Composing distributions: mathematical structures and their linguistic interpretation

Mohan Ganesalingam[*]
University of Cambridge

Aurelie Herbelot[†]
Universität Potsdam

The goal of this paper is to consider the mathematical operations proposed in the literature for composing distributional vectors using mixture models, and assess their linguistic plausibility by looking at various aspects of their behaviour. We explore representational issues related to the space in which distributional vectors live, and to the ways composition operations affect that space. Having generally argued in favour of pointwise operations, we investigate which of those allow for full recursivity and conclude with an extensive discussion of additive and multiplicative models. Our claims are supported both by experimental results from the literature on distributional compositionality and linguistic considerations on the representation of lexical meaning.

1. Introduction

Often presented as a complement to model-theoretic semantics, distributional semantics aims to represent some aspect of lexical meaning as a function of the contexts in which a given word appears. The idea of meaning being partially provided by linguistic context is usually credited to Harris (1954), who stated that words which are similar in meaning occur in similar contexts. Following this idea, some work in computational linguistics, starting with Harper (1965), has been devoted to building and evaluating models which represent words as **distributions**, i.e., vectors in a multidimensional space where each dimension corresponds to a potential context for a lexical item (Curran 2003; Turney and Pantel 2010; Clark 2012; Erk 2012).

More recently, it has been suggested that in order to integrate distributional semantics with model-theoretic formalisms, methods should be found to compose the distributions of single words (Clark and Pulman 2007). Indeed, while it is clear that the representation of *carnivorous mammal* in formal semantics can be written as $\lambda x[\text{carnivorous}'(x) \wedge \text{mammal}'(x)]$, it is less clear how the lexical semantics of the phrase should be described in distributional terms. Several composition operations have been proposed in the literature, ranging from simple pointwise multiplication to variations on the tensor product, and extensively evaluated on a range of tasks (Mitchell and Lapata 2008, 2010; Guevara 2010, 2011; Baroni and Zamparelli 2010; Widdows 2008; Grefenstette and Sadrzadeh 2011; Giesbrecht 2009; Socher et al. 2012). The linguistic implications of using particular distributional models for representing meaning are however only just starting to be discussed (Baroni, Bernardi, and Zamparelli 2012; Erk 2013), and many questions remain unanswered. For instance, is it reasonable to fully

[*] Computer Laboratory, William Gates Building, 15 JJ Thomson Avenue, Cambridge, CB3 0FD ,UK. E-mail: mohan.ganesalingam@cl.cam.ac.uk

[†] EB Kognitionswissenschaften, Universität Potsdam, Karl-Liebknecht Straße, Golm, Germany. E-mail: aurelie.herbelot@cantab.net

regard distributional semantics as a (physical) geometrical system? Is it meaningful to use composition operations which transform the vector basis used for single words? How does vector composition behave with respect to recursivity? Etc.

The main goal of this paper is to start analysing the various mathematical operations suggested for composing distributions in the light of the requirements which we feel are necessary for a sensible linguistic interpretation of distributional representations. In order to do this, we will rely heavily on the experimental results reported so far in the literature – primarily, those of Mitchell and Lapata (2010), who provided results on a wide range of functions and thereby supplied us with excellent data to examine. By attempting to explain those results theoretically, we hope to get to a better understanding of the properties needed by a distributional account to reflect linguistic data.

Distributional systems are complex: they involve choices at many levels, from the construction of an appropriate semantic space to the weighting function applied to the components of the vectors, to the actual composition operation and the similarity measure used for evaluation. This paper is by no means an attempt to cover all possible interactions between those levels. In particular, we will leave out the complex question of how dimensionality reduction techniques such as Latent Semantic Analysis (LSA, Deerwester et al. (1990)) or topic models (Steyvers and Griffiths (2007)), the goal of which is to compress the semantic space in a meaningful fashion, interact with composition. Further, our focus will be on 'mixture' models, i.e. functions that apply to two (or more) column vectors (with or without scalar weighting). Models involving matrices as functional operations will be mentioned when appropriate but we will leave their analysis for further work[1].

The paper is structured as follows. After a section on related work, we first consider the issue of space in distributional systems. Specifically, we ask whether some vector bases are more meaningful than others and how particular composition operations affect distributional representations by transforming the space in which words live. Having argued in favour of pointwise operations, we turn to the issue of modelling recursivity and discuss the problems arising when weighting such operations. We eventually restrict the field of meaningful composition functions to additive and multiplicative models and discuss both approaches as well as the related tensor framework of Clark, Coecke, and Sadrzadeh (2008), highlighting their merits and disadvantages.

2. Related work

The composition of a phrase in distributional semantics is usually obtained by either 'combining' the vectors of the components of the phrase (in so-called 'mixture models' – which we will focus on in this paper) or by treating some words as functions 'acting' on other words. Various mathematical methods can be used to perform such compositions. In this section, we review the main attempts so far, focusing on the papers that compare different methods or introduce a new framework.

Mitchell and Lapata (2010) (henceforth M&L), expanding on Mitchell and Lapata (2008), perform similarity experiments over a range of 8 different composition functions and as such, provide particularly interesting data to comment on. Their system is evaluated on pairs of frequent phrases covering three grammatical categories: adjective-

[1] For a linguistically-motivated discussion of distributional functions involving tensor spaces, see again Baroni, Bernardi, and Zamparelli (2012).

noun phrases (e.g. *new information – further evidence*), noun-noun compounds (e.g. *party official – opposition member*) and verb-object constructions (e.g. *write book – hear word*). Those phrase pairs (36 per categories) are annotated with similarity values by human participants on a scale of 1 to 7. The goal of the system is then to produce similarity figures which correlate with human judgements. We will draw heavily on the findings of M&L in the course of this paper and therefore will not provide a detailed description of their experiments in this section. The main point to note is that pointwise multiplication performs best across the three grammatical constructions.

Clark, Coecke, and Sadrzadeh (2008) and Coecke, Sadrzadeh, and Clark (2010), as implemented in Grefenstette and Sadrzadeh (2011), introduce a tensor-based framework which combines a categorial grammar with distributional representations. Their proposal, which we describe in more detail in §5.3, has the benefit of taking syntax into account when performing composition. It also considers the question of which semantic space should be chosen to contain the meaning of sentences. Experimental results are reported on similarity tasks which partially overlap with Mitchell and Lapata (2008). The framework is shows to perform better than simple additive and multiplicative models.

Widdows (2008) gives an overview of vector operations that he suggests may be used for composing distributions. He describes small-scale experiments designed to assess the strengths and weaknesses of some of the proposed operators: addition, vector product, tensor product and convolution. Of relevance to our work, one of his experiments concerns the similarity of verb-noun pairs (e.g. *earn money* against *pay wages*). Widdows' results suggest that tensor product performs better than addition and vector product.

Guevara (2010, 2011) argues that different syntactic constructs will probably be expressed by different composition operations and that, for each construction, it may be possible to learn an appropriate function, representing the effect of one class of words over its arguments. Accordingly, he experiments with models based on addition, multiplication, circular convolution and partial least squares regression. One novelty of his approach is the evaluation and, where appropriate, training of the models using distributions of observed phrases (i.e. *black* composed with *cat* is evaluated against *black_cat*). His experiments show that for adjective-noun pairs, the partial least squares regression (PLSR) model performs best, while the additive model gives better results on verb-noun pairs. It is worth noting that the PLSR model can be described as linear combination with parameters learnt from actual adjective-noun phrases. As such, it is essentially as an additive model of the form

$$\mathbf{p} = \mathbf{A}\mathbf{u} + \mathbf{B}\mathbf{v}$$

where $\mathbf{A}$ and $\mathbf{B}$ are matrices learnt via PLSR.

Also working on adjectives, Baroni and Zamparelli (2010) follow Guevara's intuition that an appropriate composition function can be learnt by observing actual phrases in a large enough corpus. However, they go one step further in tailoring the function to particular linguistic contexts and argue that a different operation should be learnt for each adjective in their data. Additionally, they regard modification as a function from noun meaning to noun meaning and propose to represent the adjective as a matrix 'applying' to the noun to produce another noun. Using partial least squares regression, they learn each matrix individually and perform composition by multiplying this matrix by the noun vector. Their model is therefore of the form

$$\mathbf{p} = \mathbf{B}\mathbf{v}$$

where $\mathbf{B}$ is a matrix representing the adjective, learnt via PLSR. It outperforms both addition and multiplication, as well as Guevara's account.

Similarly, Socher et al. (2012) implement composition as function application, but model all words as both a vector and a matrix. The vector encodes the lexical meaning of the word while the matrix represents how it modifies the meaning of its arguments (composition happens in the order given by the syntactic parse tree for the phrase/sentence to be modelled). The model is trained using a neural network and evaluated on two sentiment analysis tasks and the classification of semantic relationships.

Finally, we should mention models which cannot be directly described as composition operations but attempt to represent the meaning of a word in context by modifying its corpus-wide vector. Erk and Padó (2008) is an example of such line of work.

3. Considerations on distributional spaces

In this section, we will investigate the implications that particular composition functions have for the semantic space in which their arguments live. Focusing on column vector distributions, we will argue for distributional representations which are informative at the component level and, using as evidence for our claims the experimental results reported so far in the literature, suggest that linguistically transparent vector spaces contribute to better models of phrasal meaning.

3.1 Geometry and basis-independence

Distributional models can be interpreted geometrically: words are 'vectors' in a 'space' and distance measures applicable to geometrical systems can be used to query the semantic relatedness of lexical items. In this section, we will consider this interpretation with regard to a particular geometrical property: basis-independence.

M&L draw a distinction between basis-dependent and basis-independent operations, and introduce one operation, which they term 'dilation', precisely because it is a basis-independent analogue of another operation they consider. They find however that one basis-dependent operation, namely pointwise multiplication of vectors, was consistently as effective or more effective than basis-independent operations.

There is one central context in which basis independence of vectors is indispensable. In the physical world, there are no privileged axes; that is, there are no canonical directions which can be labelled 'X', 'Y' and 'Z', i.e., there is no canonical basis. One consequence of this is that physical theories need to make sense when interpreted with respect to any (suitably orthogonal) basis. This in turn means that operations on vectors which yield different physical vectors in different bases are simply nonsensical: they cannot possibly form part of reasonable physical theories. This point is so central to physics that to a physicist or applied mathematician, the term 'vector' denotes an object which is *by definition* basis-independent; such an object is not itself in, list of numbers, but can be converted into a list of numbers given a basis.

In the linguistic context, we are not dealing with a physical vector: a distribution vector does not represent a physical quantity. It is therefore worth trying to understand what basis independence might mean from a linguistic perspective and to determine whether it is appropriate here. When doing this, it will be useful to have a illustrative numerical example at hand. We will borrow such an example from M&L: the following are example vectors for the word 'practical' and 'difficulty':[2]

$$\begin{array}{rccccccc} & & \text{music} & \text{solution} & \text{economy} & \text{craft} & \text{reasonable} & \\ \textbf{practical} & (& 0 & 6 & 2 & 10 & 4 &) \\ \textbf{difficulty} & (& 1 & 8 & 4 & 4 & 0 &) \end{array}$$

Here '6' measures the propensity of 'practical' to occur near 'solution' in texts, and so on. What does it mean to transform this vector into a different basis? The question is most easily answered by choosing a particularly simple transformation, namely a rotation by 45° in two axes which leaves the other three axes untouched. The change of basis matrix for this transformation is as follows:

$$\begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 & 0 & 0 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Accordingly, the components of the vector in the transformed basis are as follows:

$$\begin{array}{rccccccc} & & \frac{\text{music}+\text{solution}}{\sqrt{2}} & \frac{\text{music}-\text{solution}}{\sqrt{2}} & \text{economy} & \text{craft} & \text{reasonable} & \\ \textbf{practical} & (& 6/\sqrt{2} & -6/\sqrt{2} & 2 & 10 & 4 &) \\ \textbf{difficulty} & (& 9/\sqrt{2} & -7/\sqrt{2} & 4 & 4 & 0 &) \end{array}$$

The first component of the first of these vectors measures the tendency of 'practical' to occur near both 'music' and 'solution'; the second component measures the tendency of practical to occur near 'music' more often than it occurs near 'solution'. This is of course a particularly simple example; with a general rotation, components will mix information about the propensity of 'practical' to occur with five words ('music', 'solution', 'economy', 'craft' and 'reasonable').

A basis-independent operation is an operation that can be applied to rotated vectors just as easily as to the original vectors. For example, consider pointwise multiplication. When applied to the vectors in the original basis, it will yield the following vector for 'practical difficulty':

$$\begin{array}{ccccccc} & \text{music} & \text{solution} & \text{economy} & \text{craft} & \text{reasonable} & \\ (& 1 & 48 & 8 & 40 & 0 &) \end{array}$$

If this combined vector is transformed into the rotated basis, we obtain:

$$\begin{array}{ccccccc} & \frac{\text{music}+\text{solution}}{\sqrt{2}} & \frac{\text{music}-\text{solution}}{\sqrt{2}} & \text{economy} & \text{craft} & \text{reasonable} & \\ (& 49/\sqrt{2} & -47/\sqrt{2} & 8 & 40 & 0 &) \end{array} \tag{1}$$

[2] It would be more conventional to write these as column vectors, but row vectors are equally valid and considerably more compact.

When applied to the vectors measured relative to the rotated basis, it will yield:

$$\begin{array}{ccccccc} & \frac{\text{music}+\text{solution}}{\sqrt{2}} & \frac{\text{music}-\text{solution}}{\sqrt{2}} & \text{economy} & \text{craft} & \text{reasonable} & \\ (& 27 & 21 & 8 & 40 & 0 &) \end{array} \tag{2}$$

Since the vectors of (1) and (2) are not the same, it follows that pointwise multiplication is not a basis-independent operation.

Generally, if we object to an operation on the grounds that it is basis-dependent, what we are objecting to is precisely the fact that the operation cannot be applied to vectors measured in terms of $\frac{\text{music}+\text{solution}}{\sqrt{2}}$, etc. In a physical system, this kind of constraint is completely natural. For the linguistic system we are dealing with, we would contend that this is not the case: there seems to be no reason to require operations to be naturally applicable to vectors measured in terms of $\frac{\text{music}+\text{solution}}{\sqrt{2}}$, etc. In consequence, we do not believe that basis independence is a natural desideratum for operations on distribution vectors.

M&L's argument in favour of using basis independence is that it has the benefit of being applicable to vectors in their original form and in a reduced semantic space. Whether the results of an operation performed on a full vector space and the equivalent reduced space should be identical is a nontrivial question which we are unable to answer here. But even if we subscribe to this view, it should be noted that not all dimensionality reduction techniques relate to a change in basis. Principal Component Analysis (PCA) *is* essentially just a change of basis, so basis-independent methods are likely to perform as well with PCA as without. We are not however aware of any results stating that topic model methods such as Latent Dirichlet Allocation, for instance, is related to change of basis. Without such a result there is no reason to prefer a basis-independent method.

M&L's results confirm that the operation does not bring any significant improvement and performs, in fact, slightly worse than the multiplicative model on a full vector space. Our view is therefore that, although some properties of (physical) geometrical systems can be put to good use in distributional semantics, some others are not linguistically relevant and can even harm the representation.

3.2 Basis-transforming operations: tensor product and circular convolution

We now consider two operations with basis-transforming properties: tensor product and circular convolution. M&L found that composition of distribution vectors via pointwise multiplication outperformed composition using the tensor product, and that both of these significantly outperformed vector composition via circular convolution. In fact, circular convolution was found to be substantially worse than every other model in every category considered. In this section, we will attempt to explain these experimental findings.

3.2.1 Tensor product. It may be useful to begin by giving an overview of the tensor product. As in the previous section, this is best done with the aid of an example; once again, we will borrow the example vectors of M&L:

$$\begin{array}{rcccccc} & \text{music} & \text{solution} & \text{economy} & \text{craft} & \text{reasonable} & \\ \textbf{practical} \; (& 0 & 6 & 2 & 10 & 4 &) \\ \textbf{difficulty} \; (& 1 & 8 & 4 & 4 & 0 &) \end{array}$$

The simplest description of the tensor product of these two vectors is as follows: it contains the product of every component of the first vector with every component of the second vector. Each vector has 5 components; consequently the tensor product has 25 components. It is convenient (though not obligatory) to write down the tensor product in the form of a matrix:

$$\begin{array}{l} \\ \text{music} \\ \text{solution} \\ \text{economy} \\ \text{craft} \\ \text{reasonable} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{music} & \text{solution} & \text{economy} & \text{craft} & \text{reasonable} \end{array} \\ \begin{pmatrix} 0 \times 1 & 0 \times 8 & 0 \times 4 & 0 \times 4 & 0 \times 0 \\ 6 \times 1 & 6 \times 8 & 6 \times 4 & 6 \times 4 & 6 \times 0 \\ 2 \times 1 & 2 \times 8 & 2 \times 4 & 2 \times 4 & 2 \times 0 \\ 10 \times 1 & 10 \times 8 & 10 \times 4 & 10 \times 4 & 10 \times 0 \\ 4 \times 1 & 4 \times 8 & 4 \times 4 & 4 \times 4 & 4 \times 0 \end{pmatrix} \end{array}$$

In other words, the tensor product is:

$$\begin{array}{l} \\ \text{music} \\ \text{solution} \\ \text{economy} \\ \text{craft} \\ \text{reasonable} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{music} & \text{solution} & \text{economy} & \text{craft} & \text{reasonable} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \mathbf{6} & 48 & 24 & 24 & 0 \\ 2 & 16 & 8 & 8 & 0 \\ 10 & 80 & 40 & 40 & 0 \\ 4 & 32 & 16 & 16 & 0 \end{pmatrix} \end{array}$$

For illustrative purposes, let us fix on one element of this tensor product, say the '6' marked in bold. This element is the product of the propensity of 'practical' to occur near 'solution' and the propensity of 'difficulty' to occur near 'music'.

M&L always measure the similarity of two vectors using the cosine of the angle between them.[3] In the case of tensor products, they (presumably) follow the same approach, by treating each tensor as a single long vector. In our example above, 'practical difficulty' would be represented by the following vector (with 25 elements):

$$(0\ 0\ 0\ 0\ 0\ 6\ 48\ 24\ 24\ 0\ 2\ 16\ 8\ 8\ 0\ 10\ 80\ 40\ 40\ 10\ 4\ 32\ 16\ 16\ 0)$$

One can find the angle between this vector and another 25-element vector (obtained from another tensor product, representing another two-word phrase) in the usual fash-

[3] Due to space restrictions, as well as our focus on explaining the results of M&L, we will not consider any other similarity measure in this paper. We leave the exhaustive study of particular interactions between composition and similarity operations for further work.

ion, using the following formula:

$$\text{cosine of angle between } \mathbf{a} \text{ and } \mathbf{b} = \frac{\mathbf{a}.\mathbf{b}}{\sqrt{(\mathbf{a}.\mathbf{a})(\mathbf{b}.\mathbf{b})}}$$

It is difficult to directly obtain an intuitive sense of what this angle in a high-dimensional space means in linguistic terms. Fortunately, through the use of a mathematical identity, we can obtain a much more natural, and comprehensible, perspective on the situation. The relevant identity has very short proof using the summation convention (Einstein 1916), but we prefer to present a more easily understood proof using a concrete example.[4] We will temporarily restrict ourselves to illustrating the situation using three-component vectors, rather than five-component vectors, in order to keep the relevant equations to a reasonable length.

Let us therefore suppose that we are comparing two adjective-noun pairs. The first adjective and noun will be represented by vectors $\mathbf{a} = (a_1, a_2, a_3)$ and $\mathbf{n} = (n_1, n_2, n_3)$, and the second adjective and noun by vectors $\mathbf{A} = (A_1, A_2, A_3)$ and $\mathbf{N} = (N_1, N_2, N_3)$. The two relevant tensor products, $\mathbf{a} \otimes \mathbf{n}$ and $\mathbf{A} \otimes \mathbf{N}$, are therefore:

$$\begin{pmatrix} a_1n_1 & a_1n_2 & a_1n_3 \\ a_2n_1 & a_2n_2 & a_2n_3 \\ a_3n_1 & a_3n_2 & a_3n_3 \end{pmatrix} \qquad \text{and} \qquad \begin{pmatrix} A_1N_1 & A_1N_2 & A_1N_3 \\ A_2N_1 & A_2N_2 & A_2N_3 \\ A_3N_1 & A_3N_2 & A_3N_3 \end{pmatrix}$$

Or, represented as vectors:

$$\begin{pmatrix} a_1n_1 & a_1n_2 & a_1n_3 & a_2n_1 & a_2n_2 & a_2n_3 & a_3n_1 & a_3n_2 & a_3n_3 \end{pmatrix}$$

and

$$\begin{pmatrix} A_1N_1 & A_1N_2 & A_1N_3 & A_2N_1 & A_2N_2 & A_2N_3 & A_3N_1 & A_3N_2 & A_3N_3 \end{pmatrix}$$

To take the dot product of these vectors, we simply multiply the corresponding components and add in the usual fashion to obtain:

$$\begin{aligned}(\mathbf{a} \otimes \mathbf{n}).(\mathbf{A} \otimes \mathbf{N}) &= a_1n_1A_1N_1 + a_1n_2A_1N_2 + a_1n_3A_1N_3 \\ &\quad + a_2n_1A_2N_1 + a_2n_2A_2N_2 + a_2n_3A_2N_3 \\ &\quad + a_3n_1A_3N_1 + a_3n_2A_3N_2 + a_3n_3A_3N_3\end{aligned}$$

Simply by shuffling the order of terms in products, we find that this is equal to:

$$\begin{aligned} &a_1A_1n_1N_1 + a_1A_1n_2N_2 + a_1A_1n_3N_3 \\ +&a_2A_2n_1N_1 + a_2A_2n_2N_2 + a_2A_2n_3N_3 \\ +&a_3A_3n_1N_1 + a_3A_3n_2N_2 + a_3A_3n_3N_3\end{aligned}$$

[4] The end result of this proof is briefly mentioned by e.g. Widdows (2008) or again Giesbrecht (2009) but we find it helpful to spell all steps out, to convince the reader of our subsequent argument.

Collecting common factors shows that this in turn is equal to:

$$\begin{aligned}
&a_1A_1(n_1N_1 + n_2N_2 + n_3N_3)\\
&+a_2A_2(n_1N_1 + n_2N_2 + n_3N_3)\\
&+a_3A_3(n_1N_1 + n_2N_2 + n_3N_3)\\
= &(a_1A_1+a_2A_2 + a_3A_3)(n_1N_1 + n_2N_2 + n_3N_3)
\end{aligned}$$

But $a_1A_1 + a_2A_2 + a_3A_3$ is just $\mathbf{a}.\mathbf{A}$, and $n_1N_1 + n_2N_2 + n_3N_3$ is just $\mathbf{n}.\mathbf{N}$. So the dot product of tensor products can be computed using the following simple formula:

$$(\mathbf{a} \otimes \mathbf{n}).(\mathbf{A} \otimes \mathbf{N}) = (\mathbf{a}.\mathbf{A})(\mathbf{n}.\mathbf{N})$$

Although we have only deduced this identity for vectors of length 3, it holds for vectors of arbitrary length. As noted above, this can be proved very compactly using the summation convention:

$$\begin{aligned}
(\mathbf{a} \otimes \mathbf{n}).(\mathbf{A} \otimes \mathbf{N}) &= (\mathbf{a} \otimes \mathbf{n})_{(i,j)}(\mathbf{A} \otimes \mathbf{N})_{(i,j)}\\
&= (a_in_j)(A_iN_j)\\
&= a_iA_in_jN_j = (\mathbf{a}.\mathbf{A})(\mathbf{n}.\mathbf{N})
\end{aligned}$$

By the exact same calculation we performed above, we can also deduce that

$$(\mathbf{a} \otimes \mathbf{n}).(\mathbf{a} \otimes \mathbf{n}) = (\mathbf{a}.\mathbf{a})(\mathbf{n}.\mathbf{n}) = |\mathbf{a}|^2|\mathbf{n}|^2$$

and

$$(\mathbf{A} \otimes \mathbf{N}).(\mathbf{A} \otimes \mathbf{N}) = (\mathbf{A}.\mathbf{A})(\mathbf{N}.\mathbf{N}) = |\mathbf{A}|^2|\mathbf{N}|^2$$

Using these identities, we find that the cosine of the angle between $\mathbf{a} \otimes \mathbf{n}$ and $\mathbf{A} \otimes \mathbf{N}$ is:

$$\begin{aligned}
&\frac{(\mathbf{a} \otimes \mathbf{n}).(\mathbf{A} \otimes \mathbf{N})}{\sqrt{((\mathbf{a} \otimes \mathbf{n}).(\mathbf{a} \otimes \mathbf{n}))((\mathbf{A} \otimes \mathbf{N}).(\mathbf{A} \otimes \mathbf{N}))}}\\
&= \frac{(\mathbf{a}.\mathbf{A})(\mathbf{n}.\mathbf{N})}{\sqrt{|\mathbf{a}|^2|\mathbf{n}|^2|\mathbf{A}|^2|\mathbf{N}|^2}}\\
&= \frac{(\mathbf{a}.\mathbf{A})(\mathbf{n}.\mathbf{N})}{|\mathbf{a}||\mathbf{n}||\mathbf{A}||\mathbf{N}|}\\
&= \frac{(\mathbf{a}.\mathbf{A})}{|\mathbf{a}||\mathbf{A}|} \times \frac{(\mathbf{n}.\mathbf{N})}{|\mathbf{a}||\mathbf{N}|}
\end{aligned}$$

But this is simply the product of the direction cosine of $\mathbf{a}$ and $\mathbf{A}$ and the direction cosine of $\mathbf{n}$ and $\mathbf{N}$. So by using the tensor product, one is simply computing the similarity of $\mathbf{a}$ to $\mathbf{A}$, independently computing the similarity of $\mathbf{n}$ and $\mathbf{N}$, and then multiplying the two. Or in more informal terms, one is independently comparing adjective to adjective and noun to noun.

Now that we have a handle on what it actually means to compare tensor products, we are in a position to explain the results of M&L. The relevant data is as follows (see §2 for a short description of the evaluation task):

Model	Adjective-Noun	Noun-Noun	Verb-Object
Multiplicative Full vector space	0.46	0.49	0.37
Tensor product Full vector space	0.41	0.36	0.33

Table 1
M&L Precision figures for the multiplication and tensor product models (similarity task)

There are in fact two separate points to explain here. First, the tensor product is generally weaker than pointwise multiplication. Second, the tensor product approach performs particularly badly when applied to noun-noun compounds in the full vector space. The explanation for the first point is that independently comparing, say, adjectives $\mathbf{a}$ and $\mathbf{A}$ and nouns $\mathbf{n}$ and $\mathbf{N}$ disregards the possibility that $\mathbf{a}$ might be similar to $\mathbf{N}$, or $\mathbf{A}$ to $\mathbf{n}$. For example, consider the phrases 'musical friend' and 'friendly musician'. Insofar as 'musical' is unrelated to 'friendly', and 'friend' unrelated to 'musician', the tensor product approach will predict that these items are highly dissimilar. By contrast, pointwise multiplication combines the information from adjectives and nouns before comparison, and should therefore rate 'musical friend' and 'friendly musician' as very similar. This effect may have caused the tensor product system to rate pairs such as *better job – economic problem* on the low side while overrating pairs such as *new language – modern technology* (because of the presumably high similarity between *new* and *modern*).

Turning to the second point, the reason that this effect is particularly pronounced with noun-noun compounds is that the elements in the two positions are of the same kind, i.e. that they are both nouns. It is more likely that a noun will be similar to another noun than that it will be similar to another adjective. Together with a degree of flexibility in the ordering of elements in noun-noun phrases, this means that in the noun-noun case, it is more likely that one has similarity between the first element in phrase A and the second element in phrase B (or vice versa). Pairs such as *assistant manager – board member* may consequently be rated lower than they should be by the model.

Our conclusion is therefore that the combination of tensor product and cosine measure in a similarity task does *not* evaluate the composition operation itself, and the results obtained from such a setup will depend on the evaluation data. The fact that Widdows (2008) reports contrary results to M&L can be taken, we believe, as an artefact of the test data. We will thus not attempt to make claims about the performance of the tensor product as a composition operation. We will however note that, as in the case of some rotations (see §3.1), the space produced by the operation does not have a straightforward linguistic interpretation (we will expand on this point at the end of the next section).

To finish our discussion of the tensor product, we should note that one of its drawbacks is that the dimensionality of the composed vector (n^2 for a two-word phrase in a vector space with n dimensions) makes it difficult to compare phrases of different lengths (Washtell 2011). Circular convolution has been suggested in the literature as an operation which overcomes this issue (Jones and Mewhort 2007) so we will now turn to it.

3.2.2 Circular convolution. Circular convolution computes a highly compressed version of the tensor product. Once again, we will use our familiar example:

$$\begin{array}{l|ccccc|}
 & \text{music} & \text{solution} & \text{economy} & \text{craft} & \text{reasonable} \\
\text{music} & 0 & 0 & \mathbf{0} & 0 & 0 \\
\text{solution} & 6 & 48 & 24 & \mathbf{24} & 0 \\
\text{economy} & 2 & 16 & 8 & 8 & \mathbf{0} \\
\text{craft} & \mathbf{10} & 80 & 40 & 40 & 0 \\
\text{reasonable} & 4 & \mathbf{32} & 16 & 16 & 0
\end{array}$$

The components of the convolution are simply obtained by summing elements along with diagonals of the tensor product. If one 'runs off' the bottom (resp. right hand side) of the matrix before obtaining five values, one 'wraps around' to the top (resp. left hand side) of the matrix. So, for example, the first element of the circular convolution of the matrix above is obtained by summing the elements along the main diagonal, $0 + 48 + 8 + 40 + 0 = 96$. Another element is obtained by summing the five entries marked in bold, $10 + 32 + 0 + 24 + 0 = 66$.

As elsewhere in this paper, our aim is to interpret the linguistic significance of this mathematical operation. As in previous cases, we will try to illustrate this with the help of a concrete example. Let us focus on the diagonal marked in bold in the matrix above. We may start by recalling the significance of the elements in the tensor product. The '10' represents the tendency of 'practical difficulty' to have its first element occurring near 'craft' and its second element occurring near 'music', computed by multiplying the appropriate measures for 'practical' , and 'difficulty'. Schematically, we might write:

$$\begin{aligned}
\text{practical}_{\text{craft}} &= 10 \\
\text{difficulty}_{\text{music}} &= 1 \\
\text{practical difficulty}_{(\text{craft},\text{music})} &= 1 \times 10 = 10
\end{aligned}$$

The other elements of the relevant diagonal are as follows:

$$\begin{aligned}
&\text{practical difficulty}_{(\text{reasonable},\text{solution})} = 32 \\
&\text{practical difficulty}_{(\text{music},\text{economy})} = 0 \\
&\text{practical difficulty}_{(\text{solution},\text{craft})} = 24 \\
&\text{practical difficulty}_{(\text{economy},\text{reasonable})} = 0
\end{aligned}$$

The element of the circular convolution obtained by summing these is

$$\begin{aligned} &\text{practical difficulty}_{(\text{craft},\text{music})}\\ +&\text{practical difficulty}_{(\text{reasonable},\text{solution})}\\ +&\text{practical difficulty}_{(\text{music},\text{economy})}\\ +&\text{practical difficulty}_{(\text{solution},\text{craft})}\\ +&\text{practical difficulty}_{(\text{economy},\text{reasonable})} = 66 \end{aligned}$$

This represents the propensity of 'practical difficulty' to have its first element occurring near 'craft' and its second element occurring near 'music', *or* to have its first element occurring near 'reasonable' and its second element occurring near 'solution', *or* its first element occurring near 'music' and its second element occurring near 'economy', *or* etc. We are unable to assign any linguistic significance to this quantity. The dimensions of the obtained vector are a) not interpretable as a linguistically relevant basis b) not directly comparable to those found in a phrase of different length (because the basis differs). Further, it should be clear at this point that the actual elements of a circular convolution are highly dependent on the base chosen. If we had used the basis (solution, music, economy, craft, reasonable) instead of (music, solution, economy, craft, reasonable), we would have obtained a completely different vector. There does not seem to be any particular linguistic requirement for having a basis with a particular ordering of dimensions.

We should note that the component computed from the main diagonal *is* meaningful. This component measures the propensity of the two words in the phrase being considered (e.g. 'practical' and 'difficulty') to occur near the same words, and thus gives a measure of the similarity of the two words. However, if one is working with vectors of a realistic length, the information in this one component will have a negligible effect on the overall result. We would therefore predict that a vector constructed using circular convolution will perform badly in tasks focusing on phrasal meaning. This is confirmed by the results of M&L.

3.3 Desiderata for a distributional space

We have so far expressed two desiderata with regard to distributional spaces and the vectorial representations associated with them:

1. The dimensions of a distributional space should be linguistically interpretable.
2. Phrases of various lengths should be comparable.

These desiderata can be satisfied by a) having a sensible distributional space for single words and b) choosing composition operations which leave that space intact.

This brings us to a representational point. Distributional models are often evaluated with regard to their ability to simulate semantic similarity. In such evaluations, the general shape of the distributional vector is what matters, and the precise values of the vector components – or for that matter, what those components represent – are not vital.

If we are to claim that distributions are a *general* representation of (at least some aspects of) lexical meaning, though, it seems important to ask whether they are fit to model information which we might consider to be part of lexical semantics. This, we argue, involves asking what we mean when we say that vector $\vec{v}$ has value p along dimension d.

Let us first consider the fact that distributions approximate the kind of information given by so-called 'feature norms' (Devereux et al. 2009; Andrews, Vigliocco, and Vinson 2009), i.e. they are able to identify which features are salient for a particular concept. By choosing appropriate linguistic dimensions, this approximation holds: in the simplest vector space models, where each dimension corresponds to a word or a syntactic/semantic construct, we can say for instance that *mouse* accounts for a large proportion of the lexical meaning of *cat* because the *cat* vector has a high value along the (linguistically meaningful) dimension *mouse*. [5]

Another way in which the basis of a distributional vector can be said to be lexically informative is by comparing different vectors: by saying, for instance, that *mouse* is more associated with *cat* than it is with *dog*, we establish the respective positions of *cat* and *dog* in the semantic space and thereby account for their difference in meaning. This models both conceptual similarities and characteristics in a transparent, explanatory fashion. Again, this information is only available because we have chosen a vector space in which dimensions correspond to linguistic entities.

Retaining a sensible vector space after composition will allow us to ascertain, for instance, what is similar about *black cat* and *white owl*, and crucially, to also compare *black cat* and *cat* along particular dimensions. So our general claim is that we should prefer operations which are not basis-transforming or to the least, be able to return to a linguistically meaningful vector space from whichever basis the composition has taken us to (which, arguably, operations such as circular convolution only allow in a very lossy fashion).

The above is of course a representational point, but representation and system performance are tightly linked and there is no reason to believe that a model based on non-linguistic features should provide better results than one based on naturally occurring linguistic entities. In fact, the results reported by M&L show that transparent bases are likely to perform better in experimental setups. In the next section, we follow up on this, bringing into play 'compositionality' itself.

3.4 Pointwise Operations

Our discussion in the previous section points us towards a general point about operations on distribution vectors. In that discussion, we computed a number of quantities like:

$$\mathrm{X\,Y}_{(\mathrm{economy},\mathrm{reasonable})} = \mathrm{X}_{\mathrm{economy}} \times \mathrm{Y}_{\mathrm{reasonable}}$$

This quantity is computed from the 'economy' component of the vector for X and the 'reasonable' component of the vector for Y. It measures the tendency of a two-word

[5] We also consider 'linguistically meaninful' certain models of reduced vector spaces, like LSA or topic models, which attempt to group semantically compatible contexts in one dimension (at least from a theoretical point of view).

phrase 'X Y' to have its first element occur near instances of 'economy' and its second element occur near instances of 'reasonable'. Based on linguistic intuitions, we would not expect this combination to carry much useful information. M&L's findings provide empirical confirmation of this intuition.

The root cause of the lack of information here seems to be that combining different components of two different vectors simply does not seem to be a natural semantic operation. That is, the fact that the two parts of a phrase occur near 'economy' and 'reasonable' respectively would not, intuitively speaking, seem to tell us very much at all about the meaning of the phrase itself. Note by contrast that combining the same component of two different vectors is certainly meaningful. For example,

$$\mathrm{X\,Y}_{(\mathrm{economy},\mathrm{economy})} = \mathrm{X}_{\mathrm{economy}} \times \mathrm{Y}_{\mathrm{economy}}$$

should give a reasonable estimate of the tendency of the phrase 'X Y' to occur near the word 'economy'; this is clearly highly informative.

One might respond to this point by noting that while the particular quantity described has little significance, other quantities from the same class might be of some use. For example, given that 'dog' and 'hound' are in many ways similar, one might expect

$$\mathrm{X\,Y}_{(\mathrm{dog},\mathrm{hound})} = \mathrm{X}_{\mathrm{dog}} \times \mathrm{Y}_{\mathrm{hound}}$$

to carry more semantic information: it gives us some measure of the propensity of the phrase as a whole to occur near 'dog'-like words. This is a reasonable observation, but it is difficult to actually make use of it to construct operations for composing distribution vectors. The key point here is that if we are adhering to compositionality, we simply have to write down a mathematical function that takes two vectors and produces another vector; such a function cannot utilise any source of information about similarities of different 'components'. Our models have no source of information that tells us whether a given pair is potentially informative (like (dog, hound)) or uninformative (like (economy, reasonable)). As a result, if we try to admit the informative pairs into our model, we will inevitably end up admitting the uninformative pairs as well. Because relatively few pairs will be informative, we will find that the noise obtained from the uninformative pairs tends to swamp the actual information from the informative pairs. Indeed, it is arguable that this happened in the circular convolution example we gave above: 'music' and 'craft' are sufficiently similar that

$$\text{practical difficulty}_{(\text{craft},\text{music})}$$

may have contained useful information — but this quantity was combined with four clearly uninformative quantities (practical difficulty$_{(\text{economy},\text{reasonable})}$, etc.), and in that process any useful information would have been drowned by noise.

This situation *might* lead one to consider abandoning the principle of compositionality, and allowing extra sources of information to play a role in the composition — or better, 'combination' — of distribution vectors.[6] In this way one might be able to make use of meaningful combinations like (dog, hound) without being forced to also use the

[6] Indeed, some approaches proposed in the literature, such as that of Kintsch (2001), do exactly this.

meaningless combinations. Setting aside the serious methodological problems involved in abandoning compositionality, there is a very practical reason not to follow such an approach.

That reason is as follows. If one believes that the similarity of 'dog' to 'hound' is important and needs to be taken into account when working with distributions of vectors, then one should utilise that fact not only during composition, but *also* when measuring the similarity of distribution of vectors. More specifically, if one is attempting to compute the similarity of vectors $\mathbf{x}$ and $\mathbf{y}$, one needs to take into account the similarity of x_{dog} to y_{hound} and the similarity of x_{hound} to y_{dog}. The most widely used measure of similarity of distribution vectors, the direction cosine, does not do this.

Thus attempting to utilise similarity of components during composition is unsatisfactory in that it is addressing one surface manifestation of a problem, rather than the problem itself. A more satisfactory solution affects not just composition, but also measurements of similarity. In our view, the most natural operations of this kind involve adjusting the actual vectors at the point of measurement and this is the approach taken when using dimensionality reduction techniques such as LSA or topic models (Deerwester et al. (1990), Steyvers and Griffiths (2007)).

Our position is therefore that attempting to combine information from different components of two distribution vectors during composition is likely to be ineffective. As noted above, this theoretical prediction is borne out by the findings of M&L.

It will be useful to introduce some terminology to distinguish operations that do and do not combine different components of distribution vectors. Following the usual convention in mathematics, we will say that an operation $\odot$ on distribution vectors is *pointwise* if each component of $x \odot y$ is computed from corresponding components of the input vectors x and y. That is, $\odot$ is *pointwise* if $(x \odot y)_{\text{music}}$ is computed from x_{music} and y_{music} (only), $(x \odot y)_{\text{solution}}$ is computed from x_{solution} and y_{solution} (only), and so on. Any operation which is not of this kind will be described as *non-pointwise*.

Given this terminology, our empirically testable position can simply be stated as follows: we expect effective ways of combining distribution vectors to be pointwise, or to be highly similar to a pointwise operation. This position, of course, need some qualifications. It is perfectly possible to take an essentially pointwise operation and wrap it up in a way that makes it look non-pointwise. For example, as we found in §3.2 above, the tensor product approach used in M&L appears at first sight to be a non-pointwise operation, but by the use of an appropriate identity it can be reduced to independent pointwise operations on the parts of a two-element phrase. Equally, it will be perfectly possible to take a pointwise operation and alter it slightly to obtain something that is comparably effective; we would not consider operations of this kind to invalidate our prediction.

4. Recursive Rules

The aim of M&L and other similar investigations (see §2) is to investigate methods for compositionally constructing distribution vectors for phrases or sentences. The experiments in these investigations have largely been restricted to methods for determining distribution vectors for two-element compounds, such as 'industrial area' or 'encourage [a] child'. While this is an eminently sensible place to start investigating the composition of distribution vectors, consideration of larger phrases brings to light some constraints that narrow the space of reasonable composition strategies. More specifically, certain strategies that seem to be very reasonable when applied to two-element phrases turn out to have severe drawbacks when applied to larger phrases. The purpose of this section is to present these issues and to describe which strategies 'scale' beyond two-element phrases.

The key source of difficulty when dealing with large phrases relates to recursive syntactic rules, such as the following:

$$N' \Rightarrow AP\ N'$$

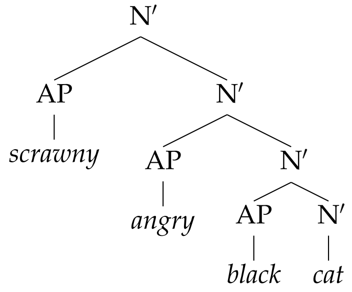
In a classical theory of compositional semantics, there will be a semantic composition rule associated with the syntactic one just given (Montague 1973). The rule will be different for different classes of adjective (Partee 1994). The underlying aim of attempts to extend the compositional approach to distribution vectors is to give semantic composition rules of just this kind, albeit rules which operate on distribution vectors, rather than on logical representations. For example, M&L propose the following rules for combining a distribution vector $\mathbf{u}$ for an adjective (phrase) and a distribution vector $\mathbf{v}$ for a noun (or nominal projection):

$$\mathbf{p} = \alpha\mathbf{u} + \beta\mathbf{v}.$$

α and β are scalar parameters which are empirically optimised on a training set; M&L obtain the values $\alpha = 0.88$ and $\beta = 0.12$.

So, for example, if our vector for 'black' is $\mathbf{v}_{\text{black}}$ and our vector for 'cat' is $\mathbf{v}_{\text{cat}}$, this compositional approach would obtain the vector $\alpha\mathbf{v}_{\text{black}} + \beta\mathbf{v}_{\text{cat}}$ for 'black cat'.

Now, because the syntactic rule we are considering is recursive, it can be applied repeatedly to analyse phrases which consist of a number of adjectives (or adjective phrases) followed by a noun. For example, the phrase 'scrawny angry black cat' can be analysed as follows:



By repeatedly applying our semantic composition rule, we can determine the distribution vectors for this phrase. As above, the vector for 'black cat' is

$$\alpha\mathbf{v}_{\text{black}} + \beta\mathbf{v}_{\text{cat}}.$$

The vector for 'angry black cat' is

$$\alpha\mathbf{v}_{\text{angry}} + \beta(\alpha\mathbf{v}_{\text{black}} + \beta\mathbf{v}_{\text{cat}}) = \alpha\mathbf{v}_{\text{angry}} + \alpha\beta\mathbf{v}_{\text{black}} + \beta^2\mathbf{v}_{\text{cat}}.$$

And so the vector for 'scrawny angry black cat' is

$$\begin{aligned}&\alpha\mathbf{v}_{\text{scrawny}} + \beta(\alpha\mathbf{v}_{\text{angry}} + \alpha\beta\mathbf{v}_{\text{black}} + \beta^2\mathbf{v}_{\text{cat}})\\ &\qquad = \alpha\mathbf{v}_{\text{scrawny}} + \alpha\beta\mathbf{v}_{\text{angry}} + \alpha\beta^2\mathbf{v}_{\text{black}} + \beta^3\mathbf{v}_{\text{cat}}.\end{aligned}$$

To get a sense of what this might mean, we can try substituting the actual parameters obtained by M&L, namely $\alpha = 0.88$ and $\beta = 0.12$. The result is :

$$\mathbf{v}_{\text{scrawny angry black cat}} = 0.88\mathbf{v}_{\text{scrawny}} + 0.106\mathbf{v}_{\text{angry}} + 0.0126\mathbf{v}_{\text{black}} + 0.00152\mathbf{v}_{\text{cat}}$$

This model predicts that the contribution of 'scrawny' to the meaning of the phrase is around eight times larger than the contribution of 'angry', and that it is around seventy times larger than the contribution of 'black'. Indeed, the contributions of 'black' and 'cat' are negligible.

Since this particular result is clearly influenced by the rather extreme values for α and β obtained in M&L, let us try some more moderate value — say $\alpha = \beta = 0.5$. In this case, we obtain:

$$\mathbf{v}_{\text{scrawny angry black cat}} = 0.5\mathbf{v}_{\text{scrawny}} + 0.25\mathbf{v}_{\text{angry}} + 0.125\mathbf{v}_{\text{black}} + 0.0625\mathbf{v}_{\text{cat}}$$

Even with these moderate values, we find that the contribution of 'scrawny' is twice as large as the contribution of 'angry', and four times as large as the contribution of 'black'. This is clearly not compatible with the underlying linguistics of the situation: for example, we would expect the meaning of 'scrawny angry black cat' to be very similar to the meaning of 'angry scrawny black cat'. Indeed, to a first approximation, given a phrase consisting of many adjectives followed by a noun, we would expect the adjectives to make an equal contribution to the meaning of the whole phrase. The equation just derived is also incompatible with the classical compositional model (Partee 1994): in that approach, the order of intersective adjectives is irrelevant, so that 'scrawny angry black cat' means exactly the same as 'angry scrawny black cat'.

The line of criticism given in this section does not only apply to the 'weighted addition' model; it applies equally to a number of other models proposed by M&L. Indeed, of those models, only the multiplication, addition, tensor product approaches (and the baseline 'head-only' approach) behave reasonably when applied to phrases with more than two words. It is striking that the models which are immune to criticism are some of the simplest models proposed. We do not believe this means that complex mathematical models of composition are invariably inappropriate; rather the correct conclusion to draw is that more complicated models need to be carefully tailored to the situation at hand, taking into account linguistic insights.

The weighted addition model is in fact unusual in that it is possible to rehabilitate it with a minor change. As before, we will use the adjective-noun case for illustration. First, recall that we had

$$\mathbf{v}_{\text{scrawny angry black cat}} = \alpha\mathbf{v}_{\text{scrawny}} + \alpha\beta\mathbf{v}_{\text{angry}} + \alpha\beta^2\mathbf{v}_{\text{black}} + \beta^3\mathbf{v}_{\text{cat}}.$$

and that we needed the various adjectives to have the same weight. All that one needs for this is $\beta = 1$; the value of α is irrelevant. (In general, the requirement is that the coefficient corresponding to the recursive category is 1.) So the model given by

$$\mathbf{p} = \alpha\mathbf{u} + \mathbf{v}$$

is not vulnerable to the criticism given in this section.

If one has a model with $\beta \neq 1$, it is easy to remedy the problem by multiplying by a constant factor. For example, consider the model the actual parameters obtained by M&L for adjective-noun combination: $\mathbf{p} = 0.88\mathbf{u} + 0.12\mathbf{v}$. Scaling up by a factor of

$1/0.12$ shows that this is equivalent to $\mathbf{p} = 7.33\mathbf{u} + \mathbf{v}$.[7] If this scaled model is applied to the 'scrawny angry black cat', then we obtain a result in which the adjectives contribute equally:

$$\mathbf{v}_{\text{scrawny angry black cat}} = 7.33(\mathbf{v}_{\text{scrawny}} + \mathbf{v}_{\text{angry}} + \mathbf{v}_{\text{black}}) + \mathbf{v}_{\text{cat}}.$$

Although we will not show this in detail here, it should be noted that very similar considerations apply to models where the weighting factors are matrices. So the model proposed by Guevara (2010, 2011) is not appropriate when it comes to recursivity. This comment, we should clarify, does not apply to Baroni and Zamparelli (2010): there, it is not the weighting factors which are matrices, but the representations for adjectives themselves. In that model one will still obtain different distribution vectors for, say, 'scrawny angry' cat and 'angry scrawny' cat, because matrix multiplication is not commutative. But depending on the actual matrices used, the distribution vectors may be suitably close.

Before closing this section, we should note that the remarks made here do not apply directly to composition of distribution vectors for a verb and for an object noun phrase to form a distribution vector for a verb phrase. Thus we cannot directly rule out the possibility of using an exotic model for verb-object composition. We would however suggest that when any such model is proposed, it is worth examining the potential effect of that model on longer phrases. Of particular significance are phrases with a number of adjectives inside the object ('saw a scrawny angry black cat') and phrases with a nested relative clause involving a verb and object ('saw a cat chasing mice'). Considering phrases of this kind also makes it clear that one cannot indefinitely consider models for different kinds of composition (adjective-noun, verb-object, etc.) in isolation; ultimately it is necessary to examine their interaction with each other, at least if one is aiming to construct a model of composition that works on any but the simplest phrases. We are inclined to believe that considering such interactions will show that the combination of exotic models for e.g. verb-object composition with simple models for e.g. adjective-noun comparison will have unexpected, and linguistically inappropriate, side-effects, unless the exotic models are carefully based on linguistic insights in the fashion described above.

5. Additive and multiplicative models

We noted in the previous section that all of the approaches proposed by M&L other than addition, modified weighted addition, multiplication and tensor products behaved inappropriately when applied to long phrases. We also noted in §3.2 that the tensor product approach was equivalent to independently comparing the first and second elements of two-word phrases. Also, for analytical purposes, the modified version of weighted addition is extremely similar to addition itself. This means that the additive and multiplicative models are of particular interest.

[7] The relative weighting of adjective to noun here seems implausibly high; we suspect that this is a quirk of the particular experimental setup used.

5.1 Pointwise addition and multiplication

In this section, we will discuss and explain the relative performance of pointwise addition and multiplication in different tests, and also present a more general discussion of the strengths and weaknesses of the two approaches and the circumstances in which each is appropriate.

The first and most important point to make is that it only makes sense to speak of 'the additive model' or 'the multiplicative model' with respect to a particular measurement, i.e. a particular way of computing entries of vectors. M&L note that they use the ratio of the probability of the context word given the target word to the probability of the context word overall, computed according to the following equation:

$$v_i(t) = \frac{p(c_i|t)}{p(c_i)} = \frac{\mathit{freq}_{c_i,t} \cdot \mathit{freq}_{\mathit{total}}}{\mathit{freq}_t \cdot \mathit{freq}_{c_i}}$$

The quantities in this equation represent the following:

$\mathit{freq}_{c_i,t}$	frequency of the context word c_i with the target word t
$\mathit{freq}_{\mathit{total}}$	total count of word tokens
freq_t	frequency of the target word t
freq_{c_i}	frequency of the context word c_i

The context window is taken to include five words on either side of the target word.

There is a subtlety worth noting here. Suppose that one simply considers a context word to occur 'with' the target word if it occurs within the target window, and computes the frequency $\mathit{freq}_{c_i,t}$ by adding up the number of times a given context word occurs within the window. In this situation, any given word will be counted as part of the target window of ten different target words, and so will contribute a total of ten to

$$\sum_t \mathit{freq}_{c_i,t}$$

It will however only be counted once in freq_{c_i}. Accordingly,

$$\frac{\mathit{freq}_{c_i,t}}{\mathit{freq}_t}$$

will not be the probability $p(c_i|t)$ of the context word given the target word; it will be ten times that quantity. This consistent scaling of vector entries by the length of the context window would in fact makes little difference to empirical studies like M&L, which only treat two-word phrases. However, such a scaling makes it harder to *interpret* the vector entries in linguistically meaningful terms, and it also leads to problems when considering longer phrases. We will therefore assume that the appropriate correction for the length of the context window has been made.[8] Under this assumption, vector

[8] We have glossed over a subtlety here. In cases where the corpus is split into sentences, the length of the context window can vary because certain target words will occur near one edge of the sentence, so that we will not *always* be overestimating by a factor of ten. The easiest way to obtain legitimate measurements is to

entries will genuinely measure the probability of the context word given that the target word is 'nearby' to the probability of the context word overall. Thus we have:

$$v_i(t) = \frac{p(c_i|t)}{p(c_i)} = \frac{p(c_i, t)}{p(c_i)p(t)}$$

The interpretation of this quantity may be aided by some examples. If $v_i(t) = 2$, then c_i is twice as likely to occur near t as it is in the corpus as a whole. If $v_i(t) = 0.5$, then c_i is one half as likely to occur near t as it is in in the corpus as a whole. And if $v_i(t) = 1$, then c_i is exactly as likely to occur near t as it is in the corpus as a whole. In a sense, 1 is the 'neutral' least informative value, which $v_i(t)$ can take; if $\mathbf{v}(t)$ were a vector consisting entirely of 1s, then knowing that t occurred in a certain position would not give us any information about what words occurred nearby.

Given this discussion of the actual measurement in use, we are in a position to return to the main point, which is that models only make sense with respect to a specific measurement. The importance of this point may be highlighted by considering a different measurement, which has been widely experimented with in the literature on distributions (see e.g. Terra and Clarke (2003), Pantel and Ravichandran (2004), Evert (2004), Mitchell and Lapata (2008), Thater, Fürstenau, and Pinkal (2010)). The *pointwise mutual information* (PMI) of c_i given t is defined as

$$\mathrm{pmi}(c_i; t) = \log \frac{p(c_i|t)}{p(c_i)} = \log \frac{p(c_i, t)}{p(c_i)p(t)}$$

That is, PMI is precisely the logarithm of the measurement $\mathbf{v}(\bullet)$ discussed above. The key point is that pointwise addition of PMIs corresponds to pointwise multiplication of $\mathbf{v}(\bullet)$ as defined above.

So the multiplicative model of M&L would be an additive model for any investigators using PMI. (Note that it makes no sense to multiply PMIs, as they may be negative; correspondingly there is no multiplicative model with respect to PMIs.)

Digressing briefly, we may note that the logarithmic relationship just noted is quite useful, in that it lets us convert compositional methods for PMI into vector composition methods for $v_i(t)$. For example, weighted addition of PMI, according to $p_i = \alpha u_i + \beta v_i$ turns into a form of 'weighted multiplication' for $v_i(t)$:[9] $p_i = u_i^\alpha v_i^\beta$. Recalling the remark about 1 being the 'neutral, least informative' value above may help in interpreting this equation. If $x > 0$, then for $\alpha > 1$, x^α will be further away from 1 than x; so x^α is more informative than x. Conversely, if $\alpha < 1$, x^α will be closer to 1 than x; so x^α is less informative than x. The effect of this equation is quite simple: it allows one to vary the relative contributions of $\mathbf{u}$ and $\mathbf{v}$, just as the weighted addition model does. It is also analytically true that the weighted multiplication model will perform at least as well as the multiplication model, for the simple reason that one can always take $\alpha =$

tally $\mathit{freq}_{c_i,t}$ directly, and then to compute the other quantities via the following equations:

$$\mathit{freq}_{c_i} = \sum_t \mathit{freq}_{c_i,t} \qquad \mathit{freq}_t = \sum_{c_i} \mathit{freq}_{c_i,t} \qquad \mathit{freq}_{total} = \sum_{c_i,t} \mathit{freq}_{c_i,t}$$

[9] The constraints given in §4 apply equally here. For example, if one is dealing with adjective-noun combination, one *must* take $\beta = 1$.

$\beta = 1$.[10,11] Given that the multiplicative model consistently performs well, the weighted multiplication model is therefore potentially quite attractive.

For the remainder of this paper, we will take 'additive model' and 'multiplicative model' to be shorthand for 'additive model with respect to $v_i(t)$' and 'multiplicative model with respect to $v_i(t)$', except where otherwise specified.

A second point which we need to make before turning to the actual analysis of the two models is that the additive model will tend to increase the values found in distribution models. The easiest way to see this is to consider the combination of two 'maximally uninformative' vectors, both consisting entirely of 1s. In the multiplicative model, these will combine to give another vector consisting entirely of 1s. In the additive model, these will combine to give a vector consisting entirely of 2s.[12] This effect is not distortive or problematic, so long as one is using a measure of vector similarity that is insensitive to scaling of vectors, i.e. a measure that satisfies $d(\mathbf{v}, \mathbf{w}) = d(\lambda\mathbf{v}, \mu\mathbf{w})$ for any $\lambda, \mu > 0$.[13] We will however need to be careful in the analysis below, since if $\mathbf{v}(t)$ is a vector produced by adding two (directly measured) distributional vectors, we can no longer interpret $v_i(t) = 2$ as meaning that c_i is twice as likely to occur near t as it is in the corpus as a whole.

Next we turn to the question of why the multiplicative model outperformed the additive model in M&L as well as Mitchell and Lapata (2008). Our comments, we should note, are very tentative. Indeed, the additive model has been found to outperform the multiplicative model in other tasks (Baroni and Zamparelli 2010; Guevara 2010), where the evaluation is performed by comparing the composed vector to the distribution of the corresponding phrase, as directly observed in a corpus. It is fair to assume that the requirements of the evaluation task will favour one operation or another. The following should therefore be seen as general reflections on how various composition methods affect different semantic phenomena.

There may be a number of contributing factors which affected the results in M&L; but one that stands out as particularly likely to be significant relates to word sense ambiguity. As elsewhere in the paper, we will try and illustrate this with the use of an example. Let us consider the word 'wave'. This word has several senses: it can refer to a ridge over a body of water, a sudden occurrence of something, a movement of the hand, etc. Suppose that we are trying to compositionally compute semantics for 'offensive wave', meaning an attack in a military context. We need to combine the distribution vectors for 'offensive' and for 'wave'. In the context in which the word is used, it is clear which sense of 'wave' is meant. If our model of composition can pick out the relevant sense during composition, it will be much more effective in capturing the meaning of the composite phrase.

Now, the directly measured vectors for the words 'offensive' and 'wave' will reflect information from all word senses. So, for example, our system (described in the appendix) outputs that $\text{wave}_{\text{beach}}$ = 28.6052, indicating that we are 28 times as likely to find 'beach' near 'wave' as we are to find it in general language. By contrast, for 'offensive',

[10] The same is true of weighted addition and addition; when M&L found that weighted addition underperformed addition, this indicated that either the training data, the evaluation data, or both were unrepresentative.

[11] This is not to say that weighted multiplication is strictly preferable to multiplication; if two models perform comparably well, one should prefer the one that involves fewer parameters.

[12] This is not an incidental effect; it is precisely because the additive model produces increasingly large vectors that it is able to escape the criticism of 4.

[13] Loosely speaking, under the additive model, the length of a vector measures the amount of information in it, whereas the direction encodes all the distributional information.

we have offensive$_{\text{beach}}$ = 3.04177x10^{-05}. Now let us consider the way in which the two different models combine these values. In the multiplicative model, we have:

$$\text{offensive wave}_{\text{beach}} = \text{offensive}_{\text{beach}} \times \text{wave}_{\text{beach}} = 8.70104 \times 10^{-04}$$

In the additive model, we have

$$\text{offensive wave}_{\text{beach}} = \text{offensive}_{\text{beach}} + \text{wave}_{\text{beach}} = 28.60523$$

Recall that values in the additive model (applied to two directly measured vectors) are twice as large as one might expect, so that the value 28.60523 corresponds (roughly) to a 14.3 in a directly measured vector. Nevertheless, the additive model has produced a much larger value for offensive wave$_{\text{beach}}$ than the multiplicative model, and so has been much less effective in filtering out the irrelevant sense of 'wave'. This is a general phenomenon: the key point is that multiplying by a sufficiently small number can yield a result that is much smaller than the original, whereas averaging with a small number can never reduce the original by more than a factor of two.

Another perspective on this issue is given by the following observation: addition does not discriminate between small values. For example, it treats 0.2, 0.1 and even 0.01 as essentially the same. In the context we are considering, there is a large informational difference between them: if offensive$_{\text{beach}}$ = 0.01, then we can be *very* certain that 'offensive' does not relate to 'beach'. This is not so clear cut with a value of 0.3, say. So addition is sensitive to information stating that a particular context word tends to co-occur with the target word, and insensitive to information stating that a particular context word tends *not* to co-occur with the target word.

Addition being less effective in separating word senses means that it will be less effective at picking out the meanings of adjective-noun, noun-noun and verb-object combinations, which may account for its relatively poor performance in M&L. We should emphasise that this does not mean that addition is always an inappropriate mechanism for combining distribution vectors. The sum of distribution vectors for target words t, t' is an effective measure of those contexts words which relate to *either t or t′*. There are contexts in which this is exactly the desired behaviour. For example, addition seems like a reasonable candidate to model the semantics of disjunction, and may even be appropriate for certain kinds of conjunction, especially sentential conjunction.[14] The more general point here is that different operations may be suited to different kinds of composition (cf. Guevara (2010)).

Before moving on, we should note that there is a straightforward empirical test that would confirm or invalidate our account of the difference in effectiveness between addition and multiplication. If the difference is indeed due to the effectiveness of multiplication in word sense disambiguation, then applying an independent method for word sense disambiguation *before* computing distribution vectors should close the gap between addition and multiplication. (We emphasise that we are not suggesting this as a general method to produce distributions, but only as a way to verify our hypothesis.)

[14] This should not be taken as asserting that addition is the only model suited to these contexts; taking the pointwise maximum (resp. minimum) of the two vectors is an alternative and very natural way to model disjunction (resp. conjunction, albeit conjunction of objects rather than sentences). The different approaches have their own theoretical strengths and weaknesses, which we do not have space to discuss here.

We finish this section with a warning. The discussion above may make it seem as if the multiplicative model is a panacea. While it is extremely attractive, it has a particular weakness, which has not been brought out by the empirical studies. We will discuss this weakness in the next section.

5.2 Confidence

The multiplicative model is extremely sensitive to small values in the input vectors. For example, if one vector contains an entry of 0.01, the impact on the result will be very large. Insofar as the entry is accurate, this is appropriate: as we noted in the previous section, if offensive$_{\text{beach}}$ = 0.01, then we can be *very* certain that 'offensive' does not relate to 'beach'. The problem is that smaller values will also tend to be more inaccurate, because they are based on a smaller amount of input data. To give an illustrative example, if we expect to see 'offensive' occur next to 'beach' 300 times, then we can be reasonably confident that the number actually observed in the corpus is close to this; it might be 270 or 330, but it is unlikely to be 150.[15] So we can be confident that the proportional deviation from the true value will be reasonably small. By contrast, if we expect to see 'offensive' occur next to 'beach' 3 times, then it is quite possible that the observed value is 1, or 6, or even 0. So the proportional deviation may well be quite large. Translated into values $v_i(t)$, this means that an expected value of 1 is likely to correspond to an observed value close to 1 (perhaps 0.9 or 1.1), but that an expected 0.01 may well be measured as 0.003 or even 0.

The issue with multiplication, then, is that a single inaccurate small entry can have a large effect on the corresponding entry in the result vector. In the worst case, if we find a 0 in one input vector (where we expect a small value), we are guaranteed to have a 0 in the output vector, regardless of the size of the corresponding entry in the other input vector. In the case of two-element phrases, this effect will be relatively minor. As one starts to compositionally apply multiplication to longer and longer phrases, such as a string of adjectives followed by a noun, the effect will become more and more pronounced. Only one of the words in a phrase needs to have a 0 entry, or a highly distorted small entry, for the corresponding entry in the result to be inaccurate. The central issue here is that our models do not utilise, or even keep track of, any notion of how *confident* we are about the entries in our distribution vectors.

We should emphasise that the issue just discussed affects some, but not all, models. For example, it will have little effect on similarity measurements using the additive model.

At this point, the obvious question to ask is, how does one minimise the impact of this issue? There are a range of strategies available. One very simple approach is as follows: one picks a minimum value which the ratio of probabilities is allowed to take. That is, we effectively stipulate that

$$v_i(t) = \frac{p(c_i|t)}{p(c_i)}$$

[15] Here 'expect' is meant in the formal sense of 'expectation': the actual measurement is a measurement of a random variable, and 'expectation' refers to the expected value of that random variable.

is never allowed to take a value smaller than some fixed constant η. For example, if we were to take $\eta = 0.01$, then we would never have $v_i(t) < 0.01$. (The actual value of η is best determined empirically.)

The simplest way of implementing this is to rule that

$$v_i(t) = max\left(\frac{p(c_i|t)}{p(c_i)}, \eta\right)$$

One potential downside of this approach is that $v_i(t)$ is, strictly speaking, no longer a ratio of probabilities. This becomes significant if one wants to use $p(c_i|t)$, etc., to compute anything other than $v_i(t)$. For example, there are various techniques, for example related to entropy, that one might use to compute the 'amount of information' in a given distribution vector. In such cases, one needs to adjust the raw frequencies before computing probabilities.

It may be worth noting that this 'minimum value' strategy is quite simplistic and, from a mathematical perspective, artificial. One can find more natural strategies by turning to Bayesian techniques; however, describing these would take us beyond the scope of this paper.

5.3 Tensor Framework

Having discussed the multiplicative model at length, we turn to a variant proposed by Clark, Coecke, and Sadrzadeh (2008). This paper introduces a general tensor-based framework for composing distribution vectors, which has two notable strengths. First, it can compositionally assign semantics-representing vectors to complex sentences in a way which fully respects their syntactic structure. Second, vectors computed from all sentences live in the same space, regardless of the internal structure of said sentences. Grefenstette and Sadrzadeh (2011) report two experiments within this framework. The first of these considers (noun+intransitive verb) composition, and performs comparably to the multiplicative model discussed above. The second considers (noun + transitive verb + noun) combination, and performs slightly better than the natural generalisation of the multiplicative model.

We will begin by focusing on the first experiment. Distribution vectors for nouns are directly computed, after the fashion of M&L. Distribution vectors for whole sentences live in the same vector space as distribution vectors for nouns. This vector space is called N, and is taken to have a basis $\overrightarrow{n_1}, \overrightarrow{n_2}, \ldots \overrightarrow{n_k}$. Distribution vectors for intransitive verbs are elements of the tensor product vector space $N \otimes N$.

The mechanism for computing the actual verb tensor is not part of the core framework of Clark, Coecke, and Sadrzadeh (2008), but follows a method given in Coecke, Sadrzadeh, and Clark (2010). That paper provides detailed equations for the case of the transitive verb; we will begin by giving the analogous equations for the intransitive case, using the same tensorial notation as the original.

To compute the representation for a given intransitive verb, one starts by listing all the nouns which occur as subjects of that verb; let us call these $S_1, S_2, \ldots S_k$. The S_i are counted according to multiplicity; that is, if a particular noun occurs twice with the verb, it occurs twice in the list. One then takes the distribution vectors of each noun in the list; let us call the resulting vectors $\overrightarrow{v_1}, \overrightarrow{v_2}, \ldots \overrightarrow{v_k}$. The vector for the verb is specified to be:

$$\overrightarrow{verb} = \sum_{i,j} C_{ij}(\overrightarrow{n_i} \otimes \overrightarrow{n_j})$$

where

$$C_{ij} = \begin{cases} \sum_l \langle \overrightarrow{v_l} | \overrightarrow{n_i} \rangle & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

(Recall that $\overrightarrow{n_1}, \overrightarrow{n_2}, \ldots \overrightarrow{n_k}$ form the basis of N.)

The vector for an actual sentence is computed using the 'inner product' map $\epsilon_N : N \otimes N \to \mathbb{R}$:

$$\begin{aligned} \overrightarrow{subject\ verb} &= (\epsilon_N \otimes 1_N)(\overrightarrow{subject} \otimes \overrightarrow{verb}) \\ &= (\epsilon_N \otimes 1_N)\left(\overrightarrow{subject} \otimes \left(\sum_{i,j} C_{ij}(\overrightarrow{n_i} \otimes \overrightarrow{n_j})\right)\right) \\ &= \sum_{i,j} C_{ij}((\epsilon_N \otimes 1_N)(\overrightarrow{subject} \otimes \overrightarrow{n_i} \otimes \overrightarrow{n_j})) \\ &= \sum_{i,j} C_{ij} \langle \overrightarrow{subject} | \overrightarrow{n_i} \rangle \overrightarrow{n_j} \end{aligned}$$

It may be worth reformulating this material in more familiar terms. We will use the language of vectors and matrices; vectors will be written in boldface, i.e. as $\mathbf{v}$ rather than $\overrightarrow{v}$, following the usual convention.

To begin with, saying that the representation of the verb is an element of $N \otimes N$ is essentially the same as saying that it is a matrix. More specifically, if nouns and sentences are represented by length m vectors, verbs are represented by m-by-m matrices. An element $\overrightarrow{a} \otimes \overrightarrow{b}$ of $N \otimes N$ is just the matrix $\mathbf{a}\mathbf{b}^T$.[16] Thus $\overrightarrow{n_i} \otimes \overrightarrow{n_j}$ is the matrix $\mathbf{n_i}\mathbf{n_j}^T$.

Since $\overrightarrow{n_i}$ or $\mathbf{n_i}$ is the vector with a '1' in its i^{th} entry and '0' elsewhere, $\overrightarrow{n_i} \otimes \overrightarrow{n_j}$ or $\mathbf{n_i}\mathbf{n_j}^T$ is just the matrix which has a '1' in the i^{th} row and j^{th} column, and '0' everywhere else.[17] Thus

$$\sum_{i,j} C_{ij}(\overrightarrow{n_i} \otimes \overrightarrow{n_j})$$

is just the matrix with C_{ij} in its i^{th} row and j^{th} column; we would normally express this matrix as (C_{ij}) or just C.

[16] Note that this is not the same as $\mathbf{a}^T\mathbf{b}$, which is the dot product $\mathbf{a}.\mathbf{b}$.

[17] For example, in $\mathbb{R}^3$ we have that

$$\mathbf{n_1}\mathbf{n_2}^T = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Now we turn to the equation defining C_{ij}. Because $\overrightarrow{n_i}$ is the i^{th} basis vector, we have that $\langle \overrightarrow{v_l} | \overrightarrow{n_i} \rangle$ is just the i^{th} component of the vector $\mathbf{v_l}$, usually written $(\mathbf{v_l})_i$. So we have

$$C_{ij} = \begin{cases} \sum_l ((\mathbf{v_l})_i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

This tells us that C is a diagonal matrix, whose i^{th} entry is equal to $\sum_l (\mathbf{v_l})_i$, or equivalently to

$$\left(\sum_l \mathbf{v_l} \right)_i$$

In other words, the i^{th} entry in C is i^{th} component of the sum of the noun vectors.

To compose representations for a noun and an intransitive verb, one simply multiplies the verb matrix by the noun vector:

$$\mathbf{v}_{\text{sentence}} = \mathrm{C}_{\text{verb}} \mathbf{v}_{\text{noun}}$$

Since the matrix C here is diagonal, we have:

$$(\mathbf{v}_{\text{sentence}})_i = \left(\sum_l \mathbf{v_l} \right)_i (\mathbf{v}_{\text{noun}})_i$$

In other words, $\mathbf{v}_{\text{sentence}}$ is obtained by pointwise multiplication of the noun vector and a vector $\sum_l \mathbf{v_l}$ which represents the verb.[18] So the tensor-based model of Grefenstette and Sadrzadeh (2011) is very similar to the multiplicative model of M&L; the only difference lies in the way in which the verb vector was constructed. This provides the first part of our promised explanation of why the accuracy of the tensor-based model here is so close to the accuracy of the multiplicative model.

The second part of the explanation involves the two different ways of constructing the verb vector. In the original multiplicative model, the verb vector is directly estimated from the context in the same way that the subject noun vector is. In the tensorial model, the verb vector is constructed by summing the vectors for all the nouns that occur as subjects of the verb. Note that since the direction cosine of vectors is length-independent, this is equivalent to averaging the vectors for all such nouns.

Averaging in this way results in (at least) two distinct and competing effects. On the one hand, verb vectors calculated by averaging are based on much more data than directly computed verb vectors. Indeed, if a given verb occurs N times in the corpus, the vector computed by averaging subjects will be based on N times as much data. Consequently we can be much more confident about the accuracy of the entries in the averaged vector (cf. §5.2). On the other hand, the relationships encoded in the averaged vector will be more distant than those in the directly computed vector: they are two-step relationships rather than direct relationships. (The verb is related to a subject noun, and

[18] Note that there is no standard notation for expressing pointwise multiplication of vectors; the reason is that, as emphasised in §3.1, pointwise multiplication is not a basis-independent operations and thus is not meaningful in physical contexts.

the subject noun to a lexical item.) As such, the information in the average vector will be of a slightly lower quality than the information in the original. The fact that the tensor model and the multiplicative model have essentially identical performance suggests that these two effects may have comparable magnitudes, and that they are consequently cancelling each other out. We should emphasise that this explanation is tentative: there may be other effects in play, and in any case the magnitude of the relevant trends in the data are too small to base solid conclusions on.

Before moving on, we have two final remarks about the tensorial model as applied to (subject + intransitive verb) combinations. The first is that it has difficulty with a certain class of verbs, namely those that have a strong tendency to take a pleonastic pronoun. For example, 'rains/rained' will almost always occur with subject 'it' in a system based on syntactic information.[19] Accordingly, the verb vector for 'rains' will be almost identical to the vector for 'it', as will the verb vector for 'snows/snowed'. Such vectors are clearly not appropriate. Verbs of this kind are sufficiently rare that there should be no noticeable impact on empirical studies. From a theoretical perspective, however, this is a noticeable weakness in the strategy of consistently constructing verb representations from the nouns they occur with.

The second remark relates to a point we made in §5. The tensorial model, like the original multiplicative model, has no way of altering the relative contributions of the subject noun and verb. In §5, we constructed a 'weighted multiplication' model, which overcame this deficiency. It is easy to make a corresponding adjustment to the tensorial model:

$$(\mathbf{v}_{\text{sentence}})_i = \left(\sum_l \mathbf{v_l}\right)_i^{\alpha} (\mathbf{v}_{\text{noun}})_i^{\beta}$$

As noted in §5, this model will provably perform at least as well as the original when trained and evaluated on sufficient data. This is not to say that is always preferable: in return for the introduction of extra parameters, we require a nontrivial improvement in performance. From a linguistic standpoint, however, it seems very possible that there is some room for improvement: there is no reason to believe that the verb and noun are exactly as informative as each other.[20]

As for the second experiment in Grefenstette and Sadrzadeh (2011), which examines transitive verbs, we believe that the authors are correct in saying that it outperforms the simple multiplicative model because it can take account of subject-object asymmetry. Again, it may be worth examining whether altering the relative weightings of subject, verb and object can improve the performance; additionally the tensorial model should be compared against a weighted multiplicative model to ensure that the gain in performance persists.

6. Conclusion

Distributional semantics is often viewed as a 'semantics of similarity' and evaluated as such: the emphasis is on the general shape of distributional vectors and how well

[19] This problem does not occur when building distributions from semantic parses.

[20] The weighted version of the equation no longer conforms with the framework of Coecke, Sadrzadeh, and Clark (2010) — but in the absence of empirical data about performance, it is premature to discuss this issue.

they model that some words or phrases are more related than others. Little is said about individual vector components. In this paper, we have argued that choosing a linguistically transparent semantics space and a composition operation which preserves it results not only in a model which is informative at the vector component level, but also in a representation which performs better in similarity tasks themselves.

More specifically, we pointed out that mathematical features which may be viewed as desirable in the domain of physics – for instance, basis independence – are not necessary or even meaningful in the field of language. We also argued against basis-transforming operations and in the process, noted the importance of considering the interaction of several mathematical operations (e.g. composition and similarity functions) when building a full distributional system. Following on these first thoughts, we argued in favour of pointwise composition operations and discussed weighting of such operations in the light of recursive phenomena. We also examined the issue of confidence in statistical measurements and the effect it has on some operations. We hope to have, in the process, shed light on some of the results reported in the literature.

Although we made a clear statement in favour of pointwise operations, we do not feel we can rank addition- and multiplication-like functions. We have shown throughout the paper that those two types of operations have different explanatory power. We should however mention the weighted multiplication model suggested in §5 as an operation which may potentially outperform the ones proposed in the literature so far.

Generally, we think there is a lot more work to be done in terms of interpreting the mathematical structures that have been proposed in applied computational linguistics. The number and variety of phenomena observable in language means that we have a complex task verifying the plausibility of any operation or representation supposed to model such an encompassing notion as 'the lexicon'. This seems to be a worthwhile task, however: we believe that by considering phenomena which are widespread in language (recursivity) or perhaps not so widespread (pleonastic pronouns) we can very quickly rule out functions or domains of application of functions.

References

Andrews, Mark, Gabriella Vigliocco, and David Vinson. 2009. Integrating experiential and distributional data to learn semantic representations. *Psychological Review*, 116(3):463.

Baroni, Marco, Raffaella Bernardi, and Roberto Zamparelli. 2012. Frege in Space: a Program for Compositional Distributional Semantics. Under review.

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 (EMNLP10)*, pages 1183–1193.

Clark, Stephen. 2012. Vector Space Models of Lexical Meaning. In Shalom Lappin and Chris Fox, editors, *Handbook of Contemporary Semantics – second edition*. Wiley-Blackwell.

Clark, Stephen, Bob Coecke, and Mehrnoosh Sadrzadeh. 2008. A compositional distributional model of meaning. In *Proceedings of the Second Quantum Interaction Symposium (QI-2008)*, pages 133–140.

Clark, Stephen and Stephen Pulman. 2007. Combining Symbolic and Distributional Models of Meaning. In *Proceedings of the AAAI Spring Symposium on Quantum Interaction*, pages 52–55, Stanford, CA.

Coecke, Bob, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis*, 36(1–4):345–384.

Curran, James. 2003. *From Distributional to Semantic Similarity*. Ph.D. thesis, University of Edinburgh, Scotland, UK.

Deerwester, Scott, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

Devereux, Barry, Nicholas Pilkington, Thierry Poibeau, and Anna Korhonen. 2009. Towards unrestricted, large-scale acquisition of feature-based conceptual representations from corpus data. *Research on Language and Computation*, 7:137.

Einstein, Albert. 1916. Die grundlage der allgemeinen relativitätstheorie. *Annalen Phys*, 49(769-822):31.

Erk, Katrin. 2012. Vector space models of word meaning and phrase meaning: a survey. *Language and Linguistics Compass*, 6:10:635–653.

Erk, Katrin. 2013. Towards a semantics for distributional representations. In *Proceedings of the Tenth International Conference on Computational Semantics (IWCS2013)*.

Erk, Katrin and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Honolulu, HI.

Evert, Stefan. 2004. *The statistics of word cooccurrences: word pairs and collocations*. Ph.D. thesis, University of Stuttgart.

Giesbrecht, E. 2009. In search of semantic compositionality in vector spaces. In *Proceedingsof the 17th International Conference on Conceptual Structures: Conceptual Structures: Leveraging Semantic Technologies, ICCS âĂŹ09*, pages 173–184.

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 (EMNLP11)*, pages 1394–1404, Edinburgh, Scotland, UK.

Guevara, Emiliano. 2010. A Regression Model of Adjective-Noun Compositionality in Distributional Semantics. *Proceedings of the 2010 Workshop on Geometrical Models of Natural Language Semantics (ACL 2010)*, pages 33–37.

Guevara, Emiliano. 2011. Computing semantic compositionality in distributional semantics. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*, pages 135–144, Oxford, England, UK.

Harper, Kenneth E. 1965. Measurement of similarity between nouns. In *Proceedings of the 1st International Conference on Computational Linguistics (COLING65)*, pages 1–23, New York, NY.

Harris, Zelig. 1954. Distributional Structure. *Word*, 10(2-3):146–162.

Jones, Michael N. and Douglas J. K. Mewhort. 2007. Representing word meaning and order information in a composite holographic lexicon. *Psychological review*, 114(1):1–37.

Kintsch, W. 2001. Predication. *Cognitive Science*, 25(2):173–202.

Leech, Geoffrey, Roger Garside, and Michael Bryant. 1994. Claws4: The tagging of the british national corpus. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING 94)*, pages 622–628, Kyoto, Japan.

Mitchell, Jeff and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244.

Mitchell, Jeff and Mirella Lapata. 2010. Composition in Distributional Models of Semantics. *Cognitive Science*, 34(8):1388–1429, November.

Montague, Richard. 1973. The Proper Treatment of Quantification in Ordinary English. In Jaakko Hintikka, Julius Moravcsik, and Patrick Suppes, editors, *Approaches to Natural Language*. Dordrecht, pages 221–242.

Pantel, Patrick and Deepak Ravichandran. 2004. Automatically Labeling Semantic Classes. In *Proceedings of Human Language Technology / North American Association for Computational Linguistics (HLT/NAACL-04)*, pages 321–328, Boston, MA.

Partee, B.H. 1994. Lexical semantics and compositionality. In Daniel Osherson, Lila Gleitman, and Mark Liberman, editors, *Invitation to Cognitive Science, second edition. Part I: Language*, volume 1. MIT Press, pages 311–360.

Socher, Richard, Brody Huval, Christopher Manning, and Andrew Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing (EMNLP2012)*, pages 1201–1211, Jeju Island, Korea.

Steyvers, Mark and Tom Griffiths. 2007. Probabilistic topic models. In T. Landauer, D. McNamara, S. Dennis, and W. Kintsch, editors, *Latent Semantic Analysis: A Road to Meaning*.

Terra, Egidio and C. L. A. Clarke. 2003. Frequency estimates for statistical word similarity measures. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 165–172, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thater, Stefan, Hagen Fürstenau, and Manfred Pinkal. 2010. Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL ’10, pages 948–957, Stroudsburg, PA, USA. Association for Computational Linguistics.

Turney, Peter D. and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Washtell, Justin. 2011. Compositional expectation: a purely distributional model of compositional semantics. In *Proceedings of the Ninth International Conference on Computational Semantics*, IWCS ’11, pages 285–294, Oxford, United Kingdom.

Widdows, Dominic. 2008. Semantic Vector Products : Some Initial Investigations. In *Second AAAI Symposium on Quantum Interaction*, number March, Oxford, England, UK.

Appendix

An implementation of a distributional system

The implementation we suggest is close to the M&L system. As background data, we use the British National Corpus (BNC) in lemmatised format. Each lemma is followed by a part of speech according to the CLAWS tagset format (Leech, Garside, and Bryant 1994). For our experiments, we only keep the first letter of each part-of-speech tag, thus obtaining broad categories such as N or V. Furthermore, we only retain words in the following categories: nouns, verbs, adjectives and adverbs (punctuation is ignored). Each article in the corpus is converted into a 11-word window format, that is, we are assuming that context in our system is defined by the five words preceding and the five words following the target.

To calculate co-occurrences, we use the equations suggested in Section 5.2:

$$\mathit{freq}_{c_i} = \sum_t \mathit{freq}_{c_i,t} \qquad \mathit{freq}_t = \sum_{c_i} \mathit{freq}_{c_i,t} \qquad \mathit{freq}_{total} = \sum_{c_i,t} \mathit{freq}_{c_i,t}$$

The weight of each context term in the distribution is given by the function suggested in M&L:

$$v_i(t) = \frac{p(c_i|t)}{p(c_i)} = \frac{freq_{c_i,t} \times freq_{total}}{freq_t \times freq_{c_i}} \tag{.3}$$

As in M&L, we use the 2000 most frequent words in our corpus as the semantic space dimensions.

We experimented with setting a threshold for $v_i(t)$ as discussed in §5.2. As our semantic space consists of very frequent contexts, however, no significant effect was observed in the few results we report here.