Should Semantic Vector Composition be Explicit? Can it be Linear?

Dominic Widdows
LivePerson Inc.
dwiddows@liveperson.com

Kristen Howell
LivePerson Inc.
khowell@liveperson.com

Trevor Cohen
University of Washington
cohenta@uw.edu

Abstract

Vector representations have become a central element in semantic language modelling, leading to mathematical overlaps with many fields including quantum theory. Compositionality is a core goal for such representations: given representations for 'wet' and 'fish', how should the concept 'wet fish' be represented?

This position paper surveys this question from two points of view. The first considers the question of whether an explicit mathematical representation can be successful using only tools from within linear algebra, or whether other mathematical tools are needed. The second considers whether semantic vector composition should be explicitly described mathematically, or whether it can be a model-internal side-effect of training a neural network.

A third and newer question is whether a compositional model can be implemented on a quantum computer. Given the fundamentally linear nature of quantum mechanics, we propose that these questions are related, and that this survey may help to highlight candidate operations for future quantum implementation.

1 Introduction

Semantic composition has been noted and studied since ancient times, including questions on which parts of language should be considered atomic, how these are combined to make new meanings, and how explicitly the process of combination can be modelled.[1]

As vectors have come to play a central role in semantic representation, these questions have naturally become asked of semantic vector models. Early examples include the weighted summation of term vectors into document vectors in information retrieval (Salton et al. 1975) and the modelling of variable-value bindings using the tensor product[2] in artificial intelligence (Smolensky 1990). The use of vectors in the context of natural language processing grew from such roots, landmark papers including the introduction of Latent Semantic Analyis (Deerwester et al. 1990), where the vectors are created using singular value decomposition, and Word Embeddings (Mikolov et al. 2013), where the vectors are created by training a neural net to predict masked tokens.

During the 20th century, logical semantics was also developed, based very much upon the discrete mathematical logic tradition of Boole (1854) and Frege (1884) rather than the continuous vector spaces that developed from the works of Hamilton (1847) and Grassmann (1862). Thus, by the beginning of this century, compositional semantics was developed mathematically, provided frameworks such as Montague semantics for connecting the truth value of a sentence with its syntactic form, but provided little insight on how the atomic parts themselves should be represented. Good examples in this tradition can be found in Gamut (1991) and Partee et al. (1993). In summary, by the year 2000, there were distributional language models with vectors, symbolic models with composition, but little in the way of distributional vector models with composition.

[1] Early examples are given by Aristotle, such as (*De Interpretatione, Ch IV):*

> *The word 'human' has meaning, but does not constitute a proposition, either positive or negative. It is only when other words are added that the whole will form an affirmation or denial.*

[2] Familiarity with tensor products is assumed throughout this paper. Readers familiar with the linear algebra of vectors but not the multilinear algebra of tensors are encouraged to read the introduction to tensors in Widdows et al. (2021, §5).

Proceedings of the 2021 Workshop on Semantic Spaces at the Intersection of NLP, Physics, and Cognitive Science, pages 76–86

2 Explicit Composition in Semantic Vector Models

The most standard operations used for composition and comparison in vector model information retrieval systems have, for many decades, been the vector sum and cosine similarity (see Salton et al. 1975), and for an introduction, Widdows (2004, Ch 5)). Cosine similarity is normally defined and calculated in terms of the natural scalar product induced by the coordinate system, i.e.,

$$\cos(a, b) = \frac{a \cdot b}{\sqrt{(a \cdot a)(b \cdot b)}}.$$

While the scalar product is a linear operator because $\lambda a \cdot \mu b = \lambda\mu(a \cdot b)$, cosine similarity is deliberately designed so that $cos(\lambda a, \mu b) = \cos(a, b)$, so that normalizing or otherwise reweighting vectors does not affect their similarity, which depends only on the angle between them.

More sophisticated vector composition in AI was introduced with cognitive models and connectionism. The work of Smolensky (1990) has already been mentioned, and the holographic reduced representations of Plate (2003) is another widely-cited influence (discussed again below as part of Vector Symbolic Architectures). While Smolensky's work is often considered to be AI rather than NLP, the application to language was a key consideration:

> Any connectionist model of natural language processing must cope with the questions of how linguistic structures are represented in connectionist models. (Smolensky 1990, §1.2)

The use of more varied mathematical operators to model natural language operations with vectors accelerated considerably during the first decade of this century. In information retrieval, van Rijsbergen (2004) explored conditional implication and Widdows (2003) developed the use of orthogonal projection for negation in vector models.

Motivated partly by formal similarities with quantum theory, Aerts & Czachor (2004) proposed modelling a sentence $w_1, \ldots, w_n$ with a tensor product $w_1 \otimes \ldots \otimes w_n$ in the Fock space $\bigoplus_{k=1}^{\infty} V^{\otimes^k}$. The authors noted that comparing the spaces $V^{\otimes^k}$ and $V^{\otimes^j}$ when $k \neq j$ is a difficulty shared by other frameworks, and of course the prospect of summing to $k = \infty$ is a mathematical notation that motivates the search for a tractable implementation proposal.

By the middle of the decade, Clark & Pulman (2007) and Widdows (2008) proposed and experimented with the use of tensor products for semantic composition, and the parallelogram rule for relation extraction. Such methods were used to obtain strong empirical results for (intransitive) verb-noun composition by Mitchell & Lapata (2008) and for adjective-noun composition by Baroni & Zamparelli (2010). One culmination of this line of research is the survey by Baroni et al. (2014), who also addressed the problem of comparing tensors $V^{\otimes^k}$ and $V^{\otimes^j}$ when $k \neq j$. For example, if (as in Baroni & Zamparelli 2010)), nouns are represented by vectors and adjectives are represented by matrices, then the space of matrices is isomorphic to $V \otimes V$ which is not naturally comparable to V, and the authors note (Baroni & Zamparelli 2010, §3.4):

> As a result, Rome and Roman, Italy and Italian cannot be declared similar, which is counter-intuitive. Even more counter-intuitively, Roman used as an adjective would not be comparable to Roman used as a noun. We think that the best way to solve such apparent paradoxes is to look, on a case-by-case basis, at the linguistic structures involved, and to exploit them to develop specific solutions.

Another approach is to use a full syntactic parse of a sentence to construct vectors in a sentence space S from nouns and verbs as constituents in their respective spaces. This features prominently in the model of Coecke et al. (2010), which has become affectionately known as DisCoCat, from '*Dis*tributional *Co*mpositional *Cat*egorical'. The mathematics is at the same time sophisticated but intuitive: its formal structure relies on pregroup grammars and morphisms between compact closed categories, and intuitively, the information from semantic word vectors flows through a network of tensor products that parallels the syntactic bindings and produces a single vector in the S space.

Various papers have demonstrated empirical successes for the DisCoCat and related models. Grefenstette & Sadrzadeh (2011) were among the first, showing comparable and sometimes improved results with those of Mitchell & Lapata (2008). By 2014, several tensor composition operations were compared by Milajevs et al. (2014), and Sadrzadeh et al. (2018) showed that word, phrase, and sentence entailment can be modelled using vectors

and density matrices. (The use of density matrices to model probability distributions for entailment was pioneered partly by van Rijsbergen (2004, p. 80) and will be discussed further in Section 4.) Further mathematical tools used in DisCoCat research include *copying* a vector $v \in V$ into a tensor in $V \otimes V$ where the coordinates of v become the diagonal entries in the matrix representation of the corresponding tensor, and *uncopying* which takes the diagonal elements of a matrix representing a tensor in $V \otimes V$ and uses these as the coordinates of a vector. With these additional operators, the tensor algebra becomes more explicitly a *Frobenius algebra*.[3] These are used in DisCoCat models by Sadrzadeh et al. (2014*a*,*b*) to represent relative and then possessive pronoun attachments (for example, representing the affect of the phrase "that chased the mouse" as part of the phrase "The cat that chased the mouse"). The method involves detailed tracking of syntactic types and their bindings, and certainly follows the suggestion from Baroni & Zamparelli (2010) to look at linguistic structures on a case-by-case basis.

There are practical concerns with tensor product formalisms. The lack of any isomorphism between $V^{\otimes^k}$ and $V^{\otimes^j}$ when $k \neq j$ and $\dim V > 1$ has already been noted, along with the difficulty this poses for comparing elements of each for similarity. Also, there is an obvious computational scaling problem: if V has dimension n, then $V^{\otimes^k}$ has dimension n^k, which leads to exponential memory consumption with classical memory registers. Taking the example of relative pronouns in the DisCoCat models of Sadrzadeh et al. (2014*a*) — these are represented as rank-4 tensors in spaces such as $N \otimes S \otimes N \otimes N$ and variants thereof, and if the basic noun space N and sentence space S have dimension 300 (a relatively standard number used e.g., by FastText vectors) then the relative pronouns would have dimension 8.1 billion. If this was represented densely and the coordinates are 4-byte floating point numbers, then just representing one pronoun would require over 30GB of memory, which is intractable even by today's cloud computing standards.

The development of Vector Symbolic Architectures (VSAs) (Gayler 2004) was partly motivated by these concerns. VSAs grew from the holographic reduced representations of Plate (2003): notable works in this intersection of cognitive science and artificial intelligence include those of Eliasmith (2013) and Kanerva (2009). At its core, a VSA is a vector space with an addition operator and a scalar product for computing similarity, along with a multiplication or *binding* operator (sometimes written as $*$, or $\otimes$ like the tensor product) which takes a product of two vectors and returns a new vector that it typically *not* similar to either of its inputs, so that $(a * b) \cdot a$ is small, but which is 'approximately reversible' — so there is an approximate inverse operator $\oslash$ where $(a * b) \oslash b$ is close to a.[4] The term 'binding' was used partly for continuity with the role-filler binding of Smolensky (1990).[5]

The VSA community has tended to avoid the full tensor product, for the reasons given above. In order to be directly comparable, it is desirable that $a * b$ should be a vector in the space V. Plate (2003) thoroughly explored the use of *circular correlation* and *circular convolution* for these operations, which involves summing the elements of the outer product matrix along diagonals. This works as a method to map $V \otimes V$ back to V, though the mapping is of course basis dependent. Partly to optimize the binding operation to $O(n)$ time, Plate (2003, Ch 5) introduces *circular vectors*, whose coordinates are unit complex numbers. There is some stretching of terminology here, because the circle group $U(1)$ of unit complex numbers is not, strictly speaking, a vector space. Circular vectors are added by adding their rectangular coordinates in a linear fashion, and then normalizing back to the unit circle by discarding the magnitude, which Plate notes is not an associative operation. Kanerva (2009) departs perhaps even further from the vector space mainstream, using binary-valued vectors throughout, with binding implemented as pairwise exclusive OR (XOR).

VSA binding operations have been used for composition of semantic vector representations, both during the training process to generate composite vector representations of term or character n-grams,

[3] Named after Georg Frobenius (1849–1917), a group theorist who contributed particularly to group representation theory. See Kartsaklis (2015) for a thorough presentation.

[4] This also explains why it is tempting to reuse or abuse the tensor product notation and use the symbol $\otimes$ for binding and $\oslash$ for the inverse or release operator, as in Widdows & Cohen (2015).

[5] The requirement that the binding $a*b$ be dissimilar to both a and b makes the Frobenius uncopying of Kartsaklis (2015) operator unsuitable for a VSA, because the coordinates of $v*w$ are the products of the corresponding coordinates in v and w, which typically makes the scalar product with either factor quite large. This is however a rather shallow observation, and the relationship between VSAs and Frobenius algebras may be a fruitful topic to investigate more thoroughly.

or semantic units such as predicate-argument pairs or syntactic dependencies, that are then further assembled to construct representations of larger units (Jones & Mewhort 2007, Kachergis et al. 2011, Cohen & Widdows 2017, Paullada et al. 2020); and to compose larger units from pretrained semantic vectors for downstream machine learning tasks (Fishbein & Eliasmith 2008, Mower et al. 2016). However, a concern with several of the standard VSA binding operators for the representation of sequences in particular is that they are commutative in nature: $x * y = y * x$. To address this concern, permutations of vector coordinates have been applied across a range of VSA implementations to break the commutative property of the binding operator, for example by permuting the second vector in sequence such that $\overrightarrow{wet} * \prod(\overrightarrow{fish})$ and $\overrightarrow{fish} * \prod(\overrightarrow{wet})$ result in different vectors (Kanerva 2009, Plate 2003, p. 121).

Thanks to their general nature and computational simplicity, permutations have been used for several other encoding and composition experiments. The use of permutations to encode positional information into word vector representations was introduced by Sahlgren et al. (2008). In this work a permutation (coordinate shuffling) operator was used to rearrange vector components during the course of training, with a different random permutation assigned to each sliding window position such that a context vector would be encoded differently depending upon its position relative to a focus term of interest. A subsequent evaluation of this method showed advantages in performance over the BEAGLE model (Jones & Mewhort 2007), which uses circular convolutions to compose representations of word n-grams, on a range of intrinsic evaluation tasks — however these advantages were primarily attributable to the permutation-based approach's ability to scale to a larger training corpus (Recchia et al. 2015). Random permutations have also been used to encode semantic relations (Cohen et al. 2009) and syntactic dependencies (Basile et al. 2011) into distributional models.

In high-dimensional space, the application of two different random permutations to the same vector has a high probability of producing vectors that are close-to-orthogonal to one another (Sahlgren et al. 2008). A more recent development has involved deliberately constructing 'graded' permutations by randomly permuting part of a parent permutation (Cohen & Widdows 2018). When this process is repeated iteratively, it results in a set of permutations that when applied to the same vector will produce a result with similarity to the parent vector that decreases in an ordinal fashion. This permits the encoding of proximity rather than position, in such a way that words in proximal positions within a sliding window will be similarly but not identically encoded. The resulting proximity-based encodings have shown advantages over comparable encodings that are based on absolute position (at word and sentence level) or are position-agnostic (at word and character level) across a range of evaluations (Cohen & Widdows 2018, Schubert et al. 2020, Kelly et al. 2020).

Note that coordinate permutations are all Euclidean transformations: odd permutations are reflections, and even permutations are rotations. Thus all permutation operations are also linear.

This survey of explicit composition in semantic vector models is not exhaustive, but gives some idea of the range of linear and nonlinear operations.

3 Compositional Semantics in Neural Networks

During the past decade, many of the most successful and well-known advances in semantic vector representations have been developed using neural networks.[6] In general, such networks are trained with some objective function designed to maximize the probability of predicting a given word, sentence, or group of characters in a given context. Various related results such as those of Scarselli & Tsoi (1998) are known to demonstrate that, given enough computational resources and training data, neural networks can be used to approximate any example from large classes of functions. If these target functions are nonlinear, this cannot be done with a network of entirely linear operations, because the composition of two linear maps is another linear map — "The hidden units should be nonlinear because multiple layers of linear units can only produce linear functions." (Wichert 2020, §13.5). Thus, part of the appeal of neural networks is that they are *not* bound by linearity: though often at considerable computational cost.

The skip gram with negative sampling method was introduced by Mikolov et al. (2013), imple-

[6] An introduction to this huge topic is beyond the scope of this paper. Unfamiliar readers are encouraged to start with a general survey such as that of Géron (2019), Chapter 16 being particularly relevant to the discussion here.

mentations including the word2vec[7] package from Google and the FastText package from Facebook.[8] The objective function is analyzed more thoroughly by Goldberg & Levy (2014), and takes the form:

$$\sum_{(w,c)\in D} \log \sigma(\overrightarrow{w} \cdot \overrightarrow{c}) + \sum_{(w,\neg c)\in D'} \log \sigma(-\overrightarrow{w} \cdot \overrightarrow{\neg c})$$

Here w is a word, c is a context feature (such as a nearby word), D represents observed term/context pairs in the document collection, D' represents randomly drawn counterexamples, and $\overrightarrow{w}$ and $\overrightarrow{c}$ are word and context vectors (input and output weights of the network, respectively). σ is the sigmoid function, $\frac{1}{1+e^{-x}}$. The mathematical structure here is in the family of logistic and softmax functions — the interaction between the word and context vectors involves exponential / logarithmic concepts, not just linear operations.

There have been efforts to incorporate syntactic information explicitly in the training process of neural network models. In the specific case of adjectives, Maillard & Clark (2015) use the skip gram technique to create matrices for adjectives following the pattern of Baroni & Zamparelli (2010) discussed in Section 2. The most recent culmination of this work is its adaptation to cover a much more comprehensive collection of categorial types by Wijnholds et al. (2020). Another early example comes from Socher et al. 2012, who train a Recursive Neural Network where each node in a syntactic parse tree becomes represented by a matrix that operates on a pair of inputs. Research on tree-structured LSTMs (see inter alia Tai et al. 2015, Maillard et al. 2019) leverages syntactic parse trees in the input and composes its hidden state using an arbitrary number of child nodes, as represented in the syntax tree. Syntax-BERT (Bai et al. 2021) uses syntactic parses to generate masks that reflect different aspects of tree structure (parent, child, sibling). KERMIT (Zanzotto et al. 2020) uses compositional structure explicitly by embedding syntactic subtrees in the representation space. In both cases, the use of explicit compositional syntactic structure leads to a boost in performance on various semantic tasks.

In KERMIT, the embedding of trees and (recursively) their subtrees follows a well-developed line of research on representing discrete structures as vectors, in particular combining circular convolution and permutations to introduce *shuffled circular convolution* (Ferrone & Zanzotto 2014). Even when combined in a recursive sum over constituents called a Distributed Tree Kernel operation, this is still a sum of linear inputs, so this form of composition is still linear throughout. In such methods, the result may be a collection of related linear operators representing explicit syntactic bindings, but the training method is typically not linear due to the activation functions.

What these neural net methods and the models described in the previous section have in common is that they encode some *explicit* compositional structure: a weighted sum of word or character n-grams, a role / value binding, or a relationship in a grammatical parse tree. This raises the question: can neural language models go beyond the bag-of-words drawbacks and encode more order-dependent language structures without using traditional logical compositional machinery?

A recent and comprehensive survey of this topic is provided by Hupkes et al. (2020). This work provides a valuable survey of the field to date, and conducts experiments with compositional behavior on artificial datasets designed to demonstrate various aspects of compositionality, such as productivity (can larger unseen sequences be produced?) and substitutivity (are outputs the same when synonymous tokens are switched?). This systematic approach to breaking compositionality into many tasks is a useful guide in itself.

Since then, attention-based networks were developed and have come to the forefront of the field (Vaswani et al. 2017). The attention mechanism is designed to learn when pairs of inputs depend crucially on one another, a capability that has demonstrably improved machine translation by making sure that the translated output represents all of the given input even when their word-orders do not correspond exactly. The 'scaled dot-product attention' used by Vaswani et al. (2017) for computing the attention between a pair of constituents uses softmax normalization, another nonlinear operation.

The use of attention mechanisms has led to rapid advances in the field, including the contextualized BERT (Devlin et al. 2018) and ELMo (Peters et al. 2018) models. For example, the ELMo model reports good results on traditional NLP tasks including question answering, coreference resolution, semantic role labelling, and part-of-speech tagging,

[7] https://pypi.org/project/word2vec/
[8] https://fasttext.cc/

and the authors speculate that this success is due to the model's different neural-network layers implicitly representing several different kinds of linguistic structure. This idea is further investigated by Hewitt & Manning (2019) and Jawahar et al. (2019), who probe BERT and ELMo models to find evidence that syntactic structure is implicitly encoded in their vector representations. The survey and experiments of Hupkes et al. (2020) evaluate three such neural networks on a range of tasks related to composition, concluding that each network has strengths and weaknesses, that the results are a stepping stone rather than an endpoint, and that developing consensus around how such tasks should be designed, tested and shared is a crucial task in itself.

At the time of writing, such systems are contributors to answering a very open research question: do neural networks need extra linguistic information in their input to properly work with language, or can they actually *recover* such information as a byproduct of training on raw text input? For example, a DisCoCat model requires parsed sentences as input — so if another system performed as well without requiring grammatical sentences as input and the support of a distinct parsing component in the implementation pipeline, that would be preferable in most production applications. (Running a parser is a requirement than today can often be satisfied, albeit with an implementational and computational cost. Requiring users to type only grammatical input is a requirement that cannot typically be met at all.) At the same time, does performance on the current NLP tasks used for evaluation directly indicate semantic composition at play? If the performance of a model without linguistic information in the input is up to par, would the internal operations of such an implicit model be largely inscrutable, or can we describe the way meaningful units are composed into larger meaningful structures explicitly?

Tensor networks are one of the possible mathematical answers to this question, and continue to build upon Smolensky's introduction of tensors to AI. For example McCoy et al. (2020) present evidence that the sequence-composition effects of Recurrent Neural Networks (RNNs) can be approximated by Tensor Product Decomposition Networks, at least in cases where using this structure provides measurable benefits over bag-of-words models. It has also been shown that Tensor Product Networks can be used to construct an attention mechanism from which grammatical structure can be recovered by unbinding role-filler tensor compositions (Huang et al. 2019).

While there are many more networks that could be examined in a survey like this, those described in this section illustrate that neural networks have been used to improve results with many NLP tasks, and the training of such networks often crucially depends on nonlinear operations on vectors. Furthermore, while tensor networks have been developed as a proposed family of techniques for understanding and exploiting compositional structures more explicitly, in some of the most state-of-the-art models, relationships between such operations to more traditional semantic composition are often absent or at least not well-understood.

4 Operators from Quantum Models

Mathematical correspondences between vector models for semantics and quantum theory have been recognized for some years (van Rijsbergen 2004), and are surveyed by Widdows et al. (2021). The advent of practical quantum computing makes these correspondences especially interesting, and constructs from quantum theory have been used increasingly deliberately in NLP. In quantum computing, tensor products no longer incur quadratic costs: instead, the tensor product $A \otimes B$ is the natural mathematical representation of the physical state that arises when systems in states A and B are allowed to interact. Heightened interest in quantum computing and quantum structures in general has led to specific semantic contributions already.

Mathematically, there is a historic relationship between linearity and quantum mechanics: the principle of superposition guarantees that for any state A, the vector cA corresponds to the same physical state for any complex number c (Dirac 1930, §5).[9] Hence the question of whether a compositional operator is linear or not is particularly relevant when we consider the practicality of implementation on a quantum computer.[10]

Many developments have followed from the Dis-

[9] This itself could lead to a mathematical discussion — the magnitude of a state vector in quantum mechanics is ignored, just like cosine similarity ignores the scale factors of the scalar product, and its resilience to scale factors makes the cosine similarity technically *not* a linear operator.

[10] The dependence of quantum computing on linearity should not go unquestioned — for example, the use of quantum harmonic oscillators rather than qubits has been proposed as a way to incorporate nonlinearity into quantum hardware by Goto (2016).

CoCat framework, whose mathematical structure is closely related to quantum mechanics through category theory (Coecke et al. 2017). As of 2021, the tensor network components of two DisCoCat models have even been implemented successfully on a quantum computer (Lorenz et al. 2021), and there are proposals for how to implement the syntactic parse on quantum hardware as well (Wiebe et al. 2019, Bausch et al. 2021). Of particular semantic and mathematical interest, topics such as hyponymy (Bankova et al. 2019) and negation (Lewis 2020) have been investigated, using density matrices and positive operator-valued measures, which are mathematical generalizations of state vectors and projection operators that enable the theory to describe systems that are not in 'pure' states. Density matrices have also been used to model sentence entailment (Sadrzadeh et al. 2018) and recently lexical ambiguity (Meyer & Lewis 2020).

A comprehensive presentation of the use of density matrices to model joint probability distributions is given by Bradley (2020). This work deliberately takes a quantum probability framework and applies it to language modelling, by way of the lattice structures of Formal Concept Analysis (Ganter & Wille 1999). This work uses the *partial trace* of density operators (which are tensors in $V \otimes V$) to project tensors in $V \otimes V$ to vectors in V. This is analogous to summing the rows or columns of a two-variable joint distribution to get a single-variable marginal distribution. This captures interference and overlap between the initial concepts, and in a framework such as DisCoCat, this might be used to model transitive verb-noun composition (as in Grefenstette & Sadrzadeh 2011, Sadrzadeh et al. 2018, and others).

Another mathematical development is the quantum Procrustes alignment method of Lloyd et al. (2020), where Procrustes alignment refers to the challenge of mapping one vector space into another preserving relationships as closely as possible. Procrustes techniques have been used to align multilingual FastText word vectors (Joulin et al. 2018), and it is possible that one day these methods may be combined to give faster and more noise-tolerant multilingual concept alignment.

This again is not a complete survey, but we hope it demonstrates that the interplay between quantum theory, semantic vector composition, and practical implementation has much to offer, and that work in this area is accelerating.

5 Summary, Conclusion, and Future Work

This paper has surveyed vector composition techniques used for aspects of semantic composition in explicit linguistic models, neural networks, and quantum models, while acknowledging that these areas overlap. The operations considered are gathered and summarized in Table 1.

Some of the most successful neural network models to date have used operations that are nonlinear and implicit. Though models such as BERT and ELMo have performed exceptionally well on several benchmark tasks, they are famously difficult to explain and computationally expensive. Therefore, scientific researchers and commercial user-facing enterprises have good reason to be impressed, but still to seek alternatives that are clearer and cheaper. At the same time, progress in quantum computing raises the possibility that the practical cost of different mathematical operations may be considerably revised over the coming year. For example, if the expense of tensor products becomes linear rather than quadratic, tensor networks may find a position at the forefront of 'neural quantum computing'.

In addition, there is emerging evidence that such models can be augmented by approaches that draw on structured semantic knowledge (Michalopoulos et al. 2020, Colon-Hernandez et al. 2021), suggesting the combination of implicit and explicit approaches to semantic composition as a fruitful area for future methodological research. We hope that this approach of surveying and comparing the semantic, mathematical and computational elements of various vector operations will serve as a guide to territory yet to be explored at the intersection of compositional operators and vector representations of language.

6 Acknowledgements

The authors would like to thank the anonymous reviewers for helping to improve the coverage of this survey, and the conference organizers for allowing extra pages to accommodate these additions.

References

Aerts, D. & Czachor, M. (2004), 'Quantum aspects of semantic analysis and symbolic artificial intelligence', *J. Phys. A: Math. Gen.* **37**, L123–L132.

Bai, J., Wang, Y., Chen, Y., Yang, Y., Bai, J., Yu, J. & Tong, Y. (2021), 'Syntax-BERT: Improving

Table 1: Survey Summary of Mathematical Methods Used for Semantic Vector Composition

Mathematical Method	**Use Case**	**Inputs**	**Outputs**	**Explicitness**	**Linearity**	**Described By**
Vector sum	Document retrieval and many others	Word vectors	Document vectors	Explicit	Linear	Widespread including Salton et al. (1975)
Scalar product	Similarity scoring	Any vector	Any vector	Explicit	Linear	Various
Cosine similarity	Similarity scoring	Any vector	Any vector	Explicit	Nonlinear — deliberately ignores magnitude	Various
Tensor product	Role-filler binding	Variable name	Variable value	Explicit	Linear	Smolensky (1990)
Circular vector sum	Holographic reduced representations	Circular vectors	Circular vectors	Explicit	Nonlinear, and non-associative	Plate (2003)
(Stalnaker) Conditional	Implication	Vector / subspace representing propositions	Subspace representing truth conditions	Explicit	Linear (though for subspaces it doesn't matter)	van Rijsbergen (2004, Ch 5)
Orthogonal projection	Negation	Vector or subspace	Vector	Explicit	Linear	Widdows (2003)
Sum of subspaces	Disjunction	Vectors or subspaces	Subspace	Explicit	Linear	Widdows (2003)
Parallelogram rule	Proportional analogy	Three vectors	Fourth vector	Explicit	Linear	Various incl. Widdows (2008), Mikolov et al. (2013)
Tensor product	Word vector composition	Word vector	Sentence-fragment tensor	Explicit	Linear	Various since Aerts & Czachor (2004), Clark & Pulman (2007)
Tensor and monoidal product	Parallel semantic, syntactic composition	(vector, syntactic type) pairs	Sentence vectors	Explicit	Linear	Various since Coecke et al. (2010)
Matrix multiplication	Adjective / noun composition	Matrix and vector	Vector	Explicit	Linear	Baroni & Zamparelli (2010)
Circular convolution	Vector binding	VSA vector	VSA vector	Explicit	Sometimes	Plate (2003), options in Widdows & Cohen (2015)
Binary XOR	Binary vector binding	VSA vector	VSA vector	Explicit	Binary vectors warrant more discussion!	Kanerva (2009)
Permutation of coordinates	Non-additive composition	Vector	Vector	Explicit, though often random	Linear (because rotation or reflection)	Sahlgren et al. (2008) and various
Skipgram objective	Vector interation in training	Word and context vector	Update to both	Explicit though internal	Nonlinear	Mikolov et al. (2013)
$\tanh$, Sigmoid, ReLU, Softmax, etc.	Activation functions in neural networks	Input weights	Output weights	Typically implicit	Nonlinear	Many including Géron (2019, Ch 10)
Scaled dot-product attention	Learning pairwise dependence	Vectors	Updated vectors	Typically internal	Nonlinear	Vaswani et al. (2017)
Distributed tree kernel / shuffled circular convolution	Embedding syntactic tree in vector space	Parse tree	Sentence vector	Explicit	Linear	(Ferrone & Zanzotto 2014)
Density matrices, POVMs	More general distributions over vector spaces, e.g., representing categories, implication	Several, e.g., superpositions of pairs of vectors	Projected vectors and / or probabilities	Often explicit	Linear	van Rijsbergen (2004), Sadrzadeh et al. (2018), Lewis (2020), Bradley (2020)
Procrustes alignment	Aligning vector models U and V	Pairs of source, target vectors	Linear mapping from U to V	Explicit	Linear	Bojanowski et al. (2017), Lloyd et al. (2020)

pre-trained transformers with syntax trees', *arXiv preprint arXiv:2103.04350* .

Bankova, D., Coecke, B., Lewis, M. & Marsden, D. (2019), 'Graded hyponymy for compositional distributional semantics', *Journal of Language Modelling* **6**(2), 225–260.

Baroni, M., Bernardi, R., Zamparelli, R. et al. (2014), 'Frege in space: A program for compositional distributional semantics', *Linguistic Issues in language technology* **9**(6), 5–110.

Baroni, M. & Zamparelli, R. (2010), Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space, *in* 'Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)'.

Basile, P., Caputo, A. & Semeraro, G. (2011), Encoding syntactic dependencies by vector permutation, *in* 'Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics', pp. 43–51.

Bausch, J., Subramanian, S. & Piddock, S. (2021), 'A quantum search decoder for natural language processing', *Quantum Machine Intelligence* **3**(1), 1–24.

Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. (2017), 'Enriching word vectors with subword information', *Transactions of the Association for Computational Linguistics* **5**, 135–146.

Boole, G. (1854), *An Investigation of the Laws of Thought*, Macmillan. Dover edition, 1958.

Bradley, T.-D. (2020), 'At the interface of algebra and statistics', *PhD dissertation, arXiv preprint arXiv:2004.05631* .

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

Coecke, B., Genovese, F., Gogioso, S., Marsden, D. & Piedeleu, R. (2017), 'Uniqueness of composition in quantum theory and linguistics', *14th International Conference on Quantum Physics and Logic (QPL)* .

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

Cohen, T., Schvaneveldt, R. W. & Rindflesch, T. C. (2009), Predication-based semantic indexing: Permutations as a means to encode predications in semantic space, *in* 'AMIA Annual Symposium Proceedings', Vol. 2009, American Medical Informatics Association, p. 114.

Cohen, T. & Widdows, D. (2017), 'Embedding of semantic predications', *Journal of biomedical informatics* **68**, 150–166.

Cohen, T. & Widdows, D. (2018), Bringing order to neural word embeddings with embeddings augmented by random permutations (earp), *in* 'Proceedings of the 22nd Conference on Computational Natural Language Learning', pp. 465–475.

Colon-Hernandez, P., Havasi, C., Alonso, J., Huggins, M. & Breazeal, C. (2021), 'Combining pre-trained language models and structured knowledge', *arXiv preprint arXiv:2101.12294* .

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.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2018), 'BERT: Pre-training of deep bidirectional transformers for language understanding', *arXiv preprint arXiv:1810.04805* .

Dirac, P. (1930), *The Principles of Quantum Mechanics*, 4th edition, 1958, reprinted 1982 edn, Clarendon Press, Oxford.

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

Ferrone, L. & Zanzotto, F. (2014), Towards syntax-aware compositional distributional semantic models, *in* 'COLING 2014, 25th International Conference on Computational Linguistics'.

Fishbein, J. M. & Eliasmith, C. (2008), Integrating structure and meaning: A new method for encoding structure for text classification, *in* 'European Conference on Information Retrieval', Springer, pp. 514–521.

Frege, G. (1884), *The Foundations of Arithmetic (1884)*, 1974 (translated by J. L. Austin) edn, Blackwell.

Gamut, L. (1991), *Logic, Language, and Meaning*, University of Chicago Press.

Ganter, B. & Wille, R. (1999), *Formal Concept Analysis: Mathematical Foundations*, Springer.

Gayler, R. W. (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.

Géron, A. (2019), *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly Media.

Goldberg, Y. & Levy, O. (2014), 'Word2Vec explained: deriving Mikolov et al.'s negative-sampling word-embedding method', *arXiv preprint arXiv:1402.3722* .

Goto, H. (2016), 'Bifurcation-based adiabatic quantum computation with a nonlinear oscillator network', *Scientific reports* **6**(1), 1–8.

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

Grefenstette, E. & Sadrzadeh, M. (2011), 'Experimental support for a categorical compositional distributional model of meaning', *EMNLP* .

Hamilton, S. W. R. (1847), 'On quaternions', *Proc. Royal Irish Acad.* **3**, 1–16.

Hewitt, J. & Manning, C. D. (2019), A structural probe for finding syntax in word representations, *in* 'ACL', pp. 4129–4138.

Huang, Q., Deng, L., Wu, D., Liu, C. & He, X. (2019), Attentive tensor product learning, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence', Vol. 33, pp. 1344–1351.

Hupkes, D., Dankers, V., Mul, M. & Bruni, E. (2020), 'Compositionality decomposed: how do neural networks generalise?', *Journal of Artificial Intelligence Research* **67**, 757–795.

Jawahar, G., Sagot, B. & Seddah, D. (2019), What does bert learn about the structure of language?, *in* 'ACL 2019-57th Annual Meeting of the Association for Computational Linguistics'.

Jones, M. N. & Mewhort, D. J. K. (2007), 'Representing word meaning and order information in a composite holographic lexicon', *Psychological Review* **114**, 1–37.

Joulin, A., Bojanowski, P., Mikolov, T., Jégou, H. & Grave, E. (2018), Loss in translation: Learning bilingual word mapping with a retrieval criterion, *in* 'EMNLP'.

Kachergis, G., Cox, G. E. & Jones, M. N. (2011), Orbeagle: integrating orthography into a holographic model of the lexicon, *in* 'International conference on artificial neural networks', Springer, pp. 307–314.

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

Kartsaklis, D. (2015), 'Compositional distributional semantics with compact closed categories and frobenius algebras', *PhD thesis, Wolfson College Oxford. arXiv preprint arXiv:1505.00138* .

Kelly, M. A., Xu, Y., Calvillo, J. & Reitter, D. (2020), 'Which sentence embeddings and which layers encode syntactic structure?'.

Lewis, M. (2020), 'Towards logical negation for compositional distributional semantics', *arXiv preprint arXiv:2005.04929* .

Lloyd, S., Bosch, S., De Palma, G., Kiani, B., Liu, Z.-W., Marvian, M., Rebentrost, P. & Arvidsson-Shukur, D. M. (2020), 'Quantum polar decomposition algorithm', *arXiv preprint arXiv:2006.00841* .

Lorenz, R., Pearson, A., Meichanetzidis, K., Kartsaklis, D. & Coecke, B. (2021), 'QNLP in practice: Running compositional models of meaning on a quantum computer'.

Maillard, J. & Clark, S. (2015), Learning adjective meanings with a tensor-based skip-gram model, *in* 'Proceedings of the Nineteenth Conference on Computational Natural Language Learning', pp. 327–331.

Maillard, J., Clark, S. & Yogatama, D. (2019), 'Jointly learning sentence embeddings and syntax with unsupervised tree-lstms', *Natural Language Engineering* **25**(4), 433–449.

McCoy, R. T., Linzen, T., Dunbar, E. & Smolensky, P. (2020), 'Tensor product decomposition networks: Uncovering representations of structure learned by neural networks', *Proceedings of the Society for Computation in Linguistics* **3**(1), 474–475.

Meyer, F. & Lewis, M. (2020), 'Modelling lexical ambiguity with density matrices', *arXiv preprint arXiv:2010.05670* .

Michalopoulos, G., Wang, Y., Kaka, H., Chen, H. & Wong, A. (2020), 'Umlsbert: Clinical domain knowledge augmentation of contextual embeddings using the unified medical language system metathesaurus', *arXiv preprint arXiv:2010.10391* .

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013), 'Efficient estimation of word representations in vector space', *arXiv preprint arXiv:1301.3781* .

Milajevs, D., Kartsaklis, D., Sadrzadeh, M. & Purver, M. (2014), Evaluating neural word representations in tensor-based compositional settings, *in* 'EMNLP', pp. 708–719.

Mitchell, J. & Lapata, M. (2008), Vector-based models of semantic composition., *in* 'ACL', pp. 236–244.

Mower, J., Subramanian, D., Shang, N. & Cohen, T. (2016), Classification-by-analogy: using vector representations of implicit relationships to identify plausibly causal drug/side-effect relationships, *in* 'AMIA annual symposium proceedings', Vol. 2016, American Medical Informatics Association, p. 1940.

Partee, B. H., ter Meulen, A. & Wall, R. E. (1993), *Mathematical Methods in Linguistics*, Kluwer.

Paullada, A., Percha, B. & Cohn, T. (2020), Improving biomedical analogical retrieval with embedding of structural dependencies, *in* 'Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing', pp. 38–48.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. & Zettlemoyer, L. (2018), 'Deep contextualized word representations', *arXiv preprint arXiv:1802.05365* .

Plate, T. A. (2003), *Holographic Reduced Representations: Distributed Representation for Cognitive Structures*, CSLI Publications.

Recchia, G., Sahlgren, M., Kanerva, P. & Jones, M. N. (2015), 'Encoding sequential information in semantic space models: comparing holographic reduced representation and random permutation.', *Computational intelligence and neuroscience* .

Sadrzadeh, M., Clark, S. & Coecke, B. (2014*a*), 'The frobenius anatomy of word meanings ii: possessive relative pronouns', *Journal of Logic and Computation* **26**(2), 785–815.

Sadrzadeh, M., Clark, S. & Coecke, B. (2014*b*), 'The frobenius anatomy of word meanings ii: possessive relative pronouns', *Journal of Logic and Computation* **26**(2), 785–815.

Sadrzadeh, M., Kartsaklis, D. & Balkır, E. (2018), 'Sentence entailment in compositional distributional semantics', *Annals of Mathematics and Artificial Intelligence* **82**(4), 189–218.

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

Salton, G., Wong, A. & Yang, C.-S. (1975), 'A vector space model for automatic indexing', *Communications of the ACM* **18**(11), 613–620.

Scarselli, F. & Tsoi, A. C. (1998), 'Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results', *Neural networks* **11**(1), 15–37.

Schubert, T. M., Cohen, T. & Fischer-Baum, S. (2020), 'Reading the written language environment: Learning orthographic structure from statistical regularities', *Journal of Memory and Language* **114**, 104148.

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, C. D. & Ng, A. Y. (2012), Semantic compositionality through recursive matrix-vector spaces, *in* 'EMNLP', pp. 1201–1211.

Tai, K. S., Socher, R. & Manning, C. D. (2015), 'Improved semantic representations from tree-structured long short-term memory networks', *arXiv preprint arXiv:1503.00075* .

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

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017), Attention is all you need, *in* 'Advances in neural information processing systems', pp. 5998–6008.

Wichert, A. (2020), *Principles of quantum artificial intelligence: Quantum Problem Solving and Machine Learning (Second Edition)*, World scientific.

Widdows, D. (2003), Orthogonal negation in vector spaces for modelling word-meanings and document retrieval, *in* 'ACL 2003', 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. (2015), 'Reasoning with vectors: a continuous model for fast robust inference', *Logic Journal of IGPL* **23**(2), 141–173.

Widdows, D., Kitto, K. & Cohen, T. (2021), 'Quantum mathematics in artificial intelligence', *arXiv preprint arXiv:2101.04255* .

Wiebe, N., Bocharov, A., Smolensky, P., Troyer, M. & Svore, K. M. (2019), 'Quantum language processing', *arXiv preprint arXiv:1902.05162* .

Wijnholds, G., Sadrzadeh, M. & Clark, S. (2020), Representation learning for type-driven composition, *in* 'Proceedings of the 24th Conference on Computational Natural Language Learning', pp. 313–324.

Zanzotto, F. M., Santilli, A., Ranaldi, L., Onorati, D., Tommasino, P. & Fallucchi, F. (2020), KERMIT: Complementing transformer architectures with encoders of explicit syntactic interpretations, *in* 'EMNLP', pp. 256–267.