OPEN ACCESS

University of BRISTOL

Lewis, M. (2020). Towards logical negation for compositional distributional semantics. *IfCoLoG Journal of Logics and their Applications*, *7*(3).

Link to publication record in Explore Bristol Research
PDF-document

University of Bristol - Explore Bristol Research
General rights

Towards logical negation in compositional distributional semantics

Martha Lewis
ILLC, University of Amsterdam
`m.a.f.lewis@uva.nl`

Abstract

The categorical compositional distributional model of meaning gives the composition of words into phrases and sentences pride of place. However, it has so far lacked a model of logical negation. This paper gives some steps towards providing this operator, modelling it as a version of projection onto the subspace orthogonal to a word. We give a small demonstration of the operators performance in a sentence entailment task.

1 Introduction

Compositional models of meaning aim to represent the meaning of phrases and sentences by combining representations of the words in the sentence according to some rule. Compositional distributional models, such as described in Baroni and Zamparelli [2010], Coecke et al. [2010], Paperno et al. [2014] combine the compositional approach with vector-based models of word meaning. In these models, nouns are represented as vectors, and function words, such as verbs and adjectives, are modelled as linear maps. In this paper, we use the *categorical compositional distributional (DisCoCat)* model introduced in Coecke et al. [2010]. This model formalises the compositional approach to language using category theory, setting up a functorial mapping between the grammar of the language on the one hand, and the structures used to represent lexical meaning on the other. In modelling the meaning of words and sentences, a distinction can be made between words with lexical content, and words that can arguably be modelled as an operation on the structure of the sentence. For example, in Sadrzadeh et al. [2013], relative pronouns are modelled as routing information around a sentence using the structure of a Frobenius algebra. In Kartsaklis [2016], conjunctions are modelled using Frobenius algebras.

In the current paper we model negation as an operation on words. In Coecke et al. [2010], negation is modelled as a linear map on a two-dimensional sentence space

which sends each basis vector to the subspace orthogonal to it. This idea of modelling negation as projection to the orthogonal subspace was used in Widdows and Peters [2003], but at the vector level is somewhat unsatisfactory since a word and its negation are then of two different kinds. Furthermore within DisCoCat, words should be modelled as linear maps, which projection onto an orthogonal subspace doesn't satisfy.

Within the categorical compositional framework, we can be flexible about how to represent word meanings. In Coecke et al. [2010] the category **FVect** of vector spaces and linear maps was used, meaning that nouns and sentences are modelled as vectors, and functional words such as verbs and adjectives are modelled as linear maps. In Bolt et al. [2019] the category **ConvexRel** was used, enabling the representation of nouns and sentences as convex sets and function words as convex relations. In this paper we will use the category **CPM**(**FVect**) which models nouns and sentences as positive operators, and function words as completely positive maps. This approach to meaning was developed in Balkır et al. [2016], Bankova et al. [2019] and implemented in Lewis [2019a,b]. We model negation as an operation related to projection onto the orthogonal subspace. We discuss how negation interacts with composition, and we provide a small corpus-based implementation to illustrate the ideas.

1.1   Related work

As mentioned, the idea of negation as projection onto the orthogonal subspace has been implemented in Widdows and Peters [2003] and discussed in Coecke et al. [2010]. However, it has also been argued that negation should not be viewed in this way: rather, that the negation of a word should be fairly similar to the original. for example, Hermann et al. [2013] argue that 'not red' is still a colour, and provide a model where the vector is divided into domains and only part of the vector is inverted. Similarly, Rimell et al. [2017] view negation as antonymy, and provide a model of negation in which an encoder is trained to produce the antonym of a given adjective. Continuing the discussion of the distinction between conversational negation and logical negation, Kruszewski et al. [2016] provide an in depth analysis of the ways in which people use negation in conversation.

The kind of negation that we discuss in this paper is more akin to logical negation, and this will be exemplified by its interaction with entailment between sentences.

2 Background

2.1 Categorical compositional approaches to meaning

The categorical compositional model of meaning uses the framework of category theory to set up a mapping between the grammar of a language and the structures used to represent the manings of individual words. A formalization of grammar is chosen, and represented as a category, called the *grammar category*. A choice is made about the type of meaning representation, which again is formalized as a category, called the *meaning category*. The meaning category and the grammar category are chosen to have the same abstract structure. Type reductions in the grammar category are then functorially mapped to operations in the semantics category. In this paper, the grammar category and the meaning category are both *compact closed*. For details of what this means within the context of linguistics, see Coecke et al. [2010] or Preller and Sadrzadeh [2011]. A gentle presentation is also given in Bolt et al. [2019].

Pregroup grammar In this paper, we will use pregroup grammar, although the formalism is flexible about what can be used, and other choices are given in, for example, Coecke et al. [2013], Maillard et al. [2014], Muskens and Sadrzadeh [2016]. A pregroup is a partially ordered monoid $(X, \cdot, 1, \leq)$ where each $x \in X$ has a left and a right adjoint $(-)^l$, $(-)^r$ such that:

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

A pregroup grammar is the pregroup freely generated over a set of chosen types. We consider the set containing n for noun and s for sentence. Complex types are built up by concatenation of types, and we often leave out the dot so that $xy = x \cdot y$. If $x \leq y$ we say that x *reduces to* y.

A string of 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:

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

$$\leq 1 \ s \ 1 = s \tag{3}$$

The above reduction can be represented graphically as follows:

$$Clowns \quad tell \quad the\ truth$$

$$n \quad n^r s n^l \quad n$$
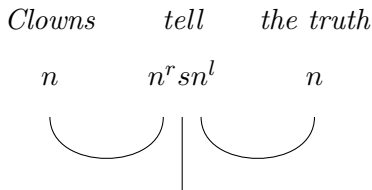
Meaning categories As a first example we describe how pregroup grammar is mapped to **FVect**, the category of vector spaces and linear transformations. The noun type n is mapped to a vector space N and the sentence type s to S. The concatenation operation in the grammar is mapped to $\otimes$, i.e., the tensor product of vector spaces. Then the morphisms ϵ_x^r and ϵ_x^l map to tensor contraction, and η_x^r and η_x^l map to identity matrices.

Function words like verbs and adjectives are modelled as (multi)linear maps. Intransitive verbs are represented as maps from N to S, or matrices in $N \otimes S$, and 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, described in Selinger [2010], or 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:
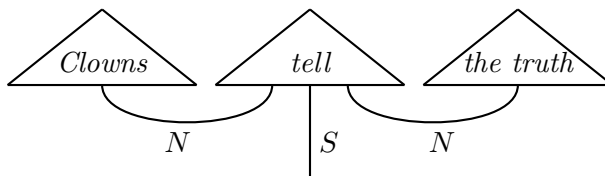
We will use this notation later to describe how to build particular representations of verbs and other function words.

2.2 Modelling words as positive operators

In Piedeleu et al. [2015], Bankova et al. [2019], and Balkır et al. [2016] the DisCoCat model is instantiated with the meaning category **CPM**(**FVect**). This has the same objects as **FVect**, but the morphisms are now completely positive maps. The **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.

Completely positive maps are linear maps that preserve positivity of operators and do so for any trivial extension.

We give an informal description of how pregroup grammar maps into the category **CPM**(**FVect**). For more details see Piedeleu et al. [2015], Bankova et al. [2019], or Balkır et al. [2016]. Within **CPM**(**FVect**), the objects are vector spaces, and morphisms are completely positive maps. The underlying spaces that we represent nouns, sentences, and other words in are now doubled up, meaning that a noun is a positive operator $N \to N$, or a positive semidefinite matrix in $N^* \otimes N$. Morphisms are completely positive maps. These are defined in Selinger [2007] as a morphism $\phi : A^* \otimes A \to B^* \otimes B$ such that there exists an object C in the underlying category, in our case **FVect**, and a morphism $k : C \otimes A \to B$ such that:

$$\phi = (k_* \otimes k) \circ (1_{A^*} \otimes \eta_C \otimes 1_A)$$

Importantly, **CPM**(**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.

Positive operators were proposed in Balkır et al. [2016], Bankova et al. [2019] as a means of representing word meanings since they have a natural ordering called the Löwner ordering. This ordering states that for two positive operators A and B,

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

This ordering can be used to represent hyponymy and lexical entailment. In Balkır et al. [2016], Lewis [2019b] concrete proposals for building positive operators representing words are given.

The space of positive operators and the properties of the Löwner ordering on this space has been examined in D'Hondt and Panangaden [2006], van de Wetering

[2016]. When the set of positive operators is restricted to those with maximum eigenvalue less than or equal to 1, the ordering has nice properties. We restrict to this set, and use the notation $\mathcal{CP}_1(V)$. When the set of positive operators is restricted to those with eigenvalues exactly 1, we have the projectors, and the Löwner ordering corresponds to subspace inclusion on projection operators.

The Löwner ordering is crisp: either the relation obtains or it doesn't. However, when considering natural language, we are also interested in graded notions of hyponymy and entailment. For example, although we may consider *dog* to be highly indicative of *pet*, not every dog is a pet, and so we want some kind of graded ordering. On the other hand, we would expect *dog* to be a full hyponym of *mammal*

Balkır et al. [2016] introduce a graded notion of hyponymy based on the relative entropy of two operators. Bankova et al. [2019] use a graded notion of hyponymy that is based on expanding the hypernym (the broader term) to include the hyponym. Lewis [2019b] extends this idea to include a wider range of gradings.

Specifically, suppose we are comparing two positive operators A and B. If $A \sqsubseteq B$ crisply, then $B = A + D$ for some positive operator D. However, if this is not the case, then we can consider an error term E so that now

$$A + D = B + E$$

Then we have that $B - A = D - E$, i.e. that there is a wholly positive and a wholly negative component to the difference $B - A$. In Bankova et al. [2019] the authors render the error term E as being of the form $(1 - k)A$, where $k \in [0, 1]$. Then the value k is the strength of the hyponymy relation between A and B. The drawback of this approach is that the span of A must be included within the span of B. Lewis [2019b] proposes two alternative gradings based on the error term that do not suffer from this drawback:

$$k_{BA}(A, B) = \frac{Tr(D - E)}{Tr(D + E)} \tag{4}$$

$$k_E(A, B) = 1 - \frac{||E||}{||A||} \tag{5}$$

In equation (4), in the worst case the positive difference term D is 0, and then $k_{BA} = -1$. In the best case $E = 0$ and then $k_{BA} = 1$. In equation (5), in the worst case $E = A$, and then $k_E = 0$. In the best case $E = 0$ and then $k_E = 1$.

2.3 Building positive operators for words

In Bankova et al. [2019], a broader term such as *mammal* is viewed as a weighted sum over projectors describing instances of mammals. For example:

$$\llbracket mammal \rrbracket = p_d \, |dog\rangle \, \langle dog| + p_c \, |cat\rangle \, \langle cat| + p_w \, |whale\rangle \, \langle whale| + ...$$

$$\text{where} \quad \forall i.p_i \geq 0 \text{ (and some kind of normalisation may be applied)}$$

Lewis [2019b] propose a means of building positive operators for words using distributional word vectors and information about hyponymy relations from resources such as WordNet Miller [1995], as follows. In general, the meaning of a word w is considered to be given by a collection of unit vectors $\{|w_i\rangle \in W\}_i$, where each $|w_i\rangle$ represents an instance of the concept expressed by the word. Then the operator:

$$\llbracket w \rrbracket = \sum_i p_i \, |w_i\rangle \, \langle w_i| \in W \otimes W \tag{6}$$

represents the word w. The p_i are weightings derived from the text, and there are various choices about what these should be.

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 $\llbracket w \rrbracket$ as in equation (6), 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.

2.4   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 the current paper, we will need to normalize operators so that their maximum eigenvalue is less than or equal to 1, as this will allow us to apply our proposed negation operator.

2.5   Composing positive operators

Building positive operators as proposed gives us representations for individual words. However, the representations are all states in one object of **CPM**(**FVect**), whereas for verbs, adjectives, and so on, we need morphisms in **CPM**(**FVect**). In order to obtain these, we use an approach outlined in Kartsaklis et al. [2012]. Firstly, we consider the spaces for noun and sentence to be the same, so now our pregroup types n and s both map to the same space W. To represent adjectives and verbs, representations of type $W \otimes W$ or $W \otimes W \otimes W$ are needed. In order to encode our representations in $W \otimes W$, we need to use the word representations we have built to define suitable morphisms in **CPM**(**FVect**). Kartsaklis et al. [2012] use the notion of a *Frobenius algebra*. Working in **FVect**, a Frobenius algebra over a finite-dimensional vector space with bases $\{|i\rangle\}_i$ is given by

$$\Delta :: |i\rangle \mapsto |i\rangle \otimes |i\rangle \qquad \iota :: |i\rangle \mapsto 1 \qquad \mu :: |i\rangle \otimes |i\rangle \mapsto |i\rangle \qquad \xi :: 1 \mapsto |i\rangle$$

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,
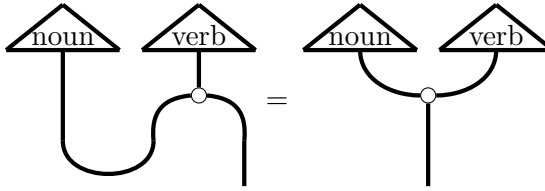
In **FVect** the multiplication μ implements pointwise multiplication of the two vectors. In **CPM**(**FVect**) we have access to the same algebra, and the multiplication μ operates similarly - namely, given two positive operators A and B, $\mu(A \otimes B)$ implements pointwise multiplication of the two operators. We call this operator **Mult** or $\odot$. Whilst simple and theoretically motivated, this operation is not desirable for linguistic purposes as it is commutative, so that 'dog bites man' gets the same representation as 'man bites dog'.

In Coecke [2019], Lewis [2019a], two other multiplications are proposed for combining positive operators. One, which we call **BMult** or $*_B$, was originally proposed in Leifer and Poulin [2008], Leifer and Spekkens [2013] as a quantum Bayesian operation. This takes two operators A and B and returns the non-commutative and non-associative product $B^{\frac{1}{2}} A B^{\frac{1}{2}}$. In Coecke and Meichanetzidis [2020], the authors show that this operation is also related to a Frobenius algebra, with the caveat that the algebra corresponds to a basis for W that diagonalises B.

The second, which we call **KMult**, or $*_K$, is to form a completely positive map from a positive matrix B by decomposing B into a weighted sum of orthogonal projectors $B = \sum_i p_i P_i$, and then forming the map

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

If we again consider a basis that diagonalises B, this operation then corresponds to the Frobenius multiplication $\mu(A \otimes B)$ in that basis. To see this, consider

$$B = \sum_i b_i |i\rangle \langle i|, \quad A = \sum_{jk} a_{jk} |j\rangle \langle k|$$

9

Then

$$\mathcal{B}(A) = \sum_i b_i \ket{i}\bra{i} \circ \left(\sum_{jk} a_{jk} \ket{j}\bra{k}\right) \circ \ket{i}\bra{i} \tag{7}$$

$$= \sum_i b_i \ket{i}\bra{i} \circ \sum_j a_{ji} \ket{j}\bra{i} \tag{8}$$

$$= \sum_i b_i a_{ii} \ket{i}\bra{i} = \mu(A \otimes B) \tag{9}$$

We therefore have three ways of combining positive operators. Moreover, each of these combination methods preserves the property that the eigenvalues must be less than or equal to 1. For the operations **Mult** and **KMult**, the spectral radius is submultiplicative with respect to the Hadamard (pointwise) product of two positive semidefinite matrices Horn and Johnson [1985], implying that the maximum eigenvalue of $A \odot B$ is bounded by 1. For the case of **BMult**, note that the product $B^{\frac{1}{2}} A B^{\frac{1}{2}}$ is similar to AB and hence has the same eigenvalues. Then the maximum eigenvalue of the product AB is bounded by the product of the maximum eigenvalues of A and of B Bhatia [2013], again implying that the maximum eigenvalue of AB is bounded by 1.

To apply these multiplications linguistically, choices must be made about the order in which they are applied, since neither **BMult** nor **KMult** are associative. In particular for transitive verbs there are a number of different choices, and some of these are discussed in Lewis [2019b]. For now, we limit to simple intransitive sentences, of the form *noun verb*.

The operators we outlined above are summarised below.

$$\text{Mult: } [\![noun \; verb]\!] = [\![noun]\!] \odot [\![verb]\!] \tag{10}$$

$$\text{BMult: } [\![noun \; verb]\!] = [\![noun]\!] *_B [\![verb]\!] = [\![verb]\!]^{\frac{1}{2}} [\![noun]\!] [\![verb]\!]^{\frac{1}{2}} \tag{11}$$

$$\text{KMult: } [\![noun \; verb]\!] = [\![noun]\!] *_K [\![verb]\!] = \sum_i p_i P_i [\![noun]\!] P_i \tag{12}$$

where in KMult $[\![verb]\!] = \sum_i p_i P_i$.

3 Modelling negation in $\mathcal{CP}_1(V)$

So far, we have shown how to build positive operators from a corpus of text, together with information about hyponymy relations. We have also shown how to lift the simple operators thus described to the maps required for functional words such as verbs and adjectives. We now describe how to model negation.

As discussed, one approach to modelling negation is to map a vector to the subspace orthogonal to it. We can incorporate this in our model very easily, since in the case of projectors, this is equivalent to subtracting the associated matrix from the identity matrix. Consider a vector $|dog\rangle$ that we have learnt in a distributional manner from a corpus. We can lift this representation to a positive operator by forming the projector $|dog\rangle\langle dog|$, which forms a one-dimension subspace of the vector space W. We can then form an operator

$$[\![not\ dog]\!] = \mathbb{I} - |dog\rangle\langle dog|$$

which encompasses the $n - 1$-dimensional subspace orthogonal to the projector $|dog\rangle\langle dog|$. In the general case, we define

$$[\![not\ w]\!] := \mathbb{I} - [\![w]\!] \tag{13}$$

When we restrict to the subset $\mathcal{CP}_1(W)$ over a vector space W, this operation preserves positivity of the operator and also maps operators into the set $\mathcal{CP}_1(W)$.

Importantly, this operation is *not* a morphism of **CPM**(**FVect**), and therefore a suitable home needs to be found for it. We do not provide an answer to that in this paper, leaving it for ongoing work. Rather, we look at how this operation interacts with composition, the Löwner ordering, and how it works in implementation.

3.1   How *not* interacts with the (graded) Löwner ordering

Consider operators A and $B \in \mathcal{CP}_1(W)$. Under the crisp Löwner ordering, we have

$$A \sqsubseteq B \iff B = A + D \tag{14}$$
$$\iff \mathbb{I} - B = \mathbb{I} - (A + D) \tag{15}$$
$$\iff \mathbb{I} - B + D = \mathbb{I} - A \iff not\ B \sqsubseteq not\ A \tag{16}$$

Considering an error term E, we use the notation $\sqsubseteq_E$ if $B + E = A + D$. With such an error term,

$$A \sqsubseteq_E B \iff B + E = A + D \tag{17}$$
$$\iff \mathbb{I} - (B + E) = \mathbb{I} - (A + D) \tag{18}$$
$$\iff \mathbb{I} - B + D = \mathbb{I} - A + E \iff not\ B \sqsubseteq_E not\ A \tag{19}$$

Depending on the grading we use, the strength of the hyponymy relation will be affected. Using the k_{BA} grading (equation (4)) we have that *not B* is a hyponym of *not A* with strength

$$k_{BA}(not\ B, not\ A) = \frac{Tr(D - E)}{Tr(D + E)} = k_{BA}(A, B)$$

Using k_E (equation (5)), we have:

$$k_E(not\ B, not\ A) = 1 - \frac{||E||}{||not\ B||} \neq k_E(A, B)$$

3.2 How *not* interacts with composition

We focus here just on the case of intransitive sentences composed of a subject and a verb. When we negate the noun we obtain the following expressions:

$$[\![not\ noun]\!] \odot [\![verb]\!] = (\mathbb{I} - [\![noun]\!]) \odot [\![verb]\!] \tag{20}$$

$$= diag([\![verb]\!]) - [\![noun]\!] \odot [\![verb]\!] \tag{21}$$

$$[\![not\ noun]\!] *_B [\![verb]\!] = (\mathbb{I} - [\![noun]\!]) *_B [\![verb]\!] \tag{22}$$

$$= [\![verb]\!]^{\frac{1}{2}}[\![verb]\!]^{\frac{1}{2}} - [\![verb]\!]^{\frac{1}{2}}[\![noun]\!][\![verb]\!]^{\frac{1}{2}} \tag{23}$$

$$= [\![verb]\!] - [\![noun]\!] *_B [\![verb]\!] \tag{24}$$

$$[\![not\ noun]\!] *_K [\![verb]\!] = (\mathbb{I} - [\![noun]\!]) *_K [\![verb]\!] \tag{25}$$

$$= \sum_i p_i P_i P_i - \sum_i p_i P_i [\![noun]\!] P_i \tag{26}$$

$$= [\![verb]\!] - [\![noun]\!] *_K [\![verb]\!] \tag{27}$$

Particularly in the case of $*_B$ and $*_K$, these feel like fairly natural interpretations of a sentence with a negated noun. We take the meaning of the verb as a whole, and then subtract out the part of the verb that is applied to the noun.

When we negate the verb we obtain the following expressions:

$$[\![noun]\!] \odot [\![not\ verb]\!] = [\![noun]\!] \odot (\mathbb{I} - [\![verb]\!]) \tag{28}$$

$$= diag([\![noun]\!]) - [\![noun]\!] \odot [\![verb]\!] \tag{29}$$

and, assuming that we use a basis in which $[\![verb]\!]$ is diagonal:

$$[\![noun]\!] *_K [\![not\ verb]\!] = ([\![noun]\!]) *_K (\mathbb{I} - [\![verb]\!]) \tag{30}$$

$$= \sum_i (1 - p_i) P_i [\![noun]\!] P_i \tag{31}$$

$$= \sum_i P_i [\![noun]\!] P_i - \sum_i p_i P_i [\![noun]\!] P_i \tag{32}$$

$$= diag([\![noun]\!]) - [\![noun]\!] *_K [\![verb]\!] \tag{33}$$

The operation $*_B$ does not have a particularly illuminating representation when the verb is negated, but in the case of $\odot$ and $*_K$, these are again fairly natural interpretations of a sentence with a negated verb.

4 Demonstrations

We give a demonstration on a small dataset that this rendering of negation works well together with the composition operators proposed. In particular, we will see that our combination operators can beat baselines that examine just the noun or the verb in the sentence. This is an important baseline since the construction of the dataset is such that entailment does follow from comparing either the nouns or the verbs. Our combination operators do not in general beat an average of two operators, however, they do in some cases.

4.1 Datasets

We build a set of datasets based on the intransitive sentence dataset introduced in Sadrzadeh et al. [2018]. The 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, we have:

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. In the case of the sentence marked T, the first noun is selected as a hyponym of the second noun, and the first verb is selected as a hyponym of the second verb.

For these sentences to be thought of as entailing, we must view them as being implicitly existentially quantified. For example, if we took the pair of sentences

gazelles sprint, mammals run

we can clearly see that the first sentence does not entail the second if we assume a universal quantification - there could easily be, and there are, non-gazelle mammals that don't run. However, if we take an existential quantification, then the fact that there is some gazelle that sprints means that there must be some mammal (the gazelle) who runs (as sprinting is a kind of running).

Bearing in mind that the sentences are existentially quantified, we create three further datasets that include negation. We apply negation only at the word level and not at the sentence level, as this retains the existentially quantified nature of the sentences. Consider an entailing sentence pair such as:

$$dogs\ run \models mammals\ move$$

We include negation in two places: either the noun can be negated, giving us *non-dogs* and *non-mammals*, or else the verbs can be negated, giving us *do not run* and *do not move.*

From *dogs run* $\models$ *mammals move* we then get three more pairs of entailing sentences:

$$\text{some dogs run} \models \text{some mammals move} \tag{34}$$

$$\text{some non-mammals run} \models \text{some non-dogs move} \tag{35}$$

$$\text{some dogs do not move} \models \text{some mammals do not run} \tag{36}$$

$$\text{some non-mammals do not move} \models \text{some non-dogs do not run} \tag{37}$$

To model these, we render the negation of the verb as directly acting on the verb. Another choice would be for the negation to act on the whole sentence, rendering *dogs don't move* as *not(dogs move)*, but this would mean that we now consider the sentence universally quantified. Working out how to include a full account of quantification is an area of further work.

To model these sentences, we therefore calculate, respectively:

$$[\![dogs]\!] * [\![run]\!] \sqsubseteq_k [\![mammals]\!] * [\![move]\!] \tag{38}$$

$$(\mathbb{I} - [\![mammals]\!]) * [\![run]\!] \sqsubseteq_k (\mathbb{I} - [\![dogs]\!]) * [\![move]\!] \tag{39}$$

$$[\![dogs]\!] * (\mathbb{I} - [\![move]\!]) \sqsubseteq_k [\![mammals]\!] * (\mathbb{I} - [\![run]\!]) \tag{40}$$

$$(\mathbb{I} - [\![mammals]\!]) * (\mathbb{I} - [\![move]\!]) \sqsubseteq_k (\mathbb{I} - [\![dogs]\!]) * (\mathbb{I} - [\![run]\!]) \tag{41}$$

where $\sqsubseteq_k \in \{k_{BA}, k_E\}$ is one of the graded hyponymy measures and $* \in \{\odot, *_B, *_K\}$ is one of the compositional operators.

4.2 Construction and composition of positive operators

We follow the construction methods outlined in Lewis [2019b] and summarised in this paper in section 2.3. In order to construct the basic positive operators, we use hyponyms from WordNet Miller [1995], and 50 or 300 dimensional GloVe vectors. The operators produced are normalised to have maximum eigenvalue equal to 1.

To compose positive operators, we use the three composition functions **Mult**, **Mult**, **Mult** discussed in section 2.5. We compare these with three baselines: the average of two operators, a noun-only baseline, and a verb-only baseline. Due to the construction of the datasets, we see that in fact the verb-only and noun-only baselines are fairly strong, since as long as the construction of the individual words models the hyponymy relations well then a verb-only or noun-only model will be able to perform well on these datasets. Note that taking the average of the two operators

Model	noun-verb	¬noun-verb	noun-¬verb	¬noun-¬verb
KS2016 best	0.84	-	-	-
Verb only	0.866	0.867	0.865	0.867
Noun only	0.926	0.921	0.925	0.923
Average	0.947^+	$\mathbf{0.946}^+$	$\mathbf{0.948}^+$	0.946^+
Mult	$\mathbf{0.960}^{*+}$	0.874	0.931^+	$\mathbf{0.950}^+$
BMult	0.948^+	0.892	0.928	0.947^+
BMult switched	0.949^+	0.896	0.916	0.944^+
KMult	0.950^+	0.875	0.925	0.948^+
KMult switched	0.950^+	0.874	0.920	0.948^+

Table 1: Area under ROC curve on the negation datasets, using k_{BA}, WordNet hyponyms, and 300 dimensional GloVe vectors. Figures reported are the average of the 100 values of the test statistic. * indicates significantly better than the Average baselline. $^+$ indicates significantly better than the noun-only baseline.

Metrics and significance measures   Since the entailment measures we use give back a grading, whereas we require a binary response, we calculate area under ROC curve (AUC). The AUC calculates the true positive rate vs. the false positive rate for different cutoff levels of the graded measure. The maximum that can be attained is 1.

To measure the significance of our results, we use bootstrapping Efron [1992] to calculate 100 values of the test statistic (AUC) drawn from the distribution implied by the data. We compare between models using a paired t-test and apply the Bonferroni correction to compensate for multiple model comparisons.

5   Results

We can see that across the board (tables 1, 2, 3, 4), the k_{BA} measure performs more strongly than the k_E measure. The difference in performance is likely to be because the k_{BA} measure is very symmetric, and the dataset is also, meaning that not only are there equal numbers of entailing and non-entailing sentences in the dataset, but the non-entailing datasets are the opposite of the entailing datasets. Enhancing the datasets with some random pairings would likely degrade the performance of the k_{BA} measure. Investigating the differences in performance in a less balanced dataset is an area of further work.

Model	noun-verb	¬noun-verb	noun-¬verb	¬noun-¬verb
KS2016 best	0.84	-	-	-
Verb only	0.635	0.637	0.636	0.634
Noun only	0.686	0.643	0.684	0.635
Average	0.727	0.778^+	0.777^+	0.782^+
Mult	$\mathbf{0.883^{*+}}$	$\mathbf{0.885^{*+}}$	0.899^{*+}	$\mathbf{0.952^{*+}}$
BMult	0.792^{*+}	0.678^+	0.725^+	0.719^+
BMult switched	0.786^{*+}	0.693^+	0.715^+	0.718^+
KMult	0.873^{*+}	0.725^+	$\mathbf{0.900^{*+}}$	0.732^+
KMult switched	0.839^{*+}	0.879^{*+}	0.732^+	0.666^+

Table 2: Area under ROC curve on the negation datasets, using k_E, WordNet hyponyms, and 300 dimensional GloVe vectors. Figures reported are the average of the 100 values of the test statistic. * indicates significantly better than the Average baselline. $^+$ indicates significantly better than the noun-only baseline.

Model	noun-verb	¬noun-verb	noun-¬verb	¬noun-¬verb
KS2016 best	0.84	-	-	-
Verb only	0.787	0.787	0.786	0.785
Noun only	0.907	0.906	0.903	0.904
Average	0.929^+	$\mathbf{0.925^+}$	$\mathbf{0.929^+}$	$\mathbf{0.930^+}$
Mult	$\mathbf{0.942^{*+}}$	0.836	0.915^+	0.925^+
BMult	0.917^+	0.861	0.914^+	0.920^+
BMult switched	0.918^+	0.859	0.912^+	0.922^+
KMult	0.929^+	0.829	0.910^+	0.926^+
KMult switched	0.926^+	0.821	0.911^+	0.930^+

Table 3: Area under ROC curve on the negation datasets, using k_{BA}, WordNet hyponyms, and 50 dimensional GloVe vectors. Figures reported are the average of the 100 values of the test statistic. * indicates significantly better than the Average baselline. $^+$ indicates significantly better than the noun-only baseline.

In the case of the k_{BA} measure, increasing the dimensionality of the underlying vector space improved performance across all sentence types. This was not the case for the k_E measure, where for sentences of the type *noun - not verb* and *not noun - verb* performance using the k_E measure improved with lower dimensionality (tables 2 and 4)

The best results were obtained using the k_{BA} measure and 300-dimensional GloVe vectors. In this set of results (table 1) the Average baseline proves hard to beat, however Mult also performs strongly for sentences with either no word negated or both words negated. For these two classes of sentences, it is also notable that all

Model	noun-verb	¬noun-verb	noun-¬verb	¬noun-¬verb
KS2016 best	0.84	-	-	-
Verb only	0.601	0.605	0.605	0.607
Noun only	0.708	0.724	0.706	0.720
Average	0.753^+	0.791^+	0.783^+	0.797^+
Mult	0.847^{*+}	$\mathbf{0.845}^{*+}$	$\mathbf{0.891}^{*+}$	$\mathbf{0.925}^{*+}$
BMult	0.751^+	0.694	0.738^+	0.751^+
BMult switched	0.728^+	0.707	0.727^+	0.758^+
KMult	$\mathbf{0.875}^{*+}$	0.702	0.875^{*+}	0.802^+
KMult switched	0.808^{*+}	0.791^{*+}	0.726^+	0.815^{*+}

Table 4: Area under ROC curve on the negation datasets, using k_E, WordNet hyponyms, and 50 dimensional GloVe vectors. Figures reported are the average of the 100 values of the test statistic. * indicates significantly better than the Average baseline. $^+$ indicates significantly better than the noun-only baseline.

composition functions enable better performance than the strong non-compositional noun-only baseline. A similar pattern is seen when using 50-dimensional vectors with the $k_B A$ measure (table 3), where the benefit of using a compositional operator is also seen for the sentence type *noun - not verb*.

The benefit of using compositional operators is also seen for the k_E measure (tables 2 and 4), where using a compositional operator helps in almost all cases over the (admittedly much worse) non-compositional noun-only baseline.

Across both measures and dimensionalities performance is poor on the sentence type *not noun - verb*. More research is needed to investigate why this is.

6 Discussion and Conclusions

We have introduced a negation operator for use in the **CPM**(**FVect**) flavour of DisCoCat. The operators is based on the notion of projection onto the orthogonal subspace, used previously by Widdows and Peters [2003]. The operator works well together with the composition operators Mult, BMult, and KMult discussed in Lewis [2019a,b], Coecke and Meichanetzidis [2020], and in many cases perform well on a toy dataset of sentence entailments.

More investigation into the properties of the **BMult** and **KMult** operators is needed. Coecke and Meichanetzidis [2020] have shown that the two operators can be combined together in a double density matrix setting, meaning that the operators can be given a natural home.

Work is also ongoing to build operators from corpora in a less supervised way.

Recent work on learning Gaussian embeddings Vilnis and McCallum [2014] may be leveraged to build the representations needed.

Further, testing on larger scale datasets is also needed. Ideally, the kinds of entailment relations we are looking at should be useful for textual entailment and reasoning systems. Expanding the models we currently have to test on realistic datasets is desirable.

Another major unanswered question is where the negation operator should sit theoretically. It cannot be viewed as a morphism in **CPM**(**FVect**). Some work in progress is into looking at the set $\mathcal{CP}_1(W)$ as an object of the category **ConvexRel**, introduced in Bolt et al. [2019]. Then, the negation operator can be viewed as a morphism. This is an area of further work.

References

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 and Roberto Zamparelli. 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. Association for Computational Linguistics, 2010.

Rajendra Bhatia. *Matrix analysis*, volume 169. Springer Science & Business Media, 2013.

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.

Joe Bolt, Bob Coecke, Fabrizio Genovese, Martha Lewis, Dan Marsden, and Robin Piedeleu. Interacting conceptual spaces i: Grammatical composition of concepts. In *Conceptual Spaces: Elaborations and Applications*, pages 151–181. Springer, 2019.

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

Bob Coecke and Konstantinos Meichanetzidis. Meaning updating of density matrices. *arXiv preprint arXiv:2001.00862*, 2020.

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.

Ellie D'Hondt and Prakash Panangaden. Quantum weakest preconditions. *Mathematical Structures in Computer Science*, 16(03):429–451, 2006.

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

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.

Karl Moritz Hermann, Edward Grefenstette, and Phil Blunsom. "not not bad" is not "bad": A distributional account of negation. *arXiv preprint arXiv:1306.2158*, 2013.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge University Press, 1985.

Dimitri Kartsaklis. Coordination in categorical compositional distributional semantics. *arXiv preprint arXiv:1606.01515*, 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.

Germán Kruszewski, Denis Paperno, Raffaella Bernardi, and Marco Baroni. There is no logical negation here, but there are alternatives: Modeling conversational negation with distributional semantics. *Computational Linguistics*, 42(4):637–660, December 2016. doi: 10.1162/COLI_a_00262. URL https://www.aclweb.org/anthology/J16-4003.

Matthew S Leifer and David Poulin. Quantum graphical models and belief propagation. *Annals of Physics*, 323(8):1899–1946, 2008.

Matthew S Leifer and Robert W Spekkens. Towards a formulation of quantum theory as a causally neutral theory of bayesian inference. *Physical Review A*, 88 (5):052130, 2013.

M. Lewis. Hyponymy in discocat, 2019a. URL http://www.cs.ox.ac.uk/ACT2019/preproceedings/Martha%20Lewis.pdf. Under review.

Martha Lewis. 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, September 2019b. INCOMA Ltd. doi: 10.26615/978-954-452-056-4_075. URL https://www.aclweb.org/anthology/R19-1075.

Jean Maillard, Stephen Clark, and Edward Grefenstette. 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, 2014.

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.

Denis Paperno, Marco Baroni, et al. 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)*, volume 1, pages 90–99, 2014.

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, Dimitri Kartsaklis, Bob Coecke, and Mehrnoosh Sadrzadeh. Open system categorical quantum semantics in natural language processing. *arXiv:1502.00831*, 2015.

Anne Preller and Mehrnoosh Sadrzadeh. Bell states and negative sentences in the distributed model of meaning. *Electronic Notes in Theoretical Computer Science*, 270(2):141–153, 2011.

Laura Rimell, Amandla Mabona, Luana Bulat, and Douwe Kiela. Learning to negate adjectives with bilinear models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 71–78, 2017.

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, Stephen Clark, and Bob Coecke. The Frobenius anatomy of word meanings I: subject and object relative pronouns. *Journal of Logic and Computation*, page ext044, 2013.

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.

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

Peter Selinger. A survey of graphical languages for monoidal categories. In *New structures for physics*, pages 289–355. Springer, 2010.

John van de Wetering. Entailment relations on distributions. *arXiv preprint arXiv:1608.01405*, 2016.

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.

Dominic Widdows and Stanley Peters. Word vectors and quantum logic: Experiments with negation and disjunction. *Mathematics of language*, 8(141-154), 2003.