Modelling hyponymy for DisCoCat

Martha Lewis

ILLC, University of Amsterdam, Amsterdam, The Netherlands

In natural language processing words are commonly represented as vectors. However, vector representations do not intrinsically incorporate the hierarchical relationships that obtain between many words. We model words as positive operators. These have an ordering which we interpret as modelling hierarchical information. We describe a simple way of building positive operators for words, and give methods for composing these words representations to form phrases and sentences. We test the methods on simple sentence-level entailment datasets.

1 Introduction

Modelling words as vectors has been extremely successful in recent years. Whilst such representations were originally effective in areas such as synonymy and paraphrasing, it is also desirable to model more structure in words, phrases, and sentences. One key task is commonly known as *natural language inference* (NLI) or *recognizing textual entailment*. This kind of task challenges a computational system to infer a relationship of entailment, contradiction, or neither between two texts. In order to make such a judgement, we need to be able to lexically compose words to form phrases and sentences above the word level, and we furthermore need a notion of lexical entailment that interacts nicely with our notion of composition [1]. There has been a wide range of research in this area, from logic-based models such as Bos and Markert [2006] to neural networks [Bowman et al., 2015b] and distributional approaches [Baroni et al., 2012]. Neural network approaches to NLI are very suc-

Martha Lewis: m.a.f.lewis@uva.nl, ,

[1] We will need other aspects as well but these are a minimum

cessful on the datasets for which they are designed, but there is evidence that performance drops when tested on other datasets [Talman and Chatzikyriakidis, 2018, Bernardy and Chatzikyriakidis, 2019]. A model that combines the tensor-based compositional vectors of Coecke et al. [2010] with a theory of entailment known as the distributional inclusion hypothesis (DIH) [Geffet and Dagan, 2005] is examined in Kartsaklis and Sadrzadeh [2016], and forms the baseline on which we will test our models. In Preller [2014], a description of logical aspects of natural language semantics in biproduct dagger categories is given. In that work, the semantics of a sentence is given by a Boolean vector, and describes negation, quantifiers, and discourse-level semantics. In the current paper we will aim for a sentence representation that is richer than this, however, we do not yet have means to implement negation, quantifiers, and discourse.

The theoretical grounding for the current work is given in Bankova et al. [2019], where a particular notion of hyponymy that interacts well with compositionality is described. However, no experimental support is given in that paper. Similar work is carried out in Balkır et al. [2016], and some experiments are undertaken. In this paper, we will build positive operators that represent words. The operators are built using GloVe vectors [Pennington et al., 2014] and information from WordNet [Miller, 1995]. As such, our approach is a hybrid approach, using both distributional and human-curated information. We will use two new measures for graded hyponymy, developed in Lewis [2019a], that provide a wider range of comparisons than the entropy-derived measure developed in Balkır et al. [2016] or the eigenvalue-related measure of Bankova et al. [2019]. We describe a composition method for positive operators, and discuss types of normalization that can be applied to operators, ultimately using a normalization that sets the maximum eigenvalue to 1. We test our models on the compositional dataset of Kartsaklis and Sadrzadeh [2016].

2 Background

There are a number of fast and effective methods for building vectors for individual words, dating back to Salton et al. [1975]. However, as well as deriving word meanings, we also need to give meanings to sentences and phrases. This means that we need some method for composing vector representations of words. There are a number of approaches, ranging from simple vector operations to deep neural network methods. We work within the categorical compositional distributional (DisCoCat) model of Coecke et al. [2010]. This is based in the idea that grammatical formalisms such as pregroup grammar [Lambek, 1999] have the same categorical structure as the category **FVect** whose objects are finite dimensional vector spaces and whose morphisms are linear maps. The word representations used in DisCoCat have a strong theoretical foundation based in formal grammar and semantics. Moreover, DisCoCat is flexible with regard to which grammar and semantic representations it uses.

2.1 DisCoCat

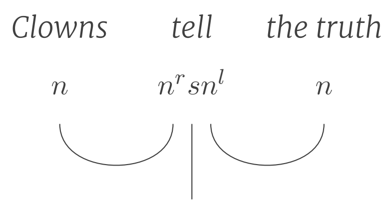
We explain DisCoCat briefly. For more details, see Coecke et al. [2010], Preller and Sadrzadeh [2011]. A grammar for English is represented in a compact closed category. The grammar is then mapped via a strong monoidal functor, as described in [Preller and Sadrzadeh, 2011], to the category **FVect** of finite-dimensional vector spaces and linear maps. The grammar we discuss here is pregroup grammar. It is possible to use other forms of grammar [Coecke et al., 2013] or λ-calculus [Muskens and Sadrzadeh, 2016]. 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:

$$\epsilon_x^r : x \cdot x^r \to 1, \qquad \epsilon_x^l : x^l \cdot x \to 1 \qquad \eta_x^r : 1 \to x^r \cdot x, \qquad \eta_x^l : 1 \to x \cdot x^l \tag{1}$$

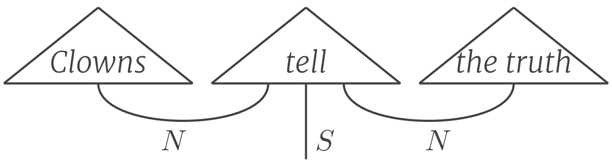
A string of grammatical types $t_1, ...t_n$ is grammatical if it reduces, via the morphisms above, to the sentence type s. For example, typing *clowns* as n, *tell* as $n^r s n^l$ and *the truth* as n, the sentence *Clowns tell the truth* has type $n(n^r s n^l)n$ and is shown to be grammatical as follows:

$$(\epsilon^r \ 1 \ \epsilon^l) n(n^r s n^l) n \to (\epsilon^r \ 1)(n \ n^r s \ 1) \to 1 \ s \ 1 = s \tag{2}$$

The above reduction can be represented graphically as follows:



This grammar is mapped to $\mathbf{FVect}$ by sending the noun type n to a vector space N and the sentence type s to S. The concatenation operation in the grammar is mapped to $\otimes$, i.e., the tensor product of vector spaces. Then the morphisms ϵ_p^r and ϵ_p^l map to tensor contraction, and η_p^r and η_p^l map to identity maps. This implies that intransitive verbs are represented as maps from N to S, or matrices in $N \otimes S$, and that transitive verbs are represented as maps from two copies of N to S, or tensors in $N \otimes S \otimes N$. So, in the example above, *Clowns* is mapped to a vector in N, as is *the truth*, and *tell* is mapped to a tensor in $N \otimes S \otimes N$. The vectors and tensors are concatenated using the tensor product, and tensor contraction is applied to map the sentence down into one sentence vector. Compact closed categories have a nice diagrammatic calculus [Kelly and Laplaza, 1980], for a linguistically couched explanation see Coecke et al. [2010]. In this calculus, the composition of the words *Clowns*, *tell*, and *the truth* into the sentence *Clowns tell the truth* is expressed as follows:



We will use this notation later to describe how to build particular representations of verbs and other function words.

2.2 DisCoCat in $\mathbf{CPM}(\mathbf{FVect})$

In Piedeleu et al. [2015], Bankova et al. [2019], and Balkır et al. [2016] the DisCoCat model is lifted to the category $\mathbf{CPM}(\mathbf{FVect})$, which has the same objects as $\mathbf{FVect}$, but whose morphisms are now completely positive maps. The $\mathbf{CPM}$ construction is introduced in Selinger [2007]. Words are now represented as positive operators rather than as vectors, and maps between them are completely positive maps. A positive operator is defined as follows, using bra-ket notation from physics. For a unit vector $|v\rangle$, the projection operator $|v\rangle\langle v|$ onto the subspace spanned by $|v\rangle$ is called a *pure state*. A positive operator is given by sum of pure states. It is an operator A such that:

1. $\forall v \in V. \langle v|A|v\rangle \geq 0$,
2. A is self-adjoint

If, in addition, A has trace 1, then A encodes a probabilistic mixture of pure states, and is called a density matrix. Relaxing this condition gives us different choices for normalization.

Importantly, $\mathbf{CPM}(\mathbf{FVect})$ is also compact closed, so that the same sort of functorial mapping can be made from the grammar category to the semantics category. Furthermore, the diagrammatic calculus can also be used in this context.

2.3 Ordering positive operators

The set of positive operators on a vector space has an ordering introduced by Löwner [1934]. For positive operators A and B, we define:

$$A \sqsubseteq B \iff B - A \text{ is positive}$$

In DisCoCat, we interpret this ordering as an hyponymy relation. If we have a positive operator $[\![mammal]\!]$ representing the word *mammal*, and a positive operator $[\![dog]\!]$ representing the word *dog*, then we would like to see:

$$[\![dog]\!] \sqsubseteq [\![mammal]\!]$$

In Bankova et al. [2019] the authors introduce a notion of graded hyponymy. Consider the relationship between *dog* and *pet*. Not all dogs are pets: some are working dogs and some are wild. We therefore want to say that $[\![dog]\!] \sqsubseteq [\![pet]\!]$ up to some value $k \in [0, 1]$. The grading is introduced by considering an error term defined as follows. Suppose that $A \sqsubseteq B$. Then $B - A = D$, i.e. $A + D = B$, where D is some positive operator. However, it may be the case that this does not hold. If not, it is possible to add in some error term E so that $A \sqsubseteq B + E$. This is viewed as saying that A entails B up to the error term E. Combining definitions, $A + D = B + E$, and so trivially A entails B up to the error term A, meaning that we can get any word A to entail another B by adding in an error term that is A itself. We may then consider the size of the error term E, and we would like to find the smallest such error term.

In Bankova et al. [2019], the error term was of the form $(1 - k)A$ and the scalar $k \in [0, 1]$ gave a graded notion of hyponymy. The effect of this scalar is to reduce the size of A until it 'fits inside' B, giving a notion of graded hyponymy that says that A is a k–hyponym of B, $A \sqsubseteq_k B$ if $B - kA$ is positive. So, if k is equal to 1, the size of the error term is zero, meaning that we have full hyponymy. If k is zero, the only hyponymy that can be induced is the trivial step of adding the whole of A in as an error term.

3 Methods

3.1 Measuring hyponymy

One of the drawbacks of the measure of graded entailment given in Bankova et al. [2019] is that if the space spanned by eigenvectors of A, called $Span(A)$, is not a subspace of $Span(B)$, then the value of k must be 0. However, it may be the case that although $Span(A)$ is not a subspace of $Span(B)$, the value of the operator on the part not included in $Span(B)$ is small. It would therefore be useful to be able to include a wider range of gradings. In Lewis [2019a] we introduce two new measures which allow us to assign non-zero gradings to these cases. We consider the error term E and calculate it as follows. If $B-A$ is not positive, it is possible to make it positive by adding in a positive operator constructed in the following manner. Firstly diagonalize $B-A$, resulting in a real-valued matrix, since $B-A$ is real symmetric. Construct a matrix E by setting all positive eigenvalues of $B-A$ to 0 and changing the sign of all negative eigenvalues. Then $B-A+E$ will give us a positive matrix. This E is our error term. In the best case, $E=0$, meaning that A is a full hyponym of E, and in the worst case, $E=A$, meaning that A does not have any overlap with B. We propose two different measures related to this error term that give us values in $\mathbb{R}$, giving a grading for hyponymy.

The first measure is

$$k_{BA} = \frac{\sum_i \lambda_i}{\sum_i |\lambda_i|} \tag{3}$$

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 $k_{BA}=-k_{AB}$.

Secondly, we propose

$$k_E = 1 - \frac{||E||}{||A||} \tag{4}$$

where $||\cdot||$ denotes the Frobenius norm. This measures the size of the error term as a proportion of the size of A. Since $A=E$ in the worst case, this measure ranges from 0 when $E=A$ to 1 when $E=0$.

In Lewis [2019a] we describe methods for building positive operators, following the approach outlined in Bankova et al. [2019]. In that work, the authors observe that each word vector has a corresponding pure matrix:

$$|cat\rangle \mapsto |cat\rangle\langle cat|$$

Words which are more general can then be built up by summing over the projectors corresponding to the hyponyms of that word. For example, the meaning of the word *pet* can be thought of as represented by:

$$[\![pet]\!] = p_d \; |dog\rangle\langle dog| + p_c \; |cat\rangle\langle cat| + p_t \; |tarantula\rangle\langle tarantula| + ...$$
$$\text{where} \quad \forall i.p_i \geq 0$$

In general, the meaning of a word w is considered to be given by a collection of unit vectors $\{|w_i\rangle\}_i$, where each $|w_i\rangle$ represents an instance of the concept expressed by the word. Then the operator:

$$[\![w]\!] = \sum_i p_i \; |w_i\rangle\langle w_i| \in W \otimes W \tag{5}$$

represents the word w. The p_i are weightings derived from the text, and there are various choices about what these should be, which we discuss in section 3.3.

We build representations of words as positive operators in the following manner. Suppose we have a dictionary of word vectors $\{v_i : |v_i\rangle \in W\}_i$ derived from a corpus using standard distributional or embedding techniques, for example GloVe, Pennington et al. [2014], FastText Bojanowski et al. [2017], or weighted co-occurrence vectors. To build a representation of a word, we obtain a set of hyponyms that are instances of that word. In this paper, we use WordNet Miller [1995], a human-curated database of word relationships including hyponym-hypernym pairs. The WordNet hyponymy relationship is naturally arranged as a directed graph with a root (it is not quite a tree). For the noun subset of the database, the root is the most general noun *entity*, and the leaves are specific nouns. For example, under the word *rocket* there are (inter alia): *test_instrument_vehicle*, *Stinger*, *takeoff_booster*, *arugula*. Notice that here we have different meanings of the word *rocket*, one as a projectile and one as a vegetable. There are also less supervised ways of obtaining these relationships using patterns derived from text, see Hearst [1992], Roller et al. [2018] for examples.

To build a positive operator for a word w, we go through the WordNet hierarchy and collect all hyponyms w_i of w at all levels. We then form $[\![w]\!]$ as in equation (5), with $p_i = 1$ for all i. When we build these operators, between 1/3 and 1/2 of the hyponyms listed in WordNet are available in GloVe, and we therefore miss a large proportion of the information included in WordNet.

3.3 Normalization

An important parameter choice is the type of normalization to use. In Bankova et al. [2019] two choices are discussed: normalizing operators to trace 1, or normalizing operators to have maximum eigenvalue less than or equal to 1. The properties of these two normalization strategies are thoroughly analyzed in van de Wetering [2017]. If operators are normalized to trace 1, then the crisp Löwner ordering becomes trivial: no two operators stand in the relation $A \sqsubseteq B$. If operators are normalized to have maximum eigenvalue 1, then the Löwner ordering has particularly nice properties. In previous work [Lewis, 2019a] we have shown that good results on lexical entailment datasets can be obtained using no normalization at all. However, applying a maximum eigenvalue normalization means that further operations like applying negation are likely to become easier, and hence in this paper we investigate how well our models can do with normalization.

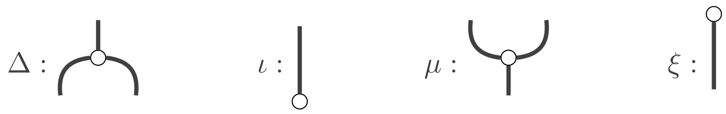
3.4 Composing positive operators

One of the strengths of DisCoCat is its formal approach to composition. Within the category $\mathbf{CPM}(\mathbf{FVect})$ objects are finite-dimensional vector spaces and morphisms are completely positive maps. In Lewis [2019a] we examined two composition methods build using a type-lifting approach, which we describe below (**Mult** and **BMult**). Here we also investigate another way of forming a completely positive map by building a Kraus operator associated with a given positive operator (**KMult**).

In order to build these maps we use the type-lifting methods outlined in Kartsaklis et al. [2012]. A Frobenius algebra over a finite-dimensional vector space with bases $\{\overrightarrow{n}_i\}_i$ is given by

$$\Delta :: \overrightarrow{n}_i \mapsto \overrightarrow{n}_i \otimes \overrightarrow{n}_i \qquad \iota :: \overrightarrow{n}_i \mapsto 1 \qquad \mu :: \overrightarrow{n}_i \otimes \overrightarrow{n}_i \mapsto \overrightarrow{n}_i \qquad \xi :: 1 \mapsto \overrightarrow{n}_i$$

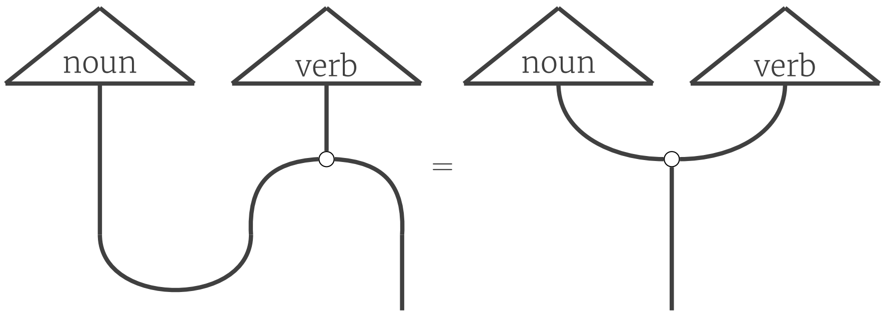
In the graphical calculus, these are given by:



A vector $|v\rangle \in W$ can be lifted to a higher-order representation in $W \otimes W$ by applying the map Δ. In **FVect**, this higher-order representation takes the vector $|v\rangle$ and embeds it along the diagonal of a matrix in $W \otimes W$. So, for example, given a vector representation of an intransitive verb $|run\rangle \in W$, we can lift that representation to a matrix in $W \otimes W$ by embedding it into the diagonal of a matrix. The Frobenius algebra interacts with the type reduction morphism ϵ_N in such a way that the result of lifting a verb and then composing with a noun is to apply the μ multiplication to the tensor product of the noun and the verb vectors, i.e.

$$(\epsilon_N \otimes 1_N) \circ (1_N \otimes \Delta_N)(|noun\rangle \otimes |verb\rangle) = \mu(|noun\rangle \otimes |verb\rangle)$$

Diagrammatically,



In **FVect** the multiplication μ implements pointwise multiplication of the two vectors. However in **CPM**(**FVect**) we have different choices for the multiplication μ. One is composition of the two operators. This results in a matrix that is no longer self-adjoint, and so Piedeleu [2014] suggests using the non-commutative and non-associative operator $\rho_2^{\frac{1}{2}}\rho_1\rho_2^{\frac{1}{2}}$ in its place. Piedeleu [2014] also notes that the pointwise multiplication of two positive operators is a completely positive map, giving us another choice for composition.

Following Kartsaklis et al. [2012], this gives us a method for building verb operators from their lower-level operators. Firstly, we assume the noun space $N \otimes N$ to be equal to the sentence space $S \otimes S$, and refer to these both as $W \otimes W$. Given a representation of an intransitive verb $[\![verb]\!] \in W \otimes W$, we lift it to $\Delta([\![verb]\!]) \in W \otimes W \otimes W \otimes W$. Composing with a noun implements $[\![noun\ verb]\!] = \mu([\![noun]\!] \otimes [\![verb]\!])$.

Lastly, we can form a completely positive map from a positive matrix A by decomposing A into a weighted sum of orthogonal projectors $A = \sum_i p_i P_i$, and then forming the map

$$\mathcal{A}(-) = \sum_i p_i P_i \circ - \circ P_i$$

The same proposal for composition is given in Coecke [2019].

For intransitive verbs we combine the noun and the verb via three operations **Mult**, **BMult**, **KMult**.

$$\text{Mult: } [\![\textit{noun verb}]\!] = [\![\textit{verb}]\!]([\![\textit{noun}]\!]) = [\![\textit{noun}]\!] \odot [\![\textit{verb}]\!] \tag{6}$$

$$\text{BMult: } [\![\textit{noun verb}]\!] = [\![\textit{verb}]\!]([\![\textit{noun}]\!]) = [\![\textit{verb}]\!]^{\frac{1}{2}}[\![\textit{noun}]\!][\![\textit{verb}]\!]^{\frac{1}{2}} \tag{7}$$

$$\text{KMult: } [\![\textit{noun verb}]\!] = [\![\textit{verb}]\!]([\![\textit{noun}]\!]) = \sum_i p_i P_i [\![\textit{noun}]\!] P_i \tag{8}$$

where in KMult $[\![\textit{verb}]\!] = \sum_i p_i P_i$. We also investigate switched versions of **BMult** and **KMult**, where the order of composition is switched.

For transitive verbs there is one possibility for pointwise multiplication of the operators, since this is both commutative and associative. For **BMult** and **KMult** there are a number of composition orders. We will concentrate on two which reflect the difference between viewing verb as operator and viewing nouns as operator. Both compose verb and object, then verb phrase and subject. We therefore have:

$$\text{Mult: } [\![\textit{subj verb obj}]\!] = [\![\textit{subj}]\!] \odot [\![\textit{verb}]\!] \odot [\![\textit{obj}]\!] \tag{9}$$

$$\text{BMult-V: } [\![\textit{subj verb obj}]\!] = [\![\textit{vp}]\!]^{\frac{1}{2}}[\![\textit{subj}]\!][\![\textit{vp}]\!]^{\frac{1}{2}} \text{ where } [\![\textit{vp}]\!] = [\![\textit{verb}]\!]([\![\textit{obj}]\!]) \tag{10}$$

$$\text{KMult-V: } [\![\textit{subj verb obj}]\!] = \sum_i p_i P_i [\![\textit{subj}]\!] P_i \text{ where } \sum_i p_i P_i = [\![\textit{verb}]\!]([\![\textit{obj}]\!]) \tag{11}$$

$$\text{BMult-N: } [\![\textit{subj verb obj}]\!] = [\![\textit{subj}]\!]^{\frac{1}{2}}[\![\textit{vp}]\!][\![\textit{subj}]\!]^{\frac{1}{2}} \text{ where } [\![\textit{vp}]\!] = [\![\textit{obj}]\!]([\![\textit{verb}]\!]) \tag{12}$$

$$\text{BMult-N: } [\![\textit{subj verb obj}]\!] = \sum_i p_i P_i [\![\textit{vp}]\!] P_i \text{ where } \sum_i p_i P_i = [\![\textit{subj}]\!] \tag{13}$$

The notation $[\![A]\!]([\![B]\!])$ refers to the corresponding two-place operation, either **KMult** or **BMult**.

4 Experimental setting

We test our word representations and composition methods on the compositional datasets of Sadrzadeh et al. [2018]. This is a series of three datasets, covering simple intransitive sentences, transitive sentences, and verb phrases.

The intransitive verb dataset consists of paired sentences consisting of a subject and a verb. In half the cases the first sentence entails the second, and in the other half of cases, the order of the sentences is reversed. For example:

summer finish, season end, T

season end, summer finish, F

The first sentence is marked as entailing, whereas the second is marked as not entailing. The dataset is created by selecting nouns and verbs from WordNet that stand in the correct relationship. The transitive verb and verb phrase datasets are similarly created.

To test our models, we build the basic word representations as in equation (5). We then use the compositional methods outlined in section 3.4 to create the sentence representations. We calculate the graded entailment value between the composed sentence representations. In previous literature [Kartsaklis and Sadrzadeh, 2016], area under receiver operating characteristic (ROC) curve was reported. For comparison purposes, we calculate the same quantity. To test for significance of our results, we bootstrap the data with 100 repetitions [Efron, 1992] and use a one-sample t-test to compare with the results given in Kartsaklis and Sadrzadeh [2016]. To compare between models we use a two sample t-test, and in each case we apply the Bonferroni correction to compensate for multiple comparisons. We use GloVe vectors in 300 dimensions. The basic operators we build are normalised to have maximum eigenvalue 1. We want to retain this property. The **BMult**, **KMult** operators and their variants preserve this property by Weyl's inequalities [Weyl, 1912] and the orthogonality of projectors in **KMult**. For the other operators, if the maximum eigenvalue of the composed expression is greater than 1, we normalize, else we leave it as is.

5 Results

On the KS2016 compositionality datasets results are reported in terms of area under ROC curve (Table 1). Overall, the k_{BA} measure works best with the composition operators, with every operator outperforming the previous best results on this dataset. Across both measures, the **Mult** operator performs particularly well, and the **KMult** also perform strongly. Interestingly, the **KMult** operators perform best when there is more structure in the phrase, indicating that per-

Table 1: Area under ROC curve on the KS2016 datasets. For the SV and VO datasets, BMult1 and KMult1 refer to the models described in equations (7) and (8). BMult2 and KMult2 refer the variants formed by switching the order of composition. For SVO, BMult1 and KMult1 refer to the models described in equations (10) and (11) and BMult2 and KMult2 refer to the models described in equation (12) and (13). A * indicates that the value is significantly higher than the baseline from Kartsaklis and Sadrzadeh [2016] ($p < 0.01$). A + indicates that the value is not significantly lower than the Mult model ($p < 0.05$).

	k_E measure			k_{BA} measure		
Model	SV	VO	SVO	SV	VO	SVO
KS2016 best	0.84	0.82	0.86	0.84	0.82	0.86
Verb only	0.632	0.632	0.663	0.868*	0.829*	0.890*
Addition	0.576	0.586	0.492	0.893*	0.892*	0.945*
Mult	0.885*	0.842*	0.966*	0.961*	0.934*	0.980*
BMult1	0.794	0.749	0.880*	0.945*	0.916*	0.977*
BMult2	0.778	0.723	0.869	0.949*	0.914*	0.980*+
KMult1	0.881*	0.833*	0.946*	0.957*+	0.934*+	0.984*+
KMult2	0.823	0.800	0.930*	0.909*	0.939*+	0.963*

haps this operation will start to outperform the simpler **Mult** in more complicated situations.

The good performance of our models is likely to be due to the fact that both the dataset and our word representations were constructed from WordNet, and hence the high performance is to be expected. However, it is still interesting that our representations work so well with the compositional operations.

6 Discussion and further work

We have suggested a mechanism for building the positive operators needed for the theory presented in Bankova et al. [2019], together with novel measures of graded hyponymy. The representations and the measures we have developed perform competitively on phrase and sentence datasets. The type of representation we have developed is a hybrid representation in the sense that we use off-the-shelf distributional vectors, but also human-provided information from WordNet. The representations are extremely quick to build, with no training time.

The datasets we have so far tested on are relatively small, and therefore testing on larger datasets such as the Stanford Natural Language Inference

(SNLI) dataset [Bowman et al., 2015a] is an important next step. Since the representations we build are not tuned to a particular dataset, the problems pointed out by Talman and Chatzikyriakidis [2018] will hopefully be lessened.

It is constructive to consider what the representations we build consist of. The modelling of words as positive operators is in a way referential, since we are forming the meaning of a word by summing over the things that fall under it as a concept. Since we obtain this information from WordNet, we cannot of course literally sum over representations of the individual objects in the world to which the word refers. The instances obtained from WordNet serve as a proxy for the objects referred to. However, if we were to implement this model as gathering data from some embodied or simulated environment, we could then view the representation of a word as summing over individual representations that an agent encounters.

Taking the information from WordNet also utilises the sense of individual words, however, since these vectors are themselves a summary representation of parts of the concept. Modelling words as positive operators allows us to incorporate multiple senses of a word, ranging from slight variances in meaning (*booster_rocket*, *space_rocket*) to full lexical ambiguity (*arugula*). The difference between the modelling of combinations of senses and combinations of referents can in fact be both separated out and reincorporated into one shared representation, within a second iteration of the $\mathbf{CPM}()$ construction, done in Ashoush and Coecke [2016]. Finding a means of constructing these representations so that the dimensionality is not prohibitive would be a useful extension.

The ordering we impose on word, phrase and sentence representations forms a hierarchy, and the individual word representations have the nice property that their similarity can be compared. This suggests further research into the notion of co-hyponymy – the relationship that obtains between two words that share a hypernym, such as *dog* and *cow*, which share the hypernym *mammal*.

Exploring the ordering of composition will be an important area of further work – it is unclear what it means to interpret the nouns as operators rather than the verbs. One line of inquiry could be to look into understanding this via the type-raising operations of categorial grammar.

Similarities to our approach can be found in the notion of words being represented as Gaussians [Jameel and Schockaert, 2017, Vilnis and McCallum, 2014]. The positive operators we build have the same structure as covariance matrices and, if appropriately normalized, are interpreted as representing a probability distribution over vectors. Exploring these connections is an area of further work.

Finally, a crucial extension to this whole approach is to be able to model hyponymy, composition, and their interaction in downwardly monotone contexts, using the natural logic introduced in Barwise and Cooper [1981], MacCartney and Manning [2007]. Whilst the representations we have built have a mechanism for lexical composition, i.e. a means to obtain the sense of a phrase or sentence from the senses of the words themselves, what we do not so far have is a notion of logical composition, assertion, or truth. Work is currently in progress to develop a model for negation, see Lewis [2019b]. This is an area of ongoing research.

References

Daniela Ashoush and Bob Coecke. Dual density operators and natural language meaning. In *Proceedings of the 2016 Workshop on Semantic Spaces at the Intersection of NLP, Physics and Cognitive Science, SLPCS@QPL 2016, Glasgow, Scotland, 11th June 2016.*, pages 1–10, 2016. DOI: 10.4204/EPTCS.221.1. URL `https://doi.org/10.4204/EPTCS.221.1`.

Esma Balkır, Mehrnoosh Sadrzadeh, and Bob Coecke. Distributional sentence entailment using density matrices. In *Topics in Theoretical Computer Science*, pages 1–22. Springer, 2016.

Dea Bankova, Bob Coecke, Martha Lewis, and Dan Marsden. Graded hyponymy for compositional distributional semantics. *Journal of Language Modelling*, 6 (2):225–260, 2019.

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-Chieh Shan. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the ACL*, pages 23–32, 2012.

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

Jean-Philippe Bernardy and Stergios Chatzikyriakidis. What kind of natural language inference are nlp systems learning: Is this enough? 2019.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

Johan Bos and Katja Markert. When logical inference helps determining textual entailment (and when it doesn't). In *Proceedings of the Second PASCAL RTE Challenge*, page 26, 2006.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015a.

Samuel R Bowman, Christopher Potts, and Christopher D Manning. Recursive neural networks can learn logical semantics. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 12–21, 2015b.

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

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. *arXiv:1003.4394*, 2010.

Bob Coecke, Edward Grefenstette, and Mehrnoosh Sadrzadeh. Lambek vs. Lambek: Functorial vector space semantics and string diagrams for Lambek calculus. *Annals of Pure and Applied Logic*, 164(11):1079–1100, 2013.

Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*, pages 569–593. Springer, 1992.

Maayan Geffet and Ido Dagan. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting on ACL*, pages 107–114. ACL, 2005.

Marti A Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics, 1992.

Shoaib Jameel and Steven Schockaert. Modeling context words as regions: An ordinal regression approach to word embedding. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 123–133, 2017.

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

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

M. Kelly and M.L. Laplaza. Coherence for compact closed categories. *Journal of Pure and Applied Algebra*, 19:193–213, 1980.

Joachim Lambek. Type grammar revisited. In *Logical aspects of computational linguistics*, pages 1–27. Springer, 1999.

M. Lewis. Compositional hyponymy with positive operators, 2019a. to appear at RANLP 2019.

M. Lewis. Towards negation in DisCoCat, 2019b. to appear at SemSpace 2019.

Karl Löwner. Über monotone matrixfunktionen. *Mathematische Zeitschrift*, 38 (1):177–216, 1934.

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

George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38 (11):39–41, November 1995. ISSN 0001-0782. DOI: 10.1145/219717.219748. URL http://doi.acm.org/10.1145/219717.219748.

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

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

Robin Piedeleu. Ambiguity in categorical models of meaning. Master's thesis, University of Oxford, 2014.

Robin Piedeleu, Dimitri Kartsaklis, Bob Coecke, and Mehrnoosh Sadrzadeh. Open system categorical quantum semantics in natural language processing. In *6th Conference on Algebra and Coalgebra in Computer Science, CALCO 2015, June 24-26, 2015, Nijmegen, The Netherlands*, pages 270–289, 2015. DOI: 10.4230/LIPIcs.CALCO.2015.270. URL https://doi.org/10.4230/LIPIcs.CALCO.2015.270.

Anne Preller. Natural language semantics in biproduct dagger categories. *Journal of Applied Logic*, 12(1):88–108, 2014.

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

Stephen Roller, Douwe Kiela, and Maximilian Nickel. Hearst patterns revisited: Automatic hypernym detection from large text corpora. *arXiv preprint arXiv:1806.03191*, 2018.

Mehrnoosh Sadrzadeh, Dimitri Kartsaklis, and Esma Balkir. Sentence entailment in compositional distributional semantics. *Ann. Math. Artif. Intell.*, 82(4):189–218, 2018. DOI: 10.1007/s10472-017-9570-x. URL https://doi.org/10.1007/s10472-017-9570-x.

G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975. ISSN 0001-0782. DOI: 10.1145/361219.361220. URL http://doi.acm.org/10.1145/361219.361220.

Peter Selinger. Dagger compact closed categories and completely positive maps. *Electronic Notes in Theoretical Computer Science*, 170:139–163, 2007.

Aarne Talman and Stergios Chatzikyriakidis. Neural network models for natural language inference fail to capture the semantics of inference. In *aProceedings of the BlackboxNLP workshop, ACL2019*, 2018.

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

Luke Vilnis and Andrew McCallum. Word representations via gaussian embedding. *arXiv preprint arXiv:1412.6623*, 2014.

Hermann Weyl. Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung). *Mathematische Annalen*, 71(4):441–479, 1912.