g**ht”. The combination of semantics and orthography is evident in other examples, such as “**t**h**in**k:**int**end”. However, this sensitivity to morphological similarity comes at a cost of introducing false similarity when common letter patterns do not have a semantic significance.

Table 5. Retrieval of Sentences from the TASA Corpus in complex (first two examples) and binary (second two examples) spaces

CUE	**the greater the force of the air the louder the sounds**
0.86	that is the smaller the wavelength the greater the energy of the radiation
0.86	the greater the amplitude the greater the amount of energy in the wave
0.85	the deeper the level of processing the stronger the trace and the better the memory
0.84	the darker the blue the deeper the water
CUE	**these four quantum numbers are used in describing electron behavior**
0.353	these numbers are important in chemistry
0.3433	these are usually used in the home
0.316	what punctuation marks are used in these four sentences
0.315	these are abbreviations that sometimes are used in written directions

Table 5 illustrates the application of the same approach at a different level of granurality to facilitate the retrieval of similar sentences. The set of superposed semantic and orthographic vectors used to generate Table 4, as well as randomly generated elemental vectors for high frequency terms that were not represented in this set, were combined with sets of demarcator vectors derived from different $D(\alpha)$ and $D(\omega)$ vectors than those used at the character level. This allows for the encoding of phrases and sentences. For example, the phrase "these numbers" would be encoded as: $S(\text{these}) \otimes D(1) + S(\text{numbers}) \otimes D(2)$. When applied to sentences from the TASA corpus, this facilitates the retrieval of sentences describing entities related to one another in similar ways to those in the cue sentence, a facility that has proved useful for the automated extraction of propositional information from narrative text [26].

6 Conclusion

In this paper we have introduced a novel approach through which the near-orthogonality of elemental vectors is deliberately violated to introduce measured similarity into semantic space. While illustrated primarily through orthographic modeling, the approach is general in nature and can be applied in any situation in which a representation of sequence that is tolerant to variation is desired. Furthermore, this approach may mediate the generation of holistic representations combining distributional and spatial information, a direction we plan to explore in future work. To facilitate further experimentation, our real, binary and complex orthographic vector implementations have been released as components of the open source Semantic Vectors package [27, 28].

Acknowledgments: This research was supported by US National Library of Medicine grant R21 LM010826. We would like to thank Lance DeVine, for the CHRR implementation used in this research, and Tom Landauer for providing the TASA corpus.

References

1. M. N. Jones and D. J. K. Mewhort, "Representing word meaning and order information in a composite holographic lexicon," *Psychological Review*, vol. 114, pp. 1–37, 2007.
2. M. Sahlgren, A. Holst, and P. Kanerva, "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 D.C., USA.*, 2008.
3. G. E. Cox, G. Kachergis, G. Recchia, and M. N. Jones, "Toward a scalable holographic word-form representation," *Behavior Research Methods*, vol. 43, pp. 602–615, Sept. 2011.
4. G. Kachergis, G. Cox, and M. Jones, "OrBEAGLE: integrating orthography into a holographic model of the lexicon," *Artificial Neural Networks and Machine Learning–ICANN 2011*, pp. 307–314, 2011.
5. T. A. Plate, *Holographic Reduced Representation: Distributed Representation for Cognitive Structures*. Stanfpord, CA.: CSLI Publications, 2003.
6. T. Cohen and D. Widdows, "Empirical distributional semantics: methods and biomedical applications," *Journal of Biomedical Informatics*, vol. 42, pp. 390–405, Apr. 2009.
7. P. D. Turney and P. Pantel, "From frequency to meaning: Vector space models of semantics," *Journal of Artificial Intelligence Research*, vol. 37, no. 1, pp. 141–188, 2010.
8. T. Landauer and S. Dumais, "A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge," *Psych. Review*, vol. 104, pp. 211–240, 1997.
9. C. Burgess, K. Livesay, and K. Lund, "Explorations in context space: Words, sentences, discourse," *Discourse Processes*, vol. 25, pp. 211–257, 1998.

10. L. De Vine and P. Bruza, “Semantic oscillations: Encoding context and structure in complex valued holographic vectors,” *Proc AAAI Fall Symp on Quantum Informatics for Cognitive, Social, and Semantic Processes*, 2010.
11. P. Basile, A. Caputo, and G. Semeraro, “Encoding syntactic dependencies by vector permutation,” in *Proceedings of the EMNLP 2011 Workshop on GEometrical Models of Natural Language Semantics, GEMS*, vol. 11, pp. 43–51, 2011.
12. T. Cohen, D. Widdows, R. Schvaneveldt, and T. Rindflesch, “Finding schizophrenia’s prozac: Emergent relational similarity in predication space,” in *Proc 5th International Symposium on Quantum Interactions. Aberdeen, Scotland. Springer-Verlag Berlin, Heidelberg.*, 2011.
13. P. Gomez, R. Ratcliff, and M. Perea, “The Overlap Model: A Model of Letter Position Coding,” *Psychological review*, vol. 115, pp. 577–600, July 2008.
14. C. J. Davis, “The spatial coding model of visual word identification.,” *Psychological Review*, vol. 117, no. 3, p. 713, 2010.
15. T. Hannagan, E. Dupoux, and A. Christophe, “Holographic string encoding,” *Cognitive science*, vol. 35, no. 1, pp. 79–118, 2011.
16. 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.
17. P. Kanerva, “Binary spatter-coding of ordered k-tuples,” *Artificial Neural Networks—ICANN 96*, pp. 869–873, 1996.
18. M. Wahle, D. Widdows, J. R. Herskovic, E. V. Bernstam, and T. Cohen, “Deterministic Binary Vectors for Efficient Automated Indexing of MEDLINE/PubMed Abstracts,” *AMIA Annual Symposium Proceedings*, vol. 2012, pp. 940–949, Nov. 2012.
19. D. Widdows and S. Peters, “Word vectors and quantum logic experiments with negation and disjunction,” in *Proc 8th Math. of Language Conference.*, (Bloomington, Indiana.), 2003.
20. T. Cohen, D. Widdows, L. D. Vine, R. Schvaneveldt, and T. C. Rindflesch, “Many Paths Lead to Discovery: Analogical Retrieval of Cancer Therapies,” in *Quantum Interaction* (J. R. Busemeyer, F. Dubois, A. Lambert-Mogiliansky, and M. Melucci, eds.), no. 7620 in Lecture Notes in Computer Science, pp. 90–101, Springer Berlin Heidelberg, Jan. 2012.
21. D. Bohm, *Quantum Theory*. Prentice-Hall, 1951. Republished by Dover, 1989.
22. D. Widdows and T. Cohen, “Real, complex, and binary semantic vectors,” in *Sixth International Symposium on Quantum Interaction*, (Paris, France), 2012.
23. T. Hannagan and J. Grainger, “Protein Analysis Meets Visual Word Recognition: A Case for String Kernels in the Brain,” *Cognitive Science*, vol. 36, no. 4, pp. 575–606, 2012.
24. D. Yarowsky and R. Wicentowski, “Minimally supervised morphological analysis by multimodal alignment,” in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL ’00, (Stroudsburg, PA, USA), pp. 207–216, Association for Computational Linguistics, 2000.
25. M. Baroni, J. Matiasek, and H. Trost, “Unsupervised discovery of morphologically related words based on orthographic and semantic similarity,” in *Proceedings of the ACL-02 workshop on Morphological and phonological learning - Volume 6*, MPL ’02, (Stroudsburg, PA, USA), pp. 48–57, Association for Computational Linguistics, 2002.
26. S. Dennis, “Introducing word order within the LSA framework,” *Handbook of Latent Semantic Analysis*, 2007.
27. D. Widdows and K. Ferraro, “Semantic vectors: A scalable open source package and online technology management application,” *Sixth International Conference on Language Resources and Evaluation (LREC 2008).*, 2008.
28. D. Widdows and T. Cohen, “The semantic vectors package: New algorithms and public tools for distributional semantics,” in *Fourth IEEE International Conference on Semantic Computing (ICSC)*, pp. 9–15, 2010.