Cats Climb Entails Mammals Move: Preserving Hyponymy in Compositional Distributional Semantics

Gemma De las Cuevas[1], Andreas Klingler[1], Martha Lewis[2], and Tim Netzer[3]

[1]*Institute for Theoretical Physics, Technikerstr. 21a, A-6020 Innsbruck, Austria*
[2]*Department of Engineering Mathematics, University of Bristol, Bristol, U.K.*
[3]*Department of Mathematics, Technikerstr. 13, A-6020 Innsbruck, Austria*
gemma.delascuevas@uibk.ac.at, andreas.klingler@uibk.ac.at, martha.lewis@bristol.ac.uk, tim.netzer@uibk.ac.at

Abstract

To give vector-based representations of meaning more structure, an approach proposed in Piedeleu et al. (2015); Sadrzadeh et al. (2018); Bankova et al. (2018) is to use positive semidefinite (psd) matrices. These allow us to model similarity of words as well as the *hyponymy* or *is-a* relationship. To compose words to form phrases and sentences, we may represent adjectives, verbs, and other functional words as multilinear, positivity preserving maps, following the compositional distributional approach introduced in Coecke et al. (2010) and extended to the realm of psd matrices in Piedeleu et al. (2015), but it is not clear how to learn representations of functional words when working with psd matrices. In this paper, we introduce a generic way of composing the psd matrices corresponding to words. We propose that psd matrices

Received 8th March 2021; Revised 31st August 2021; Accepted 9th September 2021
Journal of Cognitive Science 22(3): 311-353 September 2021

for verbs, adjectives, and other functional words be lifted to completely positive (CP) maps that match their grammatical type. This lifting is carried out by our composition rule called Compression, Compr. In contrast to previous composition rules like Fuzz and Phaser (Coecke and Meichanetzidis, 2020) (a.k.a. KMult and BMult (Lewis, 2019a)), Compr preserves hyponymy. Mathematically, Compr is itself a CP map, and is therefore linear and generally non-commutative. We give a number of proposals for the structure of Compr, based on spiders, cups, and caps, and generate a range of composition rules. We test these rules on sentence entailment datasets from Kartsaklis and Sadrzadeh (2016), and see some improvements over the performance of Fuzz and Phaser. We go on to estimate the parameters of a simplified form of Compr based on entailment information from the aforementioned datasets, and find that whilst this learnt operator does not consistently outperform previously proposed mechanisms, it is competitive and has the potential to improve with the use of a less simplified version.

Keywords: *hyponymy, matrix-based representations of meaning, positive semidefinite matrices, completely positive maps*

1. Introduction

With similarity measured by the inner product of normalised word vectors, vector-based representations of words have been extremely successful in many applications, such as translation, thesaurus generation, and paraphrasing. However, as well as similarity, there are a number of other important relationships between words or concepts, one of these being *hyponymy* or the *is-a* relation. Examples of this are that *cat* is a hyponym of *mammal*, but we can also apply this to verbs, and say that *sprint* is a hyponym of *run*. This relationship is important in modelling human concept use, as it links to the important notion of categorisation - if we see a big cat we know that it is probably dangerous, whether it is a jaguar or a leopard (Murphy, 2003). It is also important in its role in inference - from 'Some dogs bark' we may infer 'some animals bark' due to the hyponymy relation holding between dogs and animals (Lyons, 1968; Gagné et al., 2020). As such, modelling hyponymy is important for natural language processing tasks that are related to categorisation and inference. These tasks form an important and thriving area of research, and a number of challenges and large datasets have been devel-

oped in the area, for example the PASCAL Recognising Textual Entailment Challenges (Dagan et al., 2005), the FraCaS test suite (Cooper et al., 1994), the Stanford Natural Language Inference dataset (Bowman et al., 2015b) and the extended MultiNLI (Williams et al., 2018).

Within vector-based semantics based on co-occurrence statistics, there have been several alternative approaches to building word vectors that can represent hyponymy and entailment relationships. A fruitful approach in distributional semantics is based on a principle known as the *distributional inclusion hypothesis* or DIH. This hypothesizes that given two words w_1 and w_2, where w_1 is a hyponym, i.e. more specific than w_2, the more general word w_2 will be seen in all the contexts that the more specific word w_1 is seen in, and more. This approach was developed and extended in a number of works, including Geffet and Dagan (2005); Weeds et al. (2004); Kotlerman et al. (2010); Lenci and Benotto (2012), in which directed measures of similarity are developed. Related to the distributional inclusion hypothesis is the entropic approach of Herbelot and Ganesalingam (2013), who examine the Kullback-Leibler divergence between the probability distributions encoded by two distributional word vectors. More recent examples of alternative ways of building word vectors include using non-Euclidean geometries (Nickel and Kiela, 2017; Nguyen et al., 2017; Le et al., 2019), or specialising word vectors for entailment (Vulić and Mrkšić, 2018). Other approaches include defining a vector lattice for word vectors (Clarke, 2009), looking at properties that are *not* shared between words (Rimell, 2014), or looking at the dispersion of a word representations (Kiela et al., 2015).

If we wish to use the hyponymy relationships between words to provide information about entailment in sentences in which they are used, it is necessary to compose the words into sentences in some way. Within vector-based models of meaning, there are a number of approaches to composing words to form phrases and sentences. A key approach in this field is that of Mitchell and Lapata (2010). In that paper, the authors propose that word vectors can be combined by means of a multilinear map to form a combined representation. For example, suppose that we have vectors $v, w \in \mathbb{R}^n$ representing two words, and a multilinear map $C : \mathbb{R}^n \otimes \mathbb{R}^n \rightarrow \mathbb{R}^m$, then the sentence vector is represented as $s = C(v, w)$. At the time, learning the parameters of such a map was considered too difficult, and so element-wise combinations were proposed, such as pointwise multiplication or addition

of the two vectors.

Another key approach to forming phrases and sentences is via hierarchical neural network structures such as in Socher et al. (2013); Bowman et al. (2015a), or by learning sentence representations directly, as in Conneau et al. (2017); Kiros et al. (2015); Reimers et al. (2019). These approaches use large neural network structures to generate sentence vectors. The approaches in Socher et al. (2013); Bowman et al. (2015a) could be seen as extensions of the Mitchell and Lapata (2010) approach, in that the aim in these papers is to learn a composition function, which when provided with vector inputs and composition order, will output a sentence vector.

A third approach to forming phrases and sentences can be summarised as tensor-based composition. This approach was introduced in Coecke et al. (2010) and independently in Baroni and Zamparelli (2010); Paperno et al. (2014). The approach introduced in Coecke et al. (2010) is often termed DisCoCat, standing for Categorical Compositional Distributional semantics, and detail on this approach is given in section 2. Briefly, the idea is that nouns and sentences are modelled as vectors, and functional words such as adjectives and verbs are modelled as multilinear maps. In this paper we will be working within the DisCoCat framework. There are various different methods proposed for learning the word representations needed in DisCoCat. In early work, nouns were usually modelled as count-based vectors, in which the parameters of the vectors were set based on their distribution in a large corpus. To define the parameters of the matrices and tensors needed for functional words, an approach pioneered in Grefenstette and Sadrzadeh (2011); Kartsaklis et al. (2016, 2012) is to build a word representation based on the arguments of the functional word and then lift this representation to the correct space using *Frobenius algebras* - in the case of adjectives, for example, this would take a vector-based representation of an adjective and lift it to a matrix by putting the vector along the diagonal of the matrix. Another means of building the required tensors is via regression learning. Here, holistic vectors for e.g. 'red car', 'red brick', 'red house' etc. are learnt and a matrix for 'red' estimated via linear regression over the vectors for 'car', 'brick', 'house', and so on (Grefenstette et al., 2013). This approach was extended and refined in Kartsaklis et al. (2014) where word representations were improved using a step of prior disambiguation. In Wijnholds et al. (2020) the use of regression techniques was combined

with neural methods based on word2vec that allow the learning of word matrices and tensors automatically. In the present paper, our approach has most similarity with the early methods of lifting word representations to the correct vector space, however we also trial a small scale regression method.

Whilst the DisCoCat framework is successful in composing words to form phrases and sentences, it does not have an immediately obvious way of representing hyponymy and entailment. This problem was successfully addressed in Kartsaklis and Sadrzadeh (2016), which proposes a tensor-based variant of the distributional inclusion hypothesis. In that paper the authors propose combination methods for noun and verb vectors that are motivated by the distributional inclusion hypothesis. They show how features can be expected to be preserved under the differing combination mechanisms, and how this relates to the distributional inclusion hypothesis. They test their word representations and combination methods using a wide range of different metrics, and find that a metric developed by them in earlier work gives the highest figures. The work presented in the current paper addresses the same task, and we use the dataset developed in Kartsaklis and Sadrzadeh (2016) to assess performance.

An extension to the standard DisCoCat model was developed in Piedeleu et al. (2015), in which positive semidefinite (psd) matrices are used to represent words. The authors showed that psd matrices can be used within the same compositional framework as DisCoCat, and can be argued to provide more structure than vectors. In particular, psd matrices can be used to model ambiguity in word meanings. As well as providing the theoretical foundations for using psd matrices within DisCoCat, Piedeleu et al. (2015) built matrices for words by mixing together the vectors for different senses of the word. The use of density matrices for lexical ambiguity was further explored in Meyer and Lewis (2020) where a method for building density matrices for words on a larger scale was proposed.

A second way in which psd matrices provide more structure is in modelling hyponymy. In Sadrzadeh et al. (2018) the use of psd matrices to model hyponymy between words and sentences is proposed. In this work, a hyponymy relationship between two words is modelled as a function of the KL-divergence between the two matrices of the words. This measure has the nice property that hyponymy relationships between individual words lift to an entailment relationship between the two sentences. In similar work,

Bankova et al. (2018) model the relationship of hyponymy as the Löwner order between two matrices. In the particular case of projectors, the Löwner ordering can be seen as subspace inclusion – i.e., for projectors P and Q representing subspaces S_P and S_Q of a given space V, $P \leqslant Q$ iff S_P is a subspace of S_Q. Applied to psd matrices in general, if we have two words w_A and w_B represented by matrices A and B, we would like $A \leqslant B$ whenever w_A is a hyponym of w_B. Bankova et al. (2018) provide a measure of graded hyponymy between two matrices which again lifts to entailment at the sentence level.

The graded hyponymy measure proposed by Bankova et al. (2018) and the KL-divergence related measure proposed in Sadrzadeh et al. (2018) have the limitation that to form the comparison $A \leq B$, the subspace spanned by A must be a subset of the subspace spanned by B, which does not always hold. To remedy this problem, Lewis (2019a) proposed two more measures which do not have this limitation. These will be described in section 5.

A drawback of using psd matrices for word representations is that learning psd matrices from text is difficult, mainly the larger matrices required for functional words. Therefore, in Lewis (2019a); Coecke and Meichanetzidis (2020), composition rules for psd matrices have been explored that lift psd matrices into a higher dimensional space. This approach is very similar to that proposed in Grefenstette and Sadrzadeh (2011); Kartsaklis et al. (2012), where word representations for intransitive verbs are lifted from a space W to $W \otimes W$, and transitive verbs are lifted from $W \otimes W$ to $W \otimes W \otimes W$. We will discuss these composition rules in more detail in subsequent sections. In Coecke and Meichanetzidis (2020) the composition rules are called Fuzz and Phaser, in Lewis (2019a) they are KMult and BMult, respectively. For this paper, we stick with the guitar pedal terminology.

In Sadrzadeh et al. (2018); Bankova et al. (2018), the composition methods used had the desirable property that hyponymy relations at the word level lifted to the sentence level. Take two sentences of the same length with the same grammatical types in the same order, where each word in the first sentence is a hyponym of the corresponding word in the first. Then, for the right kind of entailment relation, the hyponymy relations together with the composition methods used would result in an entailment relation holding between the two resulting sentences. However, this property is not shared by Fuzz and Phaser. That is, given two pairs of words in a hyponym-hypernym

relationship, the combination of the two hyponyms is not necessarily a hyponym of the combination of the two hypernyms:

$$noun_1 \leqslant noun_2 \quad \text{and} \quad verb_1 \leqslant verb_2$$

does not imply

$$noun_1 * verb_1 \leqslant noun_2 * verb_2$$

where the nouns and verbs are psd matrices, $\leqslant$ is the Löwner ordering, and $*$ is one of Fuzz or Phaser.

This paper aims to define a composition rule which is (i) positivity preserving, and (ii) hyponymy preserving. In addition, we will require it to be bilinear. If possible, it should also be non-commutative, since commutative rules would make no difference between phrases like 'dog bites Alice' and 'Alice bites dog', which do not have the same meaning. Our composition rule is called Compression, Compr, and it is, in fact, an infinite set of rules; namely all completely positive maps from $\mathcal{M}_m$ to $\mathcal{M}_m \otimes \mathcal{M}_m$, where $\mathcal{M}_m$ denotes the set of real matrices of size $m \times m$. As a special case, we recover Mult.

We use the following notation. A^* denotes complex conjugate transpose, and A_* means complex conjugate. PSD_m denotes the set of positive semidefinite (psd) matrices of size $m \times m$ over the real numbers, and a psd element is denoted by $\geqslant 0$. We use the term *functional words* for verbs and adjectives that take arguments. Nouns are not functional words.

2. Representing grammar and meaning

We work within the categorical compositional distributional (DisCoCat) model of meaning introduced in introduced in Coecke et al. (2010), and expanded on in Preller and Sadrzadeh (2011); Sadrzadeh et al. (2013). This is a model for unifying statistical models of word meaning such as distributional semantics together with formal semantic approaches to meaning. DisCoCat uses a category-theoretic stance. It models syntax in one category, call it the *grammar category*, and semantics in another, call it the *meaning category*. A functor from the grammar category to the meaning category is defined, so that the grammatical reductions on the syntactic side can be translated into morphisms on the meaning side. As in Coecke et al. (2010) and much

subsequent research, to model grammar, we will use pregroup grammar, as described below, although there are also other choices of grammar (Coecke et al., 2013; Maillard et al., 2014; Wijnholds, 2015; Muskens and Sadrzadeh, 2016; Moortgat and Wijnholds, 2017; Moortgat et al., 2020). As first introduced by Piedeleu et al. (2015), and subsequently applied by Sadrzadeh et al. (2018); Bankova et al. (2018), to model word meanings, we will use the category of **CPM**(**FHilb**) of Hilbert spaces and completely positive maps between them, using the aforementioned Löwner order to model hyponymy.

2.1 Pregroup grammar

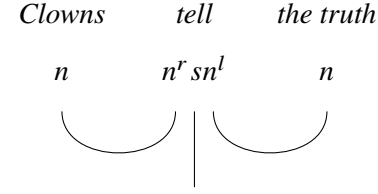
Pregroup grammar is built over a set of types. We consider the set containing n for noun and s for sentence. Each type has adjoints x^r and x^l. Complex types are built up by concatenation of types, and we often leave out the dot so that $xy = x \cdot y$. There is a unit type such that $x1 = 1x = x$. Types and their adjoints interact via:

$$\epsilon^r_x : x \cdot x^r \leq 1, \qquad \epsilon^l_x : x^l \cdot x \leq 1 \qquad \eta^r_x : 1 \leq x^r \cdot x, \qquad \eta^l_x : 1 \leq x \cdot x^l \quad (1)$$

A string of grammatical types $t_1, ...t_n$ is grammatical if it reduces, via the morphisms above, to the sentence type s. For example, typing *clowns* as n, *tell* as $n^r s n^l$ and *the truth* as n, the sentence *Clowns tell the truth* has type $n(n^r s n^l)n$ and is shown to be grammatical as follows:

$$(\epsilon^r\ 1\ \epsilon^l)\ (n\ (n^r\ s\ n^l)\ n) \leq (\epsilon^r 1)(n\ n^r\ s\ 1) \leq 1\ s\ 1 = s \quad (2)$$

The above reduction can be represented graphically as follows:



2.2 Words as positive semidefinite matrices

The application of DisCoCat to psd matrices was first proposed in Piedeleu et al. (2015). We give here an outline of the theory introduced in that paper.

DisCoCat comes with an intuitive diagrammatic calculus which we will make a little use of in later sections, but we refrain from outlining the calculus here for space reasons. For an introduction to the diagrammatic calculus as applied to linguistics please see Coecke et al. (2010) and for its specific use with psd matrices please see Piedeleu et al. (2015). Positive semidefinite matrices are represented in $\mathbf{CPM}(\mathbf{FHilb})$ as morphisms $\mathbb{R} \to M \otimes M^*$, where M is some finite-dimensional Hilbert space and M^* is its dual. A functor S from the category representing pregroup grammar to $\mathbf{CPM}(\mathbf{FHilb})$ sends nouns and sentences to psd matrices, and adjectives, verbs, and other functional words to completely positive maps, or equivalently psd matrices in a larger space. Words are represented as psd matrices in the following way. In the vector-based model of meaning, a word w is represented by a column vector, $|w\rangle \in \mathbb{R}^m$ (for some m). To pass to psd matrices, a subset of words S will be mapped to rank 1 matrices, i.e. $|w\rangle \mapsto |w\rangle \langle w|$. The words in S are the hyponyms. The other words, which are hypernyms of the words in S, will be represented as mixtures of hyponyms:

$$\rho = \sum_{w \in W \subset S} |w\rangle \langle w| \tag{3}$$

Within a compositional model of meaning, we view nouns as psd matrices in $\mathcal{M}_m$, so that $\mathsf{S}(n) = \mathcal{M}_m$ and sentences as psd matrices in $\mathcal{M}_s$ so that $\mathsf{S}(s) = \mathcal{M}_s$ (for some m and s). An intransitive verb has type $n^r s$ in the pregroup grammar, and is mapped by S to a psd element in $\mathcal{M}_m \otimes \mathcal{M}_s$. Equivalently, an intransitive verb is a completely positive (CP) map $\mathcal{M}_m \to \mathcal{M}_s$. The method for building psd matrices summarised in equation (3) maps words of all grammatical types to psd matrices in $\mathcal{M}_m$. This is the correct type for nouns, but wrong for other grammatical types. Taking the example of intransitive verbs, we need to find a mechanism to lift an intransitive verb as a psd element in $\mathcal{M}_m$ to a CP map $\mathcal{M}_m \to \mathcal{M}_s$. There have been various approaches to implementing this type lifting, which we now summarise.

Suppose n is a psd matrix for a noun, and v a psd matrix for a verb. Proposals in Lewis (2019a); Coecke (2019); Coecke and Meichanetzidis (2020) include the following – note in particular that Fuzz and Phaser defined in Coecke and Meichanetzidis (2020) coincide with KMult and BMult defined in Lewis (2019a):

- $\mathsf{Mult}(n, v) = n \odot v$ where $\odot$ is the Hadamard product, i.e. $(n \odot v)_{i,j} =$

$n_{ij}v_{ij}$.

- $\mathsf{Fuzz}(n, v) = \mathsf{KMult}(n, v) = \sum_i p_i P_i n P_i$ where $v = \sum_i p_i P_i$ is the spectral decomposition of v. That is,

$$\mathsf{Fuzz}(n, v) = \sum_i \sqrt{p_i} P_i n P_i \sqrt{p_i}$$

- $\mathsf{Phaser}(n, v) = \mathsf{BMult}(n, v) = v^{1/2} n v^{1/2}$. Let $v = \sum_i p_i P_i$ be the spectral decomposition of v. Then

$$\mathsf{Phaser}(n, v) = \sum_{i,j} \sqrt{p_i} P_i n P_j \sqrt{p_j}$$

Some benefits and drawbacks of these operations are as follows. Mult is a straightforward use of Frobenius algebras in the category $\mathbf{CPM}(\mathbf{FHilb})$. It was originally introduced in Piedeleu et al. (2015) and can be seen as an extension of the models used by Grefenstette and Sadrzadeh (2011) and Kartsaklis et al. (2012) to lift verb vectors to the status of a linear map. It is linear, completely positive and preserves hyponymy. However, linguistically it is unsatisfactory because it is commutative. So it will map 'Howard likes Jimmy' to the same psd matrix as 'Jimmy likes Howard' – which do not have the same meaning and so should not have the same matrix representation. On the other hand, both Phaser and Fuzz are non-commutative, however, they are not linear and do not preserve hyponymy. In the next section, we outline the properties we want from a composition method, and propose a general framework that will allow us to generate a number of suggestions.

3. In search of more guitar pedals: Compression

For the rest of the paper, we assume that both nouns and verbs are represented by psd matrices of the same size m, that is, we let $n \in \mathrm{PSD}_m$ and $v \in \mathrm{PSD}_m$. We are looking for a composition rule for these psd matrices. We call the desired operation Compr (for reasons we shall later see), and want it to be a map

$$\mathsf{Compr} : \mathcal{M}_m \times \mathcal{M}_m \to \mathcal{M}_m.$$

The minimal two properties required from this map are the following:

(i) *Positivity preserving*

If n, v are psd, then $\mathsf{Compr}(n, v)$ is psd:

$$\mathsf{Compr} : \mathrm{PSD}_m \times \mathrm{PSD}_m \to \mathrm{PSD}_m$$

(ii) *Hyponymy preserving*

If hyponymy is represented by the Löwner order $\leqslant$,[1]

$$n_1 \leqslant n_2, \quad v_1 \leqslant v_2 \implies \mathsf{Compr}(n_1, v_1) \leqslant \mathsf{Compr}(n_2, v_2)$$

Although these two properties are the most important ones, we now consider another property:

(iii) *Bilinearity*

Compr is linear in each of its arguments, namely for $\alpha \in \mathbb{R}$:

$\mathsf{Compr}(\alpha n, v) = \alpha \mathsf{Compr}(n, v)$

$\mathsf{Compr}(n, \alpha v) = \alpha \mathsf{Compr}(n, v)$

$\mathsf{Compr}(n + n', v) = \mathsf{Compr}(n, v) + \mathsf{Compr}(n', v)$

$\mathsf{Compr}(n, v + v') = \mathsf{Compr}(n, v) + \mathsf{Compr}(n, v')$

Assumption (iii) has two advantages. The first one is that if the map is positivity preserving on the Cartesian product [(i)] and bilinear [(iii)], then it is hyponymy preserving [(ii)]:

Lemma 1. *Assumptions (i) and (iii) imply (ii).*

Proof. Assume that $n_2 \geqslant n_1$ and $v_2 \geqslant 0$. Using (i) we have that $\mathsf{Compr}(n_2 - n_1, v_2) \geqslant 0$, and using (iii) that $\mathsf{Compr}(n_2, v_2) \geqslant \mathsf{Compr}(n_1, v_2)$. Now assume that $v_2 \geqslant v_1$ and $n_1 \geqslant 0$. Following the same argument we obtain that $\mathsf{Compr}(n_1, v_2) \geqslant \mathsf{Compr}(n_1, v_1)$. By transitivity of being psd, we obtain that $\mathsf{Compr}(n_2, v_2) \geqslant \mathsf{Compr}(n_1, v_1)$, which is condition (ii). □

[1] If ρ, σ are psd, then $\rho \leqslant \sigma$ iff $\sigma - \rho \geqslant 0$, i.e. if $\sigma - \rho$ is itself psd.

The second advantage of bilinearity is that it allows reformulating Compr in a convenient way. Since Compr is linear in both components, we construct the following linear map:

$$\begin{aligned} \mathcal{M}_m &\to \mathrm{Lin}(\mathcal{M}_m, \mathcal{M}_m) \\ v &\mapsto (\mathsf{Compr}(\cdot, v) \colon n \mapsto \mathsf{Compr}(n, v)) \end{aligned}$$

Note that linearity of Compr in the noun component is necessary for the image of this map to be $\mathrm{Lin}(\mathcal{M}_m, \mathcal{M}_m)$. In contrast, linearity in the verb component is necessary for this map itself to be linear. By slight abuse of notation we denote this new map also by Compr:

$$\mathsf{Compr} \colon \mathcal{M}_m \to \mathrm{Lin}(\mathcal{M}_m, \mathcal{M}_m).$$

Now, assumption (i) applied to this new map means that psd matrices are mapped to positivity preserving maps,

$$\mathsf{Compr} : \mathrm{PSD}_m \to \mathrm{PP}(\mathcal{M}_m, \mathcal{M}_m),$$

where $\mathrm{PP}(\mathcal{M}_m, \mathcal{M}_m)$ is the set of positivity preserving linear maps from $\mathcal{M}_m$ to $\mathcal{M}_m$, i.e. those that map psd matrices to psd matrices. To make things more tractable, one can use the isomorphism

$$\begin{aligned} \mathrm{Lin}(\mathcal{M}_m, \mathcal{M}_m) &\to \mathcal{M}_m \otimes \mathcal{M}_m \\ \varphi &\mapsto \sum_{i,j} \varphi(|e_i\rangle\langle e_j|) \otimes |e_i\rangle\langle e_j|, \end{aligned}$$

where $\{|e_i\rangle\}$ is an orthonormal basis of $\mathbb{R}^m$. Using this isomorphism, $\mathrm{PP}(\mathcal{M}_m, \mathcal{M}_m)$ corresponds to the set of block positive matrices $\mathrm{BP}(\mathcal{M}_m \otimes \mathcal{M}_m)$ on the tensor product space.[2] Summarizing, we are trying to construct a linear map

$$\mathsf{Compr} \colon \mathcal{M}_m \to \mathcal{M}_m \otimes \mathcal{M}_m$$

that maps psd matrices to block positive matrices. So far, this is just a reformulation of conditions (i) and (iii).

[2] A matrix $\rho \in \mathcal{M}_m \otimes \mathcal{M}_m$ is block positive if $(\langle v| \otimes \langle w|)\rho(|v\rangle \otimes |w\rangle) \geq 0$ for all vectors $|v\rangle$, $|w\rangle$.

To make things more tractable, we now further strengthen the conditions on the map. Namely, we require Compr to map psd matrices in $\mathcal{M}_m$ to psd matrices in $\mathcal{M}_m \otimes \mathcal{M}_m \cong \mathcal{M}_{m^2}$, i.e. positivity preserving itself. Using the isomorphism above, this means that Compr maps psd matrices to completely positive (CP) maps from $\mathcal{M}_m$ to $\mathcal{M}_m$ (see table 1):

$$\mathsf{Compr} : \mathrm{PSD}_m \to \mathrm{CP}(\mathcal{M}_m, \mathcal{M}_m).$$

Table 1. Correspondence between linear maps $\mathcal{M}_m \to \mathcal{M}_m$ and elements in $\mathcal{M}_m \otimes \mathcal{M}_m$, known as the Choi-Jamiołkowski isomorphism.

Linear map $\mathcal{M}_m \to \mathcal{M}_m$	$\leftrightarrow$	Element in $\mathcal{M}_m \otimes \mathcal{M}_m$
Positivity preserving map	$\leftrightarrow$	Block positive matrix
Completely positive map	$\leftrightarrow$	Positive semidefinite matrix

And since we are still not running out of steam, we require Compr not only to be positivity preserving but also to be completely positive itself. In total, we are trying to construct a *completely positive map*

$$\begin{aligned} \mathsf{Compr}\colon \mathcal{M}_m &\to \mathcal{M}_m \otimes \mathcal{M}_m \cong \mathcal{M}_{m^2} \\ v &\mapsto \sum_l K_l v K_l^* \end{aligned} \tag{4}$$

where we have used the well known fact that every completely positive map admits a Kraus decomposition, for certain Kraus operators $K_l \in \mathbb{R}^m \otimes \mathcal{M}_m \cong \mathcal{M}_{m^2,m}$. The proof that such decomposition exists can be found in Nielsen and Chuang (2010), Theorems 8.1 and 8.3.

In summary, we are asking for a stronger condition than just (i) and (iii). On the other hand, the weaker forms of maps mentioned above do not admit a closed description, whereas completely positive maps do. By Stinespring's Dilation Theorem, all completely positive maps can be expressed as a $*$-representation followed by a *compression* (Paulsen, 2002). This is the reason for denoting this operation Compr and is precisely what gives rise to the Kraus decomposition of the map. This also fits well with the electric guitar pedal notation from Coecke and Meichanetzidis (2020), as can be seen in figure 1.

Note that in general, Compr is non-commutative, i.e. when translated back to the initial setup of

$$\mathsf{Compr} : \mathcal{M}_m \times \mathcal{M}_m \to \mathcal{M}_m$$

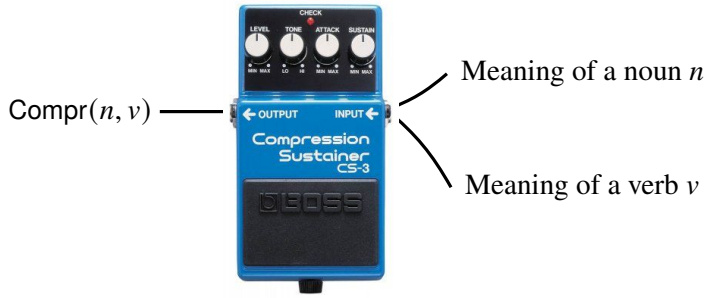


Figure 1. Not only Fuzz and Phaser, but also Compression is a valuable guitar pedal. The operation Compr takes as input an element of $\mathcal{M}_m \times \mathcal{M}_m$ (denoted n, v) representing the meaning of a noun and the meaning of a verb, and it outputs Compr(n, v), representing the meaning of the sentence $n\ v$.

we will generally have Compr$(n, v) \neq$ Compr(v, n). This is a suitable property, reflecting that a word's position in a sentence has both syntactic and semantic roles. Note also that neither Fuzz nor Phaser (nor KMult nor BMult) are linear in the verb component, i.e. they do not fulfill (iii), and are thus not special cases of Compr. However, Mult is a particular case of Compr, as we shall see.

4. Building nouns and verbs in CPM(FHilb)

In order to build a psd matrix for a given word w, we require information about its hyponyms $\{h_i\}_i$ and vectors $\{|h_i\rangle\}_i$ representing each of the $\{h_i\}_i$. We can then form the matrix

$$\rho(w) = \sum_i |h_i\rangle \langle h_i| \in W \otimes W^*.$$

where W^* is the dual of the column vector space W. $\rho(w)$ is then normalised. In this work, we normalise using the infinity norm, that is, we divide by the maximum eigenvalue. This has been shown to have nice properties (van de Wetering, 2017).

As proposed in Lewis (2019a), the set of hyponyms $\{h_i\}_i$ may, for example, be gathered from WordNet (Miller, 1995) or in a less supervised manner by extracting hyponym-hypernym pairs using Hearst patterns (Hearst, 1992;

Roller et al., 2018). Vectors representing each hyponym can be taken from pretrained word vectors such as word2vec (Mikolov et al., 2013) or GloVe (Pennington et al., 2014).

This approach to representing word meanings puts all representations in the shared space $W \otimes W^*$. If we are working in the category **CPM**(**FHilb**), this is the right kind of representation for nouns, but not for functional words. To transform a psd matrix $\rho(verb) \in \mathrm{PSD}_m$ to a psd matrix in $\mathcal{M}_{m^2}$, we use the composition rule Compr proposed in section 3:

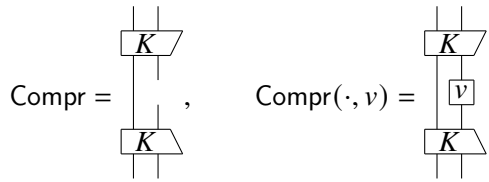
$$\begin{aligned} \mathsf{Compr}\colon \mathcal{M}_m &\to \mathcal{M}_m \otimes \mathcal{M}_m \cong \mathcal{M}_{m^2} \\ v &\mapsto \mathsf{Compr}(\cdot, v) \end{aligned}$$

4.1 Characterising Compr diagrammatically

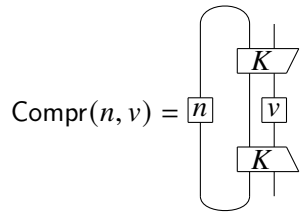
We now characterise Compr, $\mathsf{Compr}(\cdot, v)$, and $\mathsf{Compr}(n, v)$ in the diagrammatic calculus for **FHilb**. This will allow us to generate simple examples of Compr in a systematic manner. Equation (4) states:

$$\mathsf{Compr}(\cdot, v) = \sum_l K_l v K_l^*$$

Diagrammatically, this gives us:



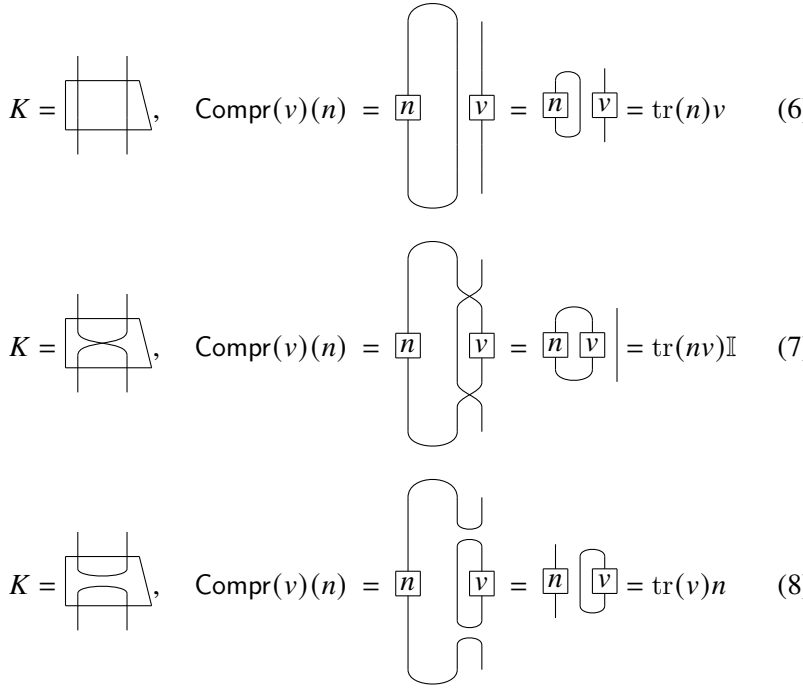
The application of $\mathsf{Compr}(\cdot, v)$ to n is then:



(5)

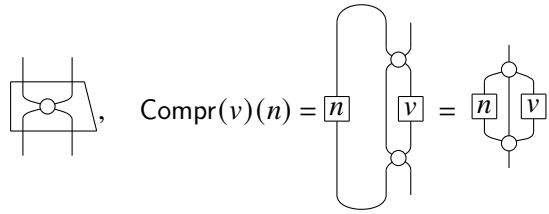
This style of diagram corresponds to the usual DisCoCat diagram style via some reshaping. Given that we have representations of v and of n, what should the K_l look like? In full generality, parameters of the K_l could be perhaps inferred using regression techniques, in a similar approach to that suggested in Lewis (2019b), inspired by Socher et al. (2013), or via methods like those in Baroni and Zamparelli (2010); Grefenstette et al. (2013). However, we can also consider purely "structural" morphisms, generated from cups, caps, swaps, and spiders. In the following, we give a number of options to specify K. We divide up the internal structure of K by specifying the number of spiders inside K. In the following, $\mathrm{diag}(A)$ denotes the matrix obtained by setting all off-diagonal elements of A to 0, $\mathrm{tr}(A)$ denotes the trace of A, and $\mathbb{I}$ denotes the identity matrix.

0 spiders



1 spider

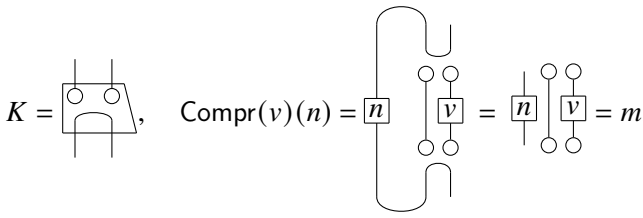
$$K = \quad , \qquad \mathsf{Compr}(v)(n) = \quad = \quad \tag{9}$$



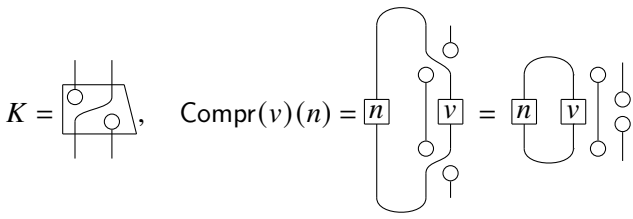
$$= \mathrm{diag}(n)\mathrm{diag}(v)$$

2 spiders

$$K = \quad , \qquad \mathsf{Compr}(v)(n) = \quad = \quad = mn \sum_{ij} v_{ij} \tag{10}$$

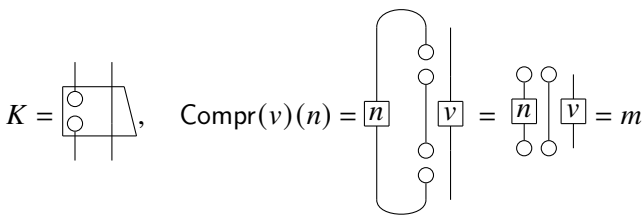


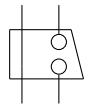
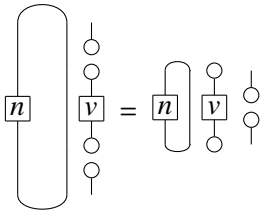
$$K = \quad , \qquad \mathsf{Compr}(v)(n) = \quad = \quad \tag{11}$$



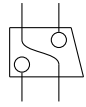
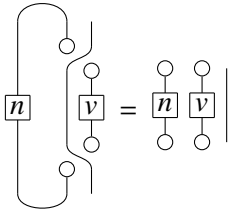
$$= m\mathrm{tr}(nv) \sum_{ij} |e_i\rangle \langle e_j|$$

$$K = \quad , \qquad \mathsf{Compr}(v)(n) = \quad = \quad = mv \sum_{ij} n_{ij} \tag{12}$$

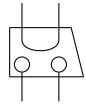
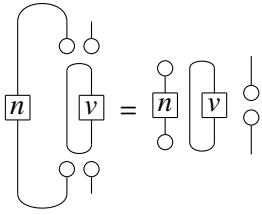


$K =$  , $\mathsf{Compr}(v)(n) =$ 

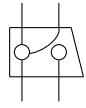
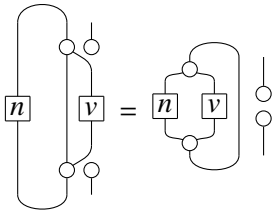
$$= \mathrm{tr}(n) \sum_{ij} v_{ij} \sum_{kl} |e_k\rangle \langle e_l| \tag{13}$$

$K =$  , $\mathsf{Compr}(v)(n) =$ 

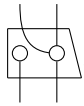
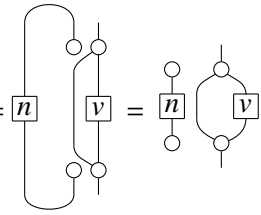
$$= \mathbb{I} \sum_{ij} n_{ij} \sum_{kl} v_{kl} \tag{14}$$

$K =$  , $\mathsf{Compr}(v)(n) =$ 

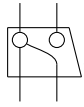
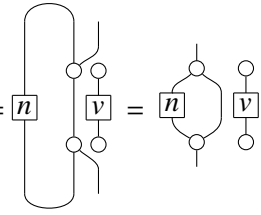
$$= \mathrm{tr}(v) \sum_{ij} n_{ij} \sum_{kl} |e_k\rangle \langle e_l| \tag{15}$$

$K =$  , $\mathsf{Compr}(v)(n) =$ 

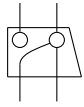
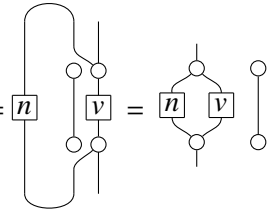
$$= \mathrm{tr}\left(\sum_{ij} n_{ij} v_{ij} |e_i\rangle \langle e_j| \right) \sum_{kl} |e_k\rangle \langle e_l| \tag{16}$$

$K =$  , $\mathsf{Compr}(v)(n) =$  (17)

$$= \sum_{ij} n_{ij} \mathrm{diag}(v)$$

$K =$  , $\mathsf{Compr}(v)(n) =$  (18)

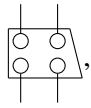
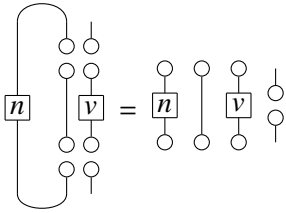
$$= \sum_{ij} v_{ij} \mathrm{diag}(n)$$

$K =$  , $\mathsf{Compr}(v)(n) =$  (19)

$$= m \sum_{ij} v_{ij} n_{ij} \, |e_i\rangle \langle e_j| = m\mathsf{Mult}(n, v)$$

3 spiders The instances with 3 spiders are subsumed by the instances with 2 spiders, since to have 3 spiders we would need two spiders with one leg and one spider with two legs. A spider with two legs is either a cup, cap, or the identity morphism, hence these have been included in the 2 spider instances.

4 spiders

$K =$  , $\mathsf{Compr}(v)(n) =$ 

$$= m \sum_{ij} n_{ij} \sum_{kl} v_{kl} \sum_{rs} |e_r\rangle \langle e_s| \tag{20}$$

This gives us a whole range of possible instantiations of Compr. Some of these options are more interesting than others. Options that give us a multiple of the identity matrix or a multiple of $\sum_{ij} |e_i\rangle \langle e_j|$ for orthonormal basis $\{|e_i\rangle\}_i$ are less interesting since this means that all phrase representations will be mapped to the same psd matrix, differing only by a scalar. This means that although hyponymy information may be preserved, information about similarity will be lost.

The instantiations that preserve some information about similarity are (6), (8), (9), (10) (12), (17), (18), and (19), the last of which was already shown to work well in Lewis (2019a). Of these, (6), (8), (10) (12), (17), and (18) only preserve similarity information about one of the components of the sentence, i.e. either the noun or the verb. The operators (9) and (19) each preserve some similarity information about both words.

All of these operators can be seen as a lifting of the psd matrix for the verb from the space $W^* \otimes W$ into $W \otimes W^* \otimes W^* \otimes W$, such that it can then act on the psd matrix for the noun to produce a psd matrix for the sentence. We will now examine each operator and their theoretical justifications.

Each of the operators (6) and (8) has the effect that one of the operators v or n is preserved and it is weighted by the trace of the other. These models assert that similarity between these short word combinations is best modelled by taking account only of one of the words (v or n), but that the entailment relation between the sentences is also affected by the hyponymy relation obtaining between the other word (n or v), since $w_1 \leqslant w_2 \implies \mathrm{tr}(w_1) \leq \mathrm{tr}(w_2)$ for psd matrices w_1 and w_2

The models represented in equations (12) and (10) have similar properties. One of the operators v or n is preserved, weighted by the sum of

the entries of the other operator. Again, this asserts that similarity between these short word combinations is best modelled by taking account only of one of the words, but that the entailment relationship is also affected by the other word in the sentence, since $w_1 \leqslant w_2 \implies \sum_{ij} w_{1_{ij}} \leq \sum_{ij} w_{2_{ij}}$ for psd matrices w_1 and w_2.

In equations (17), (18) and (9), we use the psd matrix with off-diagonal elements discarded, $diag(w)$. This construction is somewhat harder to interpret. We can view dropping the off-diagonal elements as losing information about the correlations of the dimensions of the vector space with each other, retaining only information about how a given dimension of the vector space correlates with itself. Furthermore, given a psd matrix w, the closest diagonal matrix to w under the Frobenius norm is $diag(w)$. We therefore obtain the nearest representation to w that neglects correlations between dimensions. In (17) and (18) we again only retain one of the words in the short phrase for similarity purposes, either v or n, and weight this by the sum of the entries in the other (n or v), thereby contributing to the entailment strength between the short sentences. In (9) we retain both the diagonal matrices, and form their pointwise multiplication. This retains semantic information about both the noun and the verb, and at the same time contributes to the entailment strength between the two sentences.

Lastly, the model (19) is an extension of the models seen in Grefenstette and Sadrzadeh (2011) and Kartsaklis et al. (2012), applied to intransitive verbs. This model uses Frobenius algebras in the category $\mathbf{CPM}(\mathbf{FHilb})$, as was done in Piedeleu et al. (2015), to lift the representation of the verb from the space $W^* \otimes W$ to the space $W \otimes W^* \otimes W^* \otimes W$. The outcome of this is that the matrices are pointwise multiplied together. This operation retains the full matrix for each word.

5. Demonstration

To test these composition methods, we follow the setup in Lewis (2019a, 2020). We firstly build psd matrices using GloVe vectors and lists of hyponyms. We use WordNet (Miller, 1995) to determine the hyponyms of a given word. We use GloVe vectors of dimension 50. We use a set of datasets from Kartsaklis and Sadrzadeh (2016) that contain pairs of short phrases, for which the first either does or does not entail the second.

In addition, we use a graded form of the Löwner ordering to measure hyponymy, introduced in Lewis (2019a), since in general the crisp Löwner ordering will not be obtained between two psd matrices A and B. This graded form is measured as follows. Given two psd matrices A and B, if $A \leqslant B$ then $A + D = B$ where D is itself a psd matrix. If this does not hold, we can add an error term E so that

$$A + D = B + E.$$

In the worst case, we can set $E = A$, so that $D = B$, and in fact we will always have that $E \leqslant A$. A graded measure of hyponymy is obtained by comparing the size of E and A. We set

$$k_E = 1 - \frac{||E||}{||A||},$$

where $||\cdot||$ denotes the Euclidean norm, $||A|| = \sqrt{\mathrm{tr}(A^*A)}$. The crisp Löwner order is recovered in the case that $E = 0$, so that $k_E = 1$. A second measure of graded hyponymy is obtained as follows:

$$k_{BA} = \frac{\sum_i \lambda_i}{\sum_i |\lambda_i|} \tag{21}$$

where λ_i is the ith eigenvalue of $B - A$ and $|\cdot|$ indicates absolute value. This measures the proportions of positive and negative eigenvalues in the expression $B - A$. If all eigenvalues are negative, $k_{BA} = -1$, and if all are positive, $k_{BA} = 1$. This measure is balanced in the sense that $k_{BA} = -k_{AB}$.

Datasets The datasets were originally collected for Kartsaklis and Sadrzadeh (2016). They consist of ordered pairs of short phrases in which the first entails the second, and also the same pair in the opposite order, so that the first phrase does not entail the second. The datasets were gathered using WordNet as source. The datasets contain intransitive sentences, of the form *subject verb*, verb phrases, of the form *verb object* and transitive sentences, of the form *subject verb object*. For example:

summer finish, season end, `true`

season end, summer finish, `false`

The datasets have a binary classification, so we measure performance using area under receiver operating characteristic (ROC) curve. If we imagine that our graded measure is converted to a binary measure by giving a threshold, area under ROC curve measures performance at all cutoff thresholds. A value of 1 means that the graded values are in fact a completely correct binary classification, a value of 0.5 means that the graded values are randomly correlated with the correct classification, and a value of 0 means that the graded values are binary values that are classified in exactly the wrong way (a value of 1 is mapped to 0 and 0 to 1).

Models We test the following models, for $n, v \in \mathcal{M}_m$. We denote by $\mathrm{diag}(A)$ the matrix obtained by setting all off-diagonal elements of A to 0. In order to retain the property that the maximum eigenvalue is less than or equal to 1, we divide by the dimension m or by m^2 where necessary.

1. **Traced noun**: $\mathsf{Compr}(n, v) = \frac{\mathrm{tr}(n)}{m} v$
2. **Traced verb**: $\mathsf{Compr}(n, v) = \frac{\mathrm{tr}(v)}{m} n$
3. **Diag**: $\mathsf{Compr}(n, v) = \mathrm{diag}(n)\mathrm{diag}(v)$
4. **Summed noun**: $\mathsf{Compr}(n, v) = \frac{v}{m^2} \sum_{ij} n_{ij}$
5. **Summed verb**: $\mathsf{Compr}(n, v) = \frac{n}{m^2} \sum_{ij} v_{ij}$
6. **Diag verb**: $\mathsf{Compr}(n, v) = \frac{\mathrm{diag}(v)}{m^2} \sum_{ij} n_{ij}$
7. **Diag noun**: $\mathsf{Compr}(n, v) = \frac{\mathrm{diag}(n)}{m^2} \sum_{ij} v_{ij}$
8. **Mult**: $\mathsf{Compr}(n, v) = \sum_{ij} v_{ij} n_{ij} \, |e_i\rangle \langle e_j|$

Above, we have specified models for sentences of the form *subj verb*. For verb phrases, we treat the verb as v and the object as n, so the models differ based on the grammatical type of the word, rather than its position in the argument list. For sentence of the form *subject verb object*, we first combine the verb and the object, according to their grammatical type, and then treating this verb phrase as an intransitive verb, combine the subject and verb phrase, again according to grammatical type. So, for example, iterating

the composition **Traced Verb** on psd matrices s, v, o for subject, verb, and object, we obtain:

$$\mathsf{Compr}(s, \mathsf{Compr}(o, v)) = \mathsf{Compr}(s, \frac{\mathrm{tr}(v)}{m} o) = \frac{\mathrm{tr}(v)\mathrm{tr}(o)}{m^2} s$$

We also test two combined models:

1. **Traced addition**: $\mathsf{Compr}(n, v) = \frac{\mathrm{tr}(n)}{2m} v + \frac{\mathrm{tr}(v)}{2m} n$
2. **Summed addition**: $\mathsf{Compr}(n, v) = \frac{v}{m^2} \sum_{ij} n_{ij} + \frac{n}{m^2} \sum_{ij} v_{ij}$

We compare with a verb-only baseline and with Fuzz and Phaser. These last two are tested in two directions:

1. **Verb only**: $\mathsf{Verb\ only}(n, v) = v$
2. **Fuzz**: $\mathsf{Fuzz}(n, v) = \sum_i \sqrt{p_i} P_i n P_i \sqrt{p_i}$ where $\sum_i p_i P_i$ is the spectral decomposition of v
3. **Fuzz switched**: $\mathsf{Fuzz\text{-}s}(n, v) = \sum_i \sqrt{q_i} Q_i v Q_i \sqrt{q_i}$ where $\sum_i q_i Q_i$ is the spectral decomposition of n
4. **Phaser**: $\mathsf{Phaser}(n, v) = \sqrt{v} n \sqrt{v}$
5. **Phaser switched**: $\mathsf{Phaser\text{-}s}(n, v) = \sqrt{n} v \sqrt{n}$

We also compare with the best results from Kartsaklis and Sadrzadeh (2016). To test for significance of our results, we prepare 100 bootstrap sample datasets, i.e. for each dataset with n datapoints, we create 100 datasets with n datapoints by sampling from the original with replacement (Efron, 1992) to create a distribution over the test statistic, and compare between models using a two sample t-test. We apply the Bonferroni correction to compensate for multiple comparisons. To compare with the figures from Kartsaklis and Sadrzadeh (2016) we use the same bootstrapped samples and perform a one sample t-test.

Table 2. Area under ROC curve for k_E and k_{BA} graded hyponymy measures. Figures are mean values of Area under ROC curve calculated for 100 datasets of size n sampled from the original dataset with replacement, where n is the size of the original dataset. $-'$ indicates significantly better than the best model from Kartsaklis and Sadrzadeh (2016), $p < 0.01$, $-^*$ indicates significantly better than both variants of **Fuzz**, $p < 0.01$, $-^+$ indicates significantly better than both variants of **Phaser**, $p < 0.01$.

	k_E measure			k_{BA} measure		
	SV	VO	SVO	SV	VO	SVO
K&S 2016 best	0.84	0.82	0.86	0.84	0.82	0.86
Verb only	0.599	0.588	0.647	0.787	0.743	0.839
Fuzz	0.841	0.798	0.908	0.921	0.899	0.971
Fuzz sw.	0.807	0.773	0.919	0.932	0.913	0.967
Phaser	0.819	0.755	0.913	0.920	0.888	0.970
Phaser sw.	0.749	0.726	0.918	0.931	0.912	0.971
Traced noun	0.847	0.808^{+}	$0.952'^{*+}$	$0.934'^{+}$	$0.909'$	$0.976'^{*}$
Traced verb	0.811	0.771	$0.936'^{*+}$	$0.936'^{*+}$	$0.911'$	$0.959'$
Diag	$0.899'^{*+}$	$0.860'^{*+}$	$0.946'^{*+}$	$0.936'^{+}$	$0.915'^{+}$	$0.969'$
Sum. noun	$0.867'^{+}$	0.808^{+}	$0.936'^{*+}$	$0.926'$	$0.883'$	$0.971'$
Sum. verb	0.799	0.776^{+}	$0.902'$	$0.891'$	$0.885'$	$0.935'$
Diag verb	$\mathbf{0.919}'^{*+}$	$\mathbf{0.874}'^{*+}$	$\mathbf{0.965}'^{*+}$	$0.920'$	$0.871'$	$0.962'$
Diag noun	$0.874'^{+}$	$0.853'^{*+}$	$0.959'^{*+}$	$0.882'$	$0.874'$	$0.954'$
Mult	0.848^{+}	$0.816'^{*+}$	$0.943'^{*+}$	$\mathbf{0.944}'^{*+}$	$\mathbf{0.916}'^{*+}$	$0.970'$
Traced add.	$0.869'^{+}$	$0.831'^{*+}$	$\mathbf{0.965}'^{*+}$	$0.936'^{*+}$	$0.909'$	$\mathbf{0.985}'^{*+}$
Sum. add.	$0.858'^{+}$	0.826^{*+}	$0.940'^{*+}$	$0.920'$	$0.896'$	$0.968'$

5.1 Results

Results are presented in table 2. As in previous work, the k_{BA} measure performs better than the k_E measure across almost every model and dataset. In most cases, models in which only the verb contributes to the semantic information of the sentence, i.e. **Traced noun**, **Sum. noun**, and **Diag verb** perform better than models in which only the noun contributes to the semantic information of the sentence. Exceptions are the SV dataset under the k_{BA} measure, where **Traced verb** marginally outperforms **Traced noun**, and the VO dataset under the k_{BA} measure where **Traced verb** outperforms **Traced noun**, **Sum. verb** outperforms **Sum. noun**, and **Diag noun** outperforms **Diag verb**.

Table 3. Mean values of area under ROC curve across the same 100 bootstrapped datasets

	k_E measure			k_{BA} measure		
	SV	VO	SVO	SV	VO	SVO
Verb only	0.599	0.588	0.647	0.787	0.743	0.839
Diag verb only	0.766	0.721	0.800	0.786	0.745	0.830
Sum verb	0.692	0.682	0.689	0.649	0.636	0.642
Trace verb	0.752	0.792	0.794	0.704	0.659	0.742
Subj only	0.710	-	0.725	0.906	-	0.926
Diag subj only	0.870	-	0.907	0.902	-	0.929
Sum subj	0.860	-	0.878	0.833	-	0.847
Trace subj	0.877	-	0.908	0.853	-	0.889
Obj only	-	0.680	0.647	-	0.891	0.888
Diag obj only	-	0.844	0.834	-	0.745	0.883
Sum obj	-	0.791	0.733	-	0.741	0.674
Trace obj	-	0.858	0.813	-	0.815	0.754

To investigate the pattern that models retaining the meaning of the verb tend to perform better, we run another set of baseline models that look at performance of single words - i.e. the verb alone, the subject or object alone, their diagonals, and at the magnitude of the trace or sum of each. These are presented in table 3. We see that although the verb representations

themselves do not score particularly highly, just comparing the trace or the sum of the nouns gives a remarkably good score, at least under the k_E measure. It would therefore seem that the benefit of the models that preserve the semantic information of the verb is in the information on hyponymy received from the noun(s).

Looking at table 3 another striking pattern is that under the k_E measure, discarding off-diagonal elements on the matrices (**Diag verb only**, **Diag subj only**, **Diag obj only**) increases the graded hyponymy values between the words, in many cases by a substantial amount. Recall the way that these matrices are built. We take vectors for the hyponyms of the nouns, form the rank-1 projector corresponding to each vector, add them together, and then normalise. Discarding the off-diagonal elements of these matrices means that we are only interested in how the dimensions of the vector spaces are correlated with themselves, and not in how dimensions of the vector space correlate with each other.

As mentioned, under the Frobenius norm, forming a diagonal approximation to a given square matrix is optimized by discarding off-diagonal elements, so that the resulting diagonal matrix is the nearest approximation to the original. If the square matrix is row or column diagonally dominant, meaning that the diagonal elements of the matrix are larger than the sum of the absolute values of the off-diagonals in that row/column, then the diagonal approximation is particularly close. We checked the extent to which the matrices built are diagonally dominant, however we found that there is not a single row/column across the whole set of matrices for which the diagonal value dominates. The impact of discarding off-diagonal elements is therefore not due to the diagonal matrices being particularly good approximations to the original matrix.

Furthermore, under the k_E measure, even taking the trace or the sum of the matrix improves the hyponymy score. We can see how the interaction of discarding off-diagonal elements, together with weighting the resulting representation, produces the high scores we see in table 2. Investigation into the reason why discarding the off-diagonal elements of the matrices, or even taking the trace or sum of these matrices positively affects the k_E measure is a line of future work.

This pattern does not hold for the k_{BA} measure. Here, discarding the off-diagonal elements has a slight detrimental effect in most cases. At the

single word level, scores are generally higher than for the k_E measure. This was not seen in previous work (Lewis, 2019a), which could be a factor of the particular vocabulary used in the dataset.

Under the k_{BA} measure, the model **Mult** performs strongly on the SV and VO datasets. The **Mult** model is an extension of the Frobenius copy models proposed in Kartsaklis et al. (2012), Kartsaklis et al. (2016), and was originally proposed for use in the **CPM**(**FHilb**) context in Piedeleu et al. (2015). The success of a multiplicative model has been shown in previous work at both the pure vector level (Grefenstette and Sadrzadeh, 2011; Kartsaklis et al., 2012, 2016) and at the density matrix level(Piedeleu et al., 2015; Lewis, 2019a), and so it is unsurprising that good results are also seen here. This is perhaps borne out by the fact that under the k_{BA} measure, the full density matrices perform as well as, or better, than matrices with the off-diagonals discarded, unlike the k_E measure, and also by the fact that the single matrices perform relatively strongly by themselves. As a consequence, taking both full unreduced matrices and combining them has the potential to produce a strong representation.

On the SVO dataset, the traced addition model performs strongly. Recall that this model applied at the level of two matrices is an average of the two matrices, weighted by the trace of the other. Applied to the SVO datasets, we firstly combine the V and O matrices and then combine this with the S matrix. This ordering is motivated by grammatical considerations. This has the impact that the S matrix contributes more to the overal semantics of the sentence. Examining the performance of **Subj only** in table 3 and of **Traced noun** in table 2, we see that both of these models perform strongly on the SVO dataset, suggesting that the dominance of the matrix for S, coupled with additional information about V and O, enables the **Traced addition** model to perform strongly here. It would be possible to include a number of further linear combinations of these operators, and this is an area of future research.

Across every model, for both hyponymy measures, performance on the VO dataset is worst and performance on the SVO dataset is best. This may be attributable to the sizes of the datasets: SV has 270 sentence pairs, SVO has 140 sentence pairs, whereas VO has 436 sentence pairs. A potential aspect that may make a given dataset more difficult is the number of different senses a word can have. We computed the mean number of synsets per word as given in WordNet, and found that this was fairly consistent across datasets,

with SV having on average 8.9 synsets per word, VO having 8.5 synsets per word, and SVO having 8.7 synsets per word. We also make a comparison of the total number of synsets for the words in the first sentence of a pair with the total number of synsets for the words in the second sentence of a pair. The number of synsets could affect performance adversely, if a word is used in many different senses, or it could affect performance positively, by enabling the creation of a matrix that is more mixed. If a word w_1 has a broader meaning than a word w_2, then we would expect w_2 to be a hyponym of w_1. We therefore compute the total number of synsets for each sentence in the pairs of sentences for which entailment holds. We look at whether the total number of synsets for the first, entailing, sentence is smaller than the total number of synsets for the second, entailed, and more general sentence. If the second sentence has a larger number of synsets that the first, we say that there is a *positive synset difference*.

Table 4. Proportions of entailing sentence pairs with positive synset difference in each dataset from Kartsaklis and Sadrzadeh (2016).

Dataset	SV	VO	SVO
No. entailing pairs	135	218	70
No. positive synset difference	74	113	44
Proportion with positive synset difference	0.54	0.51	0.62

Table 4 shows that the SVO dataset has a slightly higher proportion of sentence pairs with a positive synset difference. However, it is not clear whether this is beneficial. It could be the case that having a large number of synsets produces noise in the word representations which can then be detrimental when combining words or computing graded hyponymy, particularly given that the synsets cover the differing meanings of ambiguous words. In order to investigate this, we take a deeper look at the performance of the models on each dataset. In particular, we look at the sentences in each dataset for which more than half the models fail to correctly predict the entailment direction. In table 5 we see that out of the 'difficult' sentences – those for which more than half the models fail – a smaller proportion (between 0 and 0.27) have a positive synset difference, in comparison to the proportion of sentences with positive synset difference in each dataset.

Similarly, in table 6, we see that of those sentences where all models

Table 5. Numbers of sentences that more than half the proposed models fail to correctly predict (Num. S), together with number of those sentences with positive synset difference (Pos. Syn) and the corresponding proportion.

	k_E measure			k_{BA} measure		
	SV	VO	SVO	SV	VO	SVO
Num. S	32	74	10	36	74	10
Pos. syn	6	18	0	8	20	0
Proportion	0.18	0.24	0	0.22	0.27	0

correctly predict the entailment value, a higher proportion (between 0.69 and 0.78) of sentences with a positive synset difference are seen. This indicates that although modelling a number of different senses in each word could potentially be detrimental, in fact using these different senses can be beneficial for this task. This is potentially because many of the senses for a given word in WordNet are related to each other.

Table 6. Numbers of sentences that all models predict correctly (Correct S), together with number of those sentences with positive synset difference (Pos. Syn) and the corresponding proportion.

	k_E measure			k_{BA} measure		
	SV	VO	SVO	SV	VO	SVO
Correct S	134	206	84	152	238	90
Pos. syn	96	150	64	106	168	70
Proportion	0.71	0.73	0.76	0.69	0.71	0.78

6. Computing values for Compr from data

We undertake a small scale experiment to compute possible values for the operator Compr from the entailment datasets. In order to do this we introduce some simplifications to our model. We originally stated that Compr should

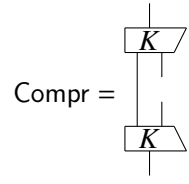
be a map $\mathsf{Compr} : \mathcal{M}_m \to \mathcal{M}_m \otimes \mathcal{M}_m$, so that when applied to a positive operator for a verb v , it gives $\mathsf{Compr}(-, v) : \mathcal{M}_m \to \mathcal{M}_m$. In fact, in the case of intransitive verbs, we can consider $\mathsf{Compr}(-, v)$ to be a map from the noun space $\mathcal{M}_m$ to the sentence space $\mathcal{M}_s$, i.e. we have $\mathsf{Compr}(-, v) : \mathcal{M}_m \to \mathcal{M}_s$. For a pair of intransitive sentences, when $\mathsf{Compr}(-, v)$ is applied to a noun n, we then require that $\mathsf{Compr}(n_1, v_1) \leqslant \mathsf{Compr}(n_2, v_2)$. In particular, we can specify that the sentence space $\mathcal{M}_s$ is the set of psd matrices of dimension 1, i.e. the positive reals. We use this simplification, and say that for a sentence S_1 to entail a sentence S_2, we require that $S_1 \leq S_2$, where the order now refers to the usual order on the reals.

The second simplification we employ is to say that the operator Compr is pure, so that in equation (4), repeated below,

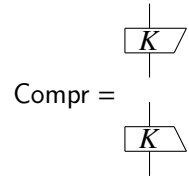
$$\begin{aligned} \mathsf{Compr}\colon \mathcal{M}_m &\to \mathcal{M}_m \otimes \mathcal{M}_m \cong \mathcal{M}_{m^2} \\ v &\mapsto \sum_l K_l v K_l^* \end{aligned}$$

we have only one matrix K rather than a set of matrices $\{K_l\}_l$.

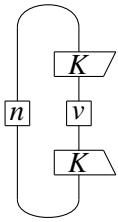
Diagrammatically, the first simplification appears as follows:



and the second gives:



The application of $\mathsf{Compr}(\cdot, v)$ to n is then:

$\mathsf{Compr}(n, v) =$ 

(22)

i.e. we have $S = \mathrm{tr}(nKvK^*)$.

Our aim is now to find a set of parameters for one matrix K, such that for given n_1, v_1, n_2, v_2,

$$\mathrm{tr}(n_1 K v_1 K^*) \leq \mathrm{tr}(n_2 K v_2 K^*)$$

whenever S_1 entails S_2.

We require

$$\mathrm{tr}(n_1 K v_1 K^*) \leq \mathrm{tr}(n_2 K v_2 K^*)$$

We therefore propose the loss function

$$\mathcal{L}(K) = \mathrm{tr}(n_1 K v_1 K^*) - \mathrm{tr}(n_2 K v_2 K^*)$$

which we want to minimise.

Since this function can get arbitrarily small, we also introduce a penalty term to ensure that the magnitude of K does not get too large, specifically the square of the Frobenius norm of K, giving us:

$$\mathcal{L}(K) = \mathrm{tr}(n_1 K v_1 K^*) - \mathrm{tr}(n_2 K v_2 K^*) + c\,\mathrm{tr}(K^* K) \tag{23}$$

where c is a hyperparameter that we will choose. The gradient of $\mathcal{L}$ with respect to K_{mj} is

$$\frac{\partial \mathcal{L}}{\partial K_{mj}} = \sum_{ig} (n_{1_{im}} K_{ig} v_{1_{gj}} - n_{2_{im}} K_{ig} v_{2_{gj}}) + \sum_{ig} (n_{1_{mi}} K_{ig} v_{1_{jg}} - n_{2_{mi}} K_{ig} v_{2_{jg}}) + 2K_{mj}$$

and we update K using gradient descent, that is, we update K according to:

$$K \mapsto K - h\frac{\partial \mathcal{L}}{\partial K_{mj}}$$

where h, the step size, will be selected via hyperparameter optimisation. We optimise over 3 hyperparameters: the value of c in (23), the step size h for the gradient descent, and the number of times we run over the dataset.

To train the operator we use the SV and VO datasets from Kartsaklis and Sadrzadeh (2016). We train only on the true entailing sentences, since by construction of the model and of the datasets, if we are correct on the true entailing pairs, we will also be correct on the false entailing pairs, and furthermore there is exactly one false entailing pair for each true entailing pair. We use 0.8 of the dataset for training, and 0.2 for testing. We use accuracy to measure performance, since our model now makes binary predictions. Within the training set, we use 5-fold cross-validation to pick the optimum values of the hyperparameters. The best values of the hyperparameters were $c = 0.01$, $h = 0.01$, and 2 dataset repetitions. The mean validation accuracy for the highest scoring parameters was 0.805 and the mean training accuracy was 0.807.

To assess the performance on the test sets, we formed 100 bootstrap sample datasets from each of the SV test set and the VO test set, and assessed accuracy on each sample dataset. We found that the mean test accuracy for the SV dataset across the 100 datasets was 0.924 with a standard deviation of 0.039, and the mean test accuracy for the VO dataset across the sample datasets was 0.757 with a standard deviation of 0.023.

The performance on the SV test set is surprisingly high. We looked at the test set sentences to examine whether they were 'easy' in terms of the positive synset difference, and found that a slightly higher proportion of sentence pairs in the test set have a positive synset difference in comparison to the dataset as a whole (0.59 vs. 0.54). On the VO dataset, where performance was lower, 0.54 sentences had a positive synset difference in comparison with 0.51 in the whole dataset, so it does not seem as if this property of the sentence pairs can explain the difference in performance.

To look at performance at the sentence level, we again compare with a simple count of the synsets in each sentence. Within the SV dataset, out of the sentences that were correctly predicted to entail, 43 had a negative

synset difference, indicating that the learned Compr function was able to overcome the difference even when the entailing sentence was in some sense broader than the entailed sentence. Out of the sentences that were incorrectly predicted not to entail, 4 did have a positive synset difference. Similarly, on the VO dataset, out of the sentences that were correctly predicted to entail, 71 had a negative synset difference. Out of the sentences that were incorrectly predicted not to entail, 17 had a positive synset difference. In general, the Compr function was able to map sentences into the correct relation.

Whilst these results are a little mixed, performance is promising. We are training a matrix of dimension 50, whereas in full generality we would have many more parameters to play with. Reversing some of the simplifications and introducing a more sophisticated training regime would help performance. One avenue for exploration is to look at different training regimes that can take into account the differences in grammatical structure between SV and VO type sentences, as well as generalising to more complex sentence structures.

7. Discussion

We have presented a general composition rule called Compr for converting a psd matrix for a functional word such as a verb or an adjective into a CP map that matches the grammatical type of the word. Compr preserves hyponymy, in contrast to previous approaches like Fuzz and Phaser. In full generality, we would like to learn the parameters of Compr from text corpora coupled with entailment datasets such as SNLI (Bowman et al., 2015b). In this paper we have provided a step towards that. We first looked at ways in which Compr may be defined using just structural morphisms such as cups, caps, and spiders. In these experiments, models that retained semantic information about the verb whilst using entailment information from the nouns were most successful. The best model variant was dependent on the variant of graded entailment used. Under the k_{BA} measure, the **Mult** model was most successful. This pattern has been seen in previous research Grefenstette and Sadrzadeh (2011); Kartsaklis et al. (2012), and is hence to be expected. Similarly, the fact that the **Traced Addition** model is successful in unsurprising, since additive models have been successful in e.g. Mitchell and Lapata (2010). What is more surprising is that the **Diag verb** model

works well. Under this model, the off-diagonal elements of the density matrix were discarded, which would seem to result in information loss. Digging into this phenomenon further, we saw that at a single word level, under the k_E measure, taking the diagonal of the matrix almost always improved the hyponymy predictions, as did taking the trace or the sum of the entries of the matrix. More research into this and into the properties of the k_E measure is ongoing.

There was also a difference in performance on the datasets. Overall, the VO dataset was harder than the SV dataset, and the SVO dataset got the highest performance. We analysed the sentences of the datasets in terms of their numbers of synsets, and found that the harder datasets tended to have more sentence pairs with larger numbers of synsets in the entailing sentence. The models we proposed tended to fail more frequently on these sentences.

We proceeded to provide a model of Compr where the parameters of Compr were estimated from information about entailment in the SV and VO datasets. Whilst performance of this learnt model was mixed, this is likely due to simplifications in the model. We again saw the pattern that the VO dataset had lower performance than the SV dataset. We also found that the Compr map was able to provide the correct prediction in many cases where the entailing sentences had a negative synset difference.

Future work for estimating the parameters of Compr from text corpora will involve desimplifying the model, so that sentences are represented by psd matrices rather than positive scalars and so that Compr is no longer pure. We then propose to extend the learning mechanisms begun here to use resources such as SNLI Bowman et al. (2015b) as input. Ideally, we would like to learn the original psd matrices from text corpora as well. This has been done for ambiguity in Piedeleu et al. (2015); Meyer and Lewis (2020).

The approach we have taken, namely that of defining a map that converts representations of functional words to a higher-order type, has also been seen in vector-based models of meaning. In Kartsaklis et al. (2012), Grefenstette and Sadrzadeh (2011), word vectors and matrices are converted using Frobenius algebras which the composition Mult is a direct analogue. Furthermore, in Mitchell and Lapata (2010), and recapitulated in Lewis (2019b), a bilinear map $C : N \otimes N \rightarrow N$ that determines the composition of two vectors is proposed. Under this approach, we would have

$$\rho(subj\ verb) = C(\rho(subj) \otimes \rho(verb))$$

and

$$\rho(subj\ verb\ obj) = C(\rho(subj) \otimes C(\rho(verb) \otimes \rho(obj)))$$

Our approach is an analogue to this one within the realm of psd matrices and CP maps.

At present, we have given two possible graded measures of hyponymy. More research into these measures is needed, including how they interact with the composition methods we have specified. The datasets we have so far tested on are relatively small and simple, and therefore testing on datasets with more complex entailment relations such as FraCas (Cooper et al., 1994) and on larger datasets such as the Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015b) is an important next step.

Another area of research is to be able to model hyponymy, composition, and their interaction in what are known as *downwardly monotone* contexts, using the natural logic introduced in Barwise and Cooper (1981); MacCartney and Manning (2007). In particular, the datasets we have so far used all contain what are called *upwardly monotone* inferences. Essentially, in the titular sentence pair 'Cats climb' and 'Mammals move', the inference from the first sentence to the second holds only if we assume that the sentences are existentially quantified. In this context, the inference holds. However, if we were to assume that the first sentence is universally quantified, i.e. that it states 'All cats climb', the inference would not hold. Rather, an inference such as 'All tabbies climb' would hold, where the second noun is now a hyponym of the first rather than vice versa. This kind of inference is called *downwardly monotone*. Yanaka et al. (2019) argue that many neural models of word meaning are ineffective at modelling this kind of inference, partly due to the fact that the larger datasets such as SNLI or MultiNLI do not have many downward entailing pairs. In that paper they introduce a new dataset focusing on modelling a wider range of lexical and logical inferences, which can be used to augment existing NLI datasets. Other research that covers this area includes the previously mentioned Cooper et al. (1994) as well as Bowman et al. (2015a) and Geiger et al. (2018), although the latter two depend heavily on syntax rather than lexical semantics. The former is very small for the purposes of large neural models but may have potential for our purposes. Whilst we do not yet have a model of logical composition, assertion, or truth, work is currently ongoing to develop a model of negation within this framework (Lewis, 2020), and linking to recent work on quan-

tification (Hedges and Sadrzadeh, 2019; Dostál et al., 2020) will be of use here *.

References

Bankova, Dea, Bob Coecke, Martha Lewis, and Dan Marsden. 2018. Graded hyponymy for compositional distributional semantics. *Journal of Language Modelling* 6 (2): 225–260.

Baroni, Marco and Roberto Zamparelli. 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*, pages 1183–1193. Cambridge, MA: Association for Computational Linguistics.

Barwise, J. and R. Cooper. 1981. Generalized quantifiers and natural language. In *Philosophy, language, and artificial intelligence*, pages 241–301. Springer.

Bowman, Samuel, Christopher Potts, and Christopher D Manning. 2015a. Recursive neural networks can learn logical semantics. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, pages 12–21.

Bowman, Samuel R., Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015b. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642. Lisbon, Portugal: Association for Computational Linguistics. doi:10.18653/v1/D15-1075.

Clarke, Daoud. 2009. Context-theoretic semantics for natural language: an overview. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 112–119. Athens, Greece: Association for Computational Linguistics.

Coecke, B. 2019. The mathematics of text structure. *arXiv:1904.03478* .

Coecke, B., E. Grefenstette, and M. Sadrzadeh. 2013. Lambek vs. Lambek: Functorial vector space semantics and string diagrams for Lambek calculus. *Ann Pure Appl Log.* 164 (11): 1079–1100.

Coecke, B., M. Sadrzadeh, and S. Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. *Lambek Festschrift, special issue of Linguistic Analysis* .

Coecke, Bob and Konstantinos Meichanetzidis. 2020. Meaning updating of density matrices. *Journal of Applied Logics* 2631 (5): 745.

*ACKNOWLEDGEMENTS: We thank the anonymous reviewers for their detailed and helpful comments. Martha Lewis was partially funded by Veni grant 'Metaphorical Meanings for Artificial Agents'

Conneau, Alexis, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

Cooper, R., R. Crouch, J. Van Eijck, C. Fox, J. Van Genabith, J. Jaspers, H. Kamp, M. Pinkal, M. Poesio, S. Pulman, et al. 1994. Fracas: A framework for computational semantics. *Deliverable D6* .

Dagan, I., O. Glickman, and B. Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Dostál, M., M. Sadrzadeh, and G. Wijnholds. 2020. Fuzzy generalised quantifiers for natural language in categorical compositional distributional semantics. *Mathematics, Logic, and their Philosophies: Essays in Honour of Mohammad Ardeshir* page 135.

Efron, B. 1992. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*. Springer.

Gagné, Christina L., Thomas L. Spalding, Patricia Spicer, Dixie Wong, Beatriz Rubio, and Karen Perez Cruz. 2020. Is buttercup a kind of cup? Hyponymy and semantic transparency in compound words. *Journal of Memory and Language* 113: 104110. doi:10.1016/j.jml.2020.104110.

Geffet, Maayan and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 107–114. Ann Arbor, Michigan: Association for Computational Linguistics. doi:10.3115/1219840.1219854.

Geiger, Atticus, Ignacio Cases, Lauri Karttunen, and Christopher Potts. 2018. Stress-testing neural models of natural language inference with multiply-quantified sentences. *arXiv preprint arXiv:1810.13033* .

Grefenstette, E., G. Dinu, Y. Zhang, M. Sadrzadeh, and M. Baroni. 2013. Multi-step regression learning for compositional distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 131–142. Potsdam, Germany: Association for Computational Linguistics.

Grefenstette, Edward and Mehrnoosh Sadrzadeh. 2011. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404. Edinburgh, Scotland, UK.: Association for Computational Linguistics.

Hearst, Marti A. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*.

Hedges, J. and M. Sadrzadeh. 2019. A generalised quantifier theory of natural

language in categorical compositional distributional semantics with bialgebras. *Math. Struct. Comput. Sci.* 29 (6): 783–809.

Herbelot, Aurélie and Mohan Ganesalingam. 2013. Measuring semantic content in distributional vectors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 440–445. Sofia, Bulgaria: Association for Computational Linguistics.

Kartsaklis, Dimitri, Nal Kalchbrenner, and Mehrnoosh Sadrzadeh. 2014. Resolving Lexical Ambiguity in Tensor Regression Models of Meaning. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 212–217. Baltimore, Maryland: Association for Computational Linguistics. doi:10.3115/v1/P14-2035.

Kartsaklis, Dimitri and Mehrnoosh Sadrzadeh. 2016. Distributional inclusion hypothesis for tensor-based composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2849–2860. Osaka, Japan: The COLING 2016 Organizing Committee.

Kartsaklis, Dimitri, Mehrnoosh Sadrzadeh, and Stephen Pulman. 2012. A unified sentence space for categorical distributional-compositional semantics: Theory and experiments. In *Proceedings of COLING 2012: Posters*, pages 549–558. Mumbai, India: The COLING 2012 Organizing Committee.

Kartsaklis, Dimitri, Mehrnoosh Sadrzadeh, Stephen Pulman, and Bob Coecke. 2016. Reasoning about meaning in natural language with compact closed categories and frobenius algebras. In Jennifer Chubb, Ali Eskandarian, and Valentina Editors Harizanov, (eds.), *Logic and Algebraic Structures in Quantum Computing*, Lecture Notes in Logic, page 199–222. Cambridge University Press. doi:10.1017/CBO9781139519687.011.

Kiela, Douwe, Laura Rimell, Ivan Vulić, and Stephen Clark. 2015. Exploiting image generality for lexical entailment detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 119–124. Beijing, China: Association for Computational Linguistics. doi:10.3115/v1/P15-2020.

Kiros, Jamie, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Kotlerman, L., I. Dagan, I. Szpektor, and M. Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Nat. Lang. Eng.* 16 (4): 359.

Le, Matthew, Stephen Roller, Laetitia Papaxanthos, Douwe Kiela, and Maximilian Nickel. 2019. Inferring concept hierarchies from text corpora via hyperbolic embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3231–3241. Florence, Italy: Association for Computational Linguistics. doi:10.18653/v1/P19-1313.

Lenci, Alessandro and Giulia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 75–79. Montréal, Canada: Association for Computational Linguistics.

Lewis, Martha. 2019a. Compositional hyponymy with positive operators. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 638–647. Varna, Bulgaria: INCOMA Ltd. doi:10.26615/978-954-452-056-4_075.

———. 2019b. Compositionality for Recursive Neural Networks. *Journal of Applied Logics* 6: 709–724.

———. 2020. Towards logical negation in compositional distributional semantics. *Journal of Applied Logics* 7 (5): 771–794.

Lyons, John. 1968. *Introduction to theoretical linguistics*, volume 510. Cambridge University Press.

MacCartney, Bill and Christopher D. Manning. 2007. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200. Prague: Association for Computational Linguistics.

Maillard, Jean, Stephen Clark, and Edward Grefenstette. 2014. A type-driven tensor-based semantics for CCG. In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pages 46–54. Gothenburg, Sweden: Association for Computational Linguistics. doi:10.3115/v1/W14-1406.

Meyer, Francois and Martha Lewis. 2020. Modelling Lexical Ambiguity with Density Matrices. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 276–290. Online: Association for Computational Linguistics. doi:10.18653/v1/2020.conll-1.21.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Miller, George A. 1995. WordNet: a lexical database for English. *Communications of the ACM* 38 (11): 39–41.

Mitchell, Jeff and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science* 34 (8): 1388–1429.

Moortgat, Michael, Mehrnoosh Sadrzadeh, and Gijs Wijnholds. 2020. A Frobenius algebraic analysis for parasitic gaps. *Journal of Applied Logics* 7 (5): 823–854.

Moortgat, Michael and Gijs Wijnholds. 2017. Lexical and derivational meaning in vector-based models of relativisation. In *Proceedings of the 21st Amsterdam Colloquium*, pages 55–64. ILLC, University of Amsterdam.

Murphy, M Lynne. 2003. *Semantic relations and the lexicon: Antonymy, synonymy*

and other paradigms. Cambridge University Press.

Muskens, Reinhard and Mehrnoosh Sadrzadeh. 2016. Context update for lambdas and vectors. In *International Conference on Logical Aspects of Computational Linguistics*, pages 247–254. Springer.

Nguyen, Kim Anh, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017. Hierarchical embeddings for hypernymy detection and directionality. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 233–243. Copenhagen, Denmark: Association for Computational Linguistics. doi:10.18653/v1/D17-1022.

Nickel, Maximillian and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, volume 30, pages 6338–6347.

Nielsen, Michael A. and Isaac L. Chuang. 2010. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511976667.

Paperno, Denis, Nghia The Pham, and Marco Baroni. 2014. A practical and linguistically-motivated approach to compositional distributional semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 90–99. Baltimore, Maryland: Association for Computational Linguistics. doi:10.3115/v1/P14-1009.

Paulsen, Vern. 2002. *Completely bounded maps and operator algebras*, volume 78 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Doha, Qatar: Association for Computational Linguistics. doi:10.3115/v1/D14-1162.

Piedeleu, Robin, Dimitri Kartsaklis, Bob Coecke, and Mehrnoosh Sadrzadeh. 2015. Open system categorical quantum semantics in natural language processing. In *6th Conference on Algebra and Coalgebra in Computer Science (CALCO 2015)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Preller, Anne and Mehrnoosh Sadrzadeh. 2011. Semantic vector models and functional models for pregroup grammars. *Journal of Logic, Language and Information* 20 (4): 419–443.

Reimers, Nils, Iryna Gurevych, Nils Reimers, Iryna Gurevych, Nandan Thakur, Nils Reimers, Johannes Daxenberger, Iryna Gurevych, Nils Reimers, Iryna Gurevych, et al. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Rimell, Laura. 2014. Distributional lexical entailment by topic coherence. In *Pro-*

ceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 511–519. Gothenburg, Sweden: Association for Computational Linguistics. doi:10.3115/v1/E14-1054.

Roller, Stephen, Douwe Kiela, and Maximilian Nickel. 2018. Hearst patterns revisited: Automatic hypernym detection from large text corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 358–363. Melbourne, Australia: Association for Computational Linguistics. doi:10.18653/v1/P18-2057.

Sadrzadeh, Mehrnoosh, Stephen Clark, and Bob Coecke. 2013. The Frobenius anatomy of word meanings I: subject and object relative pronouns. *Journal of Logic and Computation* 23 (6): 1293–1317.

Sadrzadeh, Mehrnoosh, Dimitri Kartsaklis, and Esma Balkır. 2018. Sentence entailment in compositional distributional semantics. *Annals of Mathematics and Artificial Intelligence* 82 (4): 189–218.

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642. Seattle, Washington, USA: Association for Computational Linguistics.

Vulić, Ivan and Nikola Mrkšić. 2018. Specialising word vectors for lexical entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1134–1145. New Orleans, Louisiana: Association for Computational Linguistics. doi:10.18653/v1/N18-1103.

Weeds, Julie, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1015–1021. Geneva, Switzerland: COLING.

van de Wetering, John. 2017. Ordering information on distributions. *arXiv:1701.06924* .

Wijnholds, Gijs. 2015. *Categorical foundations for extended compositional distributional models of meaning*. Master's thesis, ILLC, University of Amsterdam.

Wijnholds, Gijs, Mehrnoosh Sadrzadeh, and Stephen Clark. 2020. Representation Learning for Type-Driven Composition. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 313–324. Online: Association for Computational Linguistics. doi:10.18653/v1/2020.conll-1.24.

Williams, Adina, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. New Orleans, Louisiana: Association for Computa-

tional Linguistics. doi:10.18653/v1/N18-1101.

Yanaka, Hitomi, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019. HELP: A Dataset for Identifying Shortcomings of Neural Models in Monotonicity Reasoning. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 250–255. Minneapolis, Minnesota: Association for Computational Linguistics. doi:10.18653/v1/S19-1027.