Experimental Support for a Categorical Compositional Distributional Model of Meaning

Edward Grefenstette
University of Oxford
Department of Computer Science
Wolfson Building, Parks Road
Oxford OX1 3QD, UK
`edward.grefenstette@cs.ox.ac.uk`

Mehrnoosh Sadrzadeh
University of Oxford
Department of Computer Science
Wolfson Building, Parks Road
Oxford OX1 3QD, UK
`mehrs@cs.ox.ac.uk`

Abstract

Modelling compositional meaning for sentences using empirical distributional methods has been a challenge for computational linguists. We implement the abstract categorical model of Coecke et al. (2010) using data from the BNC and evaluate it. The implementation is based on unsupervised learning of matrices for relational words and applying them to the vectors of their arguments. The evaluation is based on the word disambiguation task developed by Mitchell and Lapata (2008) for intransitive sentences, and on a similar new experiment designed for transitive sentences. Our model matches the results of its competitors in the first experiment, and betters them in the second. The general improvement in results with increase in syntactic complexity showcases the compositional power of our model.

1 Introduction

As competent language speakers, we humans can almost trivially make sense of sentences we've never seen or heard before. We are naturally good at understanding ambiguous words given a context, and forming the meaning of a sentence from the meaning of its parts. But while human beings seem comfortable doing this, machines fail to deliver. Search engines such as Google either fall back on bag of words models—ignoring syntax and lexical relations—or exploit superficial models of lexical semantics to retrieve pages with terms related to those in the query (Manning et al., 2008).

However, such models fail to shine when it comes to processing the semantics of phrases and sentences. Discovering the process of meaning assignment in natural language is among the most challenging and foundational questions of linguistics and computer science. The findings thereof will increase our understanding of cognition and intelligence and shall assist in applications to automating language-related tasks such as document search.

Compositional type-logical approaches (Montague, 1974; Lambek, 2008) and distributional models of lexical semantics (Schutze, 1998; Firth, 1957) have provided two partial orthogonal solutions to the question. Compositional formal semantic models stem from classical ideas from mathematical logic, mainly Frege's principle that the meaning of a sentence is a function of the meaning of its parts (Frege, 1892). Distributional models are more recent and can be related to Wittgenstein's later philosophy of 'meaning is use', whereby meanings of words can be determined from their context (Wittgenstein, 1953). The logical models relate to well known and robust logical formalisms, hence offering a scalable theory of meaning which can be used to reason inferentially. The distributional models have found their way into real world applications such as thesaurus extraction (Grefenstette, 1994; Curran, 2004) or automated essay marking (Landauer, 1997), and have connections to semantically motivated information retrieval (Manning et al., 2008). This two-sortedness of defining properties of meaning: 'logical form' versus 'contextual use', has left the quest for 'what is the foundational structure of meaning?' even more of a challenge.

Recently, Coecke et al. (2010) used high level cross-disciplinary techniques from logic, category

1394

theory, and physics to bring the above two approaches together. They developed a unified mathematical framework whereby a sentence vector is by definition a function of the Kronecker product of its word vectors. A concrete instantiation of this theory was exemplified on a toy hand crafted corpus by Grefenstette et al. (2011). In this paper we implement it by training the model over the entire BNC. The highlight of our implementation is that words with relational types, such as verbs, adjectives, and adverbs are matrices that *act* on their arguments. We provide a general algorithm for building (or indeed learning) these matrices from the corpus.

The implementation is evaluated against the task provided by Mitchell and Lapata (2008) for disambiguating intransitive verbs, as well as a similar new experiment for transitive verbs. Our model improves on the best method evaluated in Mitchell and Lapata (2008) and offers promising results for the transitive case, demonstrating its scalability in comparison to that of other models. But we still feel there is need for a different class of experiments to showcase merits of compositionality in a statistically significant manner. Our work shows that the categorical compositional distributional model of meaning permits a practical implementation and that this opens the way to the production of large scale compositional models.

2 Two Orthogonal Semantic Models

Formal Semantics To compute the meaning of a sentence consisting of n words, meanings of these words must *interact* with one another. In formal semantics, this further interaction is represented as a function derived from the grammatical structure of the sentence, but meanings of words are amorphous objects of the domain: no distinction is made between words that have the same type. Such models consist of a pairing of syntactic interpretation rules (in the form of a grammar) with semantic interpretation rules, as exemplified by the simple model presented in Figure 1.

The parse of a sentence such as "cats like milk" typically produces its semantic interpretation by substituting semantic representation for their grammatical constituents and applying β-reduction where needed. Such a derivation is shown in Figure 2.

Syntactic Analysis	Semantic Interpretation
S → NP VP	$|VP|(|NP|)$
NP → cats, milk, etc.	$|\text{cats}|, |\text{milk}|, \ldots$
VP → Vt NP	$|Vt|(|NP|)$
Vt → like, hug, etc.	$\lambda yx.|\text{like}|(x, y), \ldots$

Figure 1: A simple model of formal semantics.

$$|\text{like}|(|\text{cats}|, |\text{milk}|)$$

$$|\text{cats}| \qquad \lambda x.|\text{like}|(x, |\text{milk}|)$$

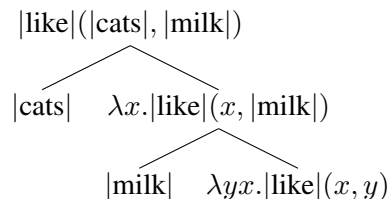$$|\text{milk}| \qquad \lambda yx.|\text{like}|(x, y)$$

Figure 2: A parse tree showing a semantic derivation.

This methodology is used to translate sentences of natural language into logical formulae, then use computer-aided automation tools to reason about them (Alshawi, 1992). One major drawback is that the result of such analysis can only deal with truth or falsity as the meaning of a sentence, and says nothing about the closeness in meaning or topic of expressions beyond their truth-conditions and what models satisfy them, hence do not perform well on language tasks such as search. Furthermore, an underlying domain of objects and a valuation function must be provided, as with any logic, leaving open the question of how we might *learn* the meaning of language using such a model, rather than just use it.

Distributional Models Distributional models of semantics, on the other hand, dismiss the interaction between syntactically linked words and are solely concerned with lexical semantics. Word meaning is obtained empirically by examining the *contexts*[1] in which a word appears, and equating the meaning of a word with the distribution of contexts it shares. The intuition is that context of use is what we appeal to in learning the meaning of a word, and that words that frequently have the same sort of context in common are likely to be semantically related.

For instance, beer and sherry are both drinks, alcoholic, and often cause a hangover. We expect these facts to be reflected in a sufficiently large corpus: the words 'beer' and 'sherry' occur within the

---

[1]*E.g.* words which appear in the same sentence or n-word window, or words which hold particular grammatical or dependency relations to the word being learned.

context of identifying words such as 'drink', 'alcoholic' and 'hangover' more frequently than they occur with other content words.

Such context distributions can be encoded as vectors in a high dimensional space with contexts as basis vectors. For any word vector $\overrightarrow{word}$, the scalar weight c_i^{word} associated with each context basis vector $\overrightarrow{n_i}$ is a function of the number of times the word has appeared in that context. Semantic vectors $(c_1^{word}, c_2^{word}, \cdots, c_n^{word})$ are also denoted by sums of such weight/basis vector pairs:

$$\overrightarrow{word} = \sum_i c_i^{word} \overrightarrow{n_i}$$

Learning a semantic vector is just learning its basis weights from the corpus. This setting offers geometric means to reason about semantic similarity (*e.g.* via cosine measure or k-means clustering), as discussed in Widdows (2005).

The principal drawback of such models is their non-compositional nature: they ignore grammatical structure and logical words, and hence cannot compute the meanings of phrases and sentences in the same efficient way that they do for words. Common operations discussed in (Mitchell and Lapata, 2008) such as vector addition ($+$) and component-wise multiplication ($\odot$, *cf.* §4 for details) are commutative, hence if $\overrightarrow{vw} = \overrightarrow{v} + \overrightarrow{w}$ or $\overrightarrow{v} \odot \overrightarrow{w}$, then $\overrightarrow{vw} = \overrightarrow{wv}$, leading to unwelcome equalities such as

$$\overrightarrow{\text{the dog bit the man}} = \overrightarrow{\text{the man bit the dog}}$$

Non-commutative operations, such as the Kronecker product (*cf.* §4 for definition) can take word-order into account (Smolensky, 1990) or even some more complex syntactic relations, as described in Clark and Pulman (2007). However, the dimensionality of sentence vectors produced in this manner differs for sentences of different length, barring all sentences from being compared in the same vector space, and growing exponentially with sentence length hence quickly becoming computationally intractable.

3 A Hybrid Logico-Distributional Model

Whereas semantic compositional mechanisms for set-theoretic constructions are well understood, there are no obvious corresponding methods for vector spaces. To solve this problem, Coecke et al.

(2010) use the abstract setting of category theory to turn the grammatical structure of a sentence into a morphism compatible with the higher level logical structure of vector spaces.

One pragmatic consequence of this abstract idea is as follows. In distributional models, there is a meaning vector for each word, *e.g.* $\overrightarrow{cats}$, $\overrightarrow{like}$, and $\overrightarrow{milk}$. The logical recipe tells us to *apply* the meaning of the verb to the meanings of subject and object. But how can a vector *apply* to other vectors? The solution proposed above implies that one needs to have different levels of meaning for words with different types. This is similar to logical models where verbs are relations and nouns are atomic sets. So verb vectors should be built differently from noun vectors, for instance as matrices.

The general information as to which words should be matrices and which words atomic vectors is in fact encoded in the type-logical representation of the grammatical structure of the sentence. This is the linear map with word vectors as input and sentence vectors as output. Hence, at least theoretically, one should be able to build sentence vectors and compare their synonymity in exactly the same way as one measures word synonymity.

Pregroup Grammars The aforementioned linear maps turn out to be the grammatical reductions of a type-logic called a Lambek pregroup grammar (Lambek, 2008)[2]. Pregroups and vector spaces share the same high level mathematical structure, referred to as a *compact closed category*, for a proof and details of this claim see Coecke et al. (2010); for a friendly introduction to category theory, see Coecke and Paquette (2011). One consequence of this parity is that the grammatical reductions of a pregroup grammar can be directly transformed into linear maps that act on vectors.

In a nutshell, pregroup types are either atomic or compound. Atomic types can be simple (*e.g.* n for noun phrases, s for statements) or left/right superscripted—referred to as adjoint types (*e.g.* n^r and n^l). An example of a compound type is that of a verb $n^r s n^l$. The superscripted types express that the verb is a relation with two arguments of type n,

---

[2]The usage of pregroup types is not essential, the types of any other logic, for instance CCG can be used, but should be translated into the language of pregroups.

which have to occur to the *right* and to the *left* of it, and that it outputs an argument of the type s. A transitive sentence has types as shown in Figure 3.

Each type n cancels out with its right adjoint n^r from the right and its left adjoint n^l from the left; mathematically speaking these mean[3]

$$n^l n \leq 1 \qquad \text{and} \qquad nn^r \leq 1$$

Here 1 is the unit of concatenation: $1n = n1 = n$. The corresponding grammatical reduction of a transitive sentence is $nn^r sn^l \leq 1s1 = s$. Each such reduction can be depicted as a wire diagram. The diagram of a transitive sentence is shown in Figure 3.
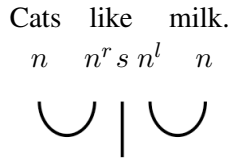

Figure 3: The pregroup types and reduction diagram for a transitive sentence.

Syntax-guided Semantic Composition According to Coecke et al. (2010) and based on a general completeness theorem between compact categories, wire diagrams, and vector spaces, the meaning of sentences can be canonically reduced to linear algebraic formulae. The following is the meaning vector of our transitive sentence:

$$\overrightarrow{\text{cats like milk}} = (f)\left(\overrightarrow{\text{cats}} \otimes \overrightarrow{\text{like}} \otimes \overrightarrow{\text{milk}}\right) \qquad \textbf{(I)}$$

Here f is the linear map that encodes the grammatical structure. The categorical morphism corresponding to it is denoted by the tensor product of 3 components: $\epsilon_V \otimes 1_S \otimes \epsilon_W$, where V and W are subject and object spaces, S is the sentence space, the ϵ's are the cups, and 1_S is the straight line in the diagram. The cups stand for taking inner products, which when done with the basis vectors imitate substitution. The straight line stands for the identity map that does nothing. By the rules of the category, equation **(I)** reduces to the following linear algebraic formula with

---

[3]The relation $\leq$ is the partial order of the pregroup. It corresponds to implication $\implies$ in a logical reading thereof. If these inequalities are replaced by equalities, *i.e.* if $n^l n = 1 = nn^r$, then the pregroup collapses into a group where $n^l = n^r$.

lower dimensions, hence the dimensional explosion problem for Kronecker products is avoided:

$$\sum_{itj} c_{itj} \langle \overrightarrow{\text{cats}} | \overrightarrow{v_i} \rangle \overrightarrow{s_t} \langle \overrightarrow{w_j} | \overrightarrow{\text{milk}} \rangle \in S \qquad \textbf{(II)}$$

$\overrightarrow{v_i}, \overrightarrow{w_j}$ are basis vectors of V and W. The inner product $\langle \overrightarrow{\text{cats}} | \overrightarrow{v_i} \rangle$ substitutes the weights of $\overrightarrow{\text{cats}}$ into the first argument place of the verb (similarly for object and second argument place). $\overrightarrow{s_t}$ is a basis vector of the sentence space S in which meanings of sentences live, regardless of their grammatical structure.

The degree of synonymity of sentences is obtained by taking the cosine measure of their vectors. S is an abstract space: it needs to be instantiated to provide concrete meanings and synonymity measures. For instance, a truth-theoretic model is obtained by taking the sentence space S to be the 2-dimensional space with basis vectors $|1\rangle$ (True) and $|0\rangle$ (False).

4 Building Matrices for Relational Words

In this section we present a general scheme to build matrices for relational words. Recall that given a vector space A with basis $\{\overrightarrow{n_i}\}_i$, the Kronecker product of two vectors $\overrightarrow{v} = \sum_i c_i^a \overrightarrow{n_i}$ and $\overrightarrow{w} = \sum_i c_i^b \overrightarrow{n_i}$ is defined as follows:

$$\overrightarrow{v} \otimes \overrightarrow{w} = \sum_{ij} c_i^a c_j^b (\overrightarrow{n_i} \otimes \overrightarrow{n_j})$$

where $(\overrightarrow{n_i} \otimes \overrightarrow{n_j})$ is just the pairing of the basis of A, *i.e.* $(\overrightarrow{n_i}, \overrightarrow{n_j})$. The Kronecker product vectors belong in the tensor product of A with itself: $A \otimes A$, hence if A has dimension r, these will be of dimensionality $r \times r$. The point-wise multiplication of these vectors is defined as follows

$$\overrightarrow{v} \odot \overrightarrow{w} = \sum_i c_i^a c_i^b \overrightarrow{n_i}$$

The intuition behind having a matrix for a relational word is that any relation R on sets X and Y, *i.e.* $R \subseteq X \times Y$ can be represented as a matrix, namely one that has as row-bases $x \in X$ and as column-bases $y \in Y$, with weight $c_{xy} = 1$ where $(x, y) \in R$ and 0 otherwise. In a distributional setting, the weights, which are natural or real numbers,

will represent more: 'the extent according to which x and y are related'. This can be determined in different ways.

Suppose X is the set of animals, and 'chase' is a relation on it: $chase \subseteq X \times X$. Take $x = $ 'dog' and $y = $ 'cat': with our type-logical glasses on, the obvious choice would be to take c_{xy} to be the number of times 'dog' has chased 'cat', *i.e.* the number of times the sentence 'the dog chases the cat' has appeared in the corpus. But in the distributional setting, this method will be too syntactic and dismissive of the actual *meaning* of 'cat' and 'dog'. If instead the corpus contains the sentence 'the hound hunted the wild cat', c_{xy} will be 0, restricting us to only assign meaning to sentences that have directly appeared in the corpus. We propose to, instead, use a level of abstraction by taking words such as verbs to be distributions over the semantic information in the vectors of their context words, rather than over the context words themselves.

Start with an r-dimensional vector space N with basis $\{\overrightarrow{n}_i\}_i$, in which meaning vectors of atomic words, such as nouns, live. The basis vectors of N are in principle all the words from the corpus, however in practice and following Mitchell and Lapata (2008) we had to restrict these to a subset of the most occurring words. These basis vectors are not restricted to nouns: they can as well be verbs, adjectives, and adverbs, so that we can define the meaning of a noun in all possible contexts—as is usual in context-based models—and not only in the context of other nouns. Note that basis words with relational types are treated as pure lexical items rather than as semantic objects represented as matrices. In short, we count how many times a noun has occurred close to words of other syntactic types such as 'elect' and 'scientific', rather than count how many times it has occurred close to their corresponding matrices: it is the lexical tokens that form the context, not their meaning.

Each relational word P with grammatical type π and m adjoint types $\alpha_1, \alpha_2, \cdots, \alpha_m$ is encoded as an $(r \times \ldots \times r)$ matrix with m dimensions. Since our vector space N has a fixed basis, each such ma-

trix is represented in vector form as follows:

$$\overrightarrow{\mathrm{P}} = \underbrace{\sum_{ij\cdots\zeta}}_{m} c_{ij\cdots\zeta} \underbrace{(\overrightarrow{n}_i \otimes \overrightarrow{n}_j \otimes \cdots \otimes \overrightarrow{n}_\zeta)}_{m}$$

This vector lives in the tensor space $\underbrace{N \otimes N \otimes \cdots \otimes N}_{m}$. Each $c_{ij\cdots\zeta}$ is computed according to the procedure described in Figure 4.

---

1) Consider a sequence of words containing a relational word 'P' and its arguments w_1, w_2, $\cdots$, w_m, occurring in the same order as described in P's grammatical type π. Refer to these sequences as 'P'-relations. Suppose there are k of them.
2) Retrieve the vector $\overrightarrow{w}_l$ of each argument w_l.
3) Suppose w_1 has weight c_i^1 on basis vector $\overrightarrow{n}_i$, w_2 has weight c_j^2 on basis vector $\overrightarrow{n}_j$, $\cdots$, and w_m has weight c_ζ^m on basis vector $\overrightarrow{n}_\zeta$. Multiply these weights

$$c_i^1 \times c_j^2 \times \cdots \times c_\zeta^m$$

4) Repeat the above steps for all the k 'P'-relations, and sum[a] the corresponding weights

$$c_{ij\cdots\zeta} = \sum_k \left(c_i^1 \times c_j^2 \times \cdots \times c_\zeta^m \right)_k$$

---
[a]We also experimented with multiplication, but the sparsity of noun vectors resulted in most verb matrices being empty.

Figure 4: Procedure for learning weights for matrices of words 'P' with relational types π of m arguments.

Linear algebraically, this procedure corresponds to computing the following

$$\overrightarrow{\mathrm{P}} = \sum_k \left(\overrightarrow{w}_1 \otimes \overrightarrow{w}_2 \otimes \cdots \otimes \overrightarrow{w}_m \right)_k$$

Type-logical examples of relational words are verbs, adjectives, and adverbs. A transitive verb is represented as a 2 dimensional matrix since its type is $n^r s n^l$ with two adjoint types n^r and n^l. The corresponding vector of this matrix is

$$\overrightarrow{\mathrm{verb}} = \sum_{ij} c_{ij} \left(\overrightarrow{n}_i \otimes \overrightarrow{n}_j \right)$$

The weight c_{ij} corresponding to basis vector $\overrightarrow{n}_i \otimes \overrightarrow{n}_j$, is the extent according to which words that have co-occurred with $\overrightarrow{n}_i$ have been the subject of the 'verb' and words that have co-occurred with $\overrightarrow{n}_j$ have been the object of the 'verb'. This example computation is demonstrated in Figure 5.

---

1) Consider phrases containing 'verb', its subject w_1 and object w_2. Suppose there are k of them.
2) Retrieve vectors $\overrightarrow{w}_1$ and $\overrightarrow{w}_2$.
3) Suppose $\overrightarrow{w}_1$ has weight c_i^1 on $\overrightarrow{n}_i$ and $\overrightarrow{w}_2$ has c_j^2 on $\overrightarrow{n}_j$. Multiply these weights $c_i^1 \times c_j^2$.
4) Repeat the above steps for all k 'verb'-relations and sum the corresponding weights $\sum_k (c_i^1 \times c_j^2)_k$

Figure 5: Procedure for learning weights for matrices of transitive verbs.

Linear algebraically, we are computing

$$\overrightarrow{verb} \quad = \quad \sum_k \left(\overrightarrow{w}_1 \otimes \overrightarrow{w}_2 \right)_k$$

As an example, consider the verb 'show' and suppose there are two 'show'-relations in the corpus:

$s_1 \quad = \quad$ table show result
$s_2 \quad = \quad$ map show location

The vector of 'show' is

$$\overrightarrow{show} \quad = \quad \overrightarrow{table} \otimes \overrightarrow{result} \ + \ \overrightarrow{map} \otimes \overrightarrow{location}$$

Consider an N space with four basis vectors 'far', 'room', 'scientific', and 'elect'. The TF/IDF-weighted values for vectors of the above four nouns (built from the BNC) are as shown in Table 1.

i	$\overrightarrow{n_i}$	table	map	result	location
1	far	6.6	5.6	7	5.9
2	room	27	7.4	0.99	7.3
3	scientific	0	5.4	13	6.1
4	elect	0	0	4.2	0

Table 1: Sample weights for selected noun vectors.

Part of the matrix of 'show' is presented in Table 2.

As a sample computation, the weight c_{11} for vector $(1,1)$, *i.e.* $(\overrightarrow{far}, \overrightarrow{far})$ is computed by multiplying weights of 'table' and 'result' on far, *i.e.* 6.6×7,

	far	room	scientific	elect
far	79.24	47.41	119.96	27.72
room	232.66	80.75	396.14	113.2
scientific	32.94	31.86	32.94	0
elect	0	0	0	0

Table 2: Sample semantic matrix for 'show'.

multiplying weights of 'map' and 'location' on $\overrightarrow{far}$, *i.e.* 5.6×5.9 then adding these $46.2 + 33.04$ and obtaining the total weight 79.24.

The same method is applied to build matrices for ditransitive verbs, which will have 3 dimensions, and adjectives and adverbs, which will be of 1 dimension each.

5 Computing Sentence Vectors

Meaning of sentences are vectors computed by taking the variables of the categorical prescription of meaning (the linear map f obtained from the grammatical reduction of the sentence) to be determined by the matrices of the relational words. For instance the meaning of the transitive sentence 'sub verb obj' is:

$$\overrightarrow{sub\ verb\ obj} = \sum_{itj} \langle \overrightarrow{sub} \mid \overrightarrow{v}_i \rangle \langle \overrightarrow{w}_j \mid \overrightarrow{obj} \rangle c_{itj} \overrightarrow{s}_t$$

We take $V := W := N$ and $S = N \otimes N$, then $\sum_{itj} c_{itj} \overrightarrow{s}_t$ is determined by the matrix of the verb, *i.e.* substitute it by $\sum_{ij} c_{ij}(\overrightarrow{n}_i \otimes \overrightarrow{n}_j)^4$. Hence $\overrightarrow{sub\ verb\ obj}$ becomes:

$$\sum_{ij} \langle \overrightarrow{sub} \mid \overrightarrow{n}_i \rangle \langle \overrightarrow{n}_j \mid \overrightarrow{obj} \rangle c_{ij}(\overrightarrow{n}_i \otimes \overrightarrow{n}_j) \quad =$$

$$\sum_{ij} c_i^{sub} c_j^{obj} c_{ij}(\overrightarrow{n}_i \otimes \overrightarrow{n}_j)$$

This can be decomposed to point-wise multiplication of two vectors as follows:

$$\left(\sum_{ij} c_i^{sub} c_j^{obj}(\overrightarrow{n}_i \otimes \overrightarrow{n}_j) \right) \odot \left(\sum_{ij} c_{ij}(\overrightarrow{n}_i \otimes \overrightarrow{n}_j) \right)$$

---

[4]Note that by doing so we are also reducing the verb space from $N \otimes (N \otimes N) \otimes N$ to $N \otimes N$, since for our construction we only need tuples of the form $\overrightarrow{n}_i \otimes \overrightarrow{n}_i \otimes \overrightarrow{n}_j \otimes \overrightarrow{n}_j$ which are isomorphic to pairs $(\overrightarrow{n}_i \otimes \overrightarrow{n}_j)$.

The left argument is the Kronecker product of subject and object vectors and the right argument is the vector of the verb, so we obtain

$$\left(\overrightarrow{\text{sub}} \otimes \overrightarrow{\text{obj}} \right) \odot \overrightarrow{\text{verb}}$$

Since $\odot$ is commutative, this provides us with a distributional version of the type-logical meaning of the sentence: point-wise multiplication of the meaning of the verb to the Kronecker product of its subject and object:

$$\overrightarrow{\text{sub verb obj}} \quad = \quad \overrightarrow{\text{verb}} \odot \left(\overrightarrow{\text{sub}} \otimes \overrightarrow{\text{obj}} \right)$$

This mathematical operation can be informally described as a structured 'mixing' of the information of the subject and object, followed by it being 'filtered' through the information of the verb applied to them, in order to produce the information of the sentence.

In the transitive case, $S = N \otimes N$, hence $\overrightarrow{s}_t = \overrightarrow{n}_i \otimes \overrightarrow{n}_j$. More generally, the vector space corresponding to the abstract sentence space S is the concrete tensor space $(N \otimes \ldots \otimes N)$ for m the dimension of the matrix of the 'verb'. As we have seen above, in practice we do not need to build this tensor space, as the computations thereof reduce to point-wise multiplications and summations.

Similar computations yield meanings of sentences with adjectives and adverbs. For instance the meaning of a transitive sentence with a modified subject and a modified verb we have

$$\overrightarrow{\text{adj sub verb obj adv}} =$$
$$\left(\overrightarrow{\text{adv}} \odot \overrightarrow{\text{verb}} \right) \odot \left(\left(\overrightarrow{\text{adj}} \odot \overrightarrow{\text{sub}} \right) \otimes \overrightarrow{\text{obj}} \right)$$

After building vectors for sentences, we can compare their meaning and measure their degree of synonymy by taking their cosine measure.

6 Evaluation

Evaluating such a framework is no easy task. What to evaluate depends heavily on what sort of application a practical instantiation of the model is geared towards. In (Grefenstette et al., 2011), it is suggested that the simplified model we presented and expanded here could be evaluated in the same way as lexical semantic models, measuring compositionally

built sentence vectors against a benchmark dataset such as that provided by Mitchell and Lapata (2008). In this section, we briefly describe the evaluation of our model against this dataset. Following this, we present a new evaluation task extending the experimental methodology of Mitchell and Lapata (2008) to transitive verb-centric sentences, and compare our model to those discussed by Mitchell and Lapata (2008) within this new experiment.

First Dataset Description   The first experiment, described in detail by Mitchell and Lapata (2008), evaluates how well compositional models disambiguate ambiguous words given the context of a potentially disambiguating noun. Each entry of the dataset provides a noun, a target verb and landmark verb (both intransitive). The noun must be composed with both verbs to produce short phrase vectors the similarity of which is measured by the candidate. Also provided with each entry is a classification ("High" or "Low") indicating whether or not the verbs are indeed semantically close within the context of the noun, as well as an evaluator-set similarity score between 1 and 7 (along with an evaluator identifier), where 1 is low similarity and 7 is high.

Evaluation Methodology   Candidate models provide a similarity score for each entry. The scores of high similarity entries and low similarity entries are averaged to produce a mean High score and mean Low score for the model. The correlation of the model's similarity judgements with the human judgements is also calculated using Spearman's ρ, a metric which is deemed to be more scrupulous, and ultimately that by which models should be ranked, by Mitchell and Lapata (2008). The mean for each model is on a $[0, 1]$ scale, except for UpperBound which is on the same $[1, 7]$ scale the annotators used. The ρ scores are on a $[-1, 1]$ scale. It is assumed that inter-annotator agreement provides the theoretical maximum ρ for any model for this experiment. The cosine measure of the verb vectors, ignoring the noun, is taken to be the baseline (no composition).

Other Models   The other models we compare ours to are those evaluated by Mitchell and Lapata (2008). We provide a selection of the results

1400

from that paper for the worst (Add) and best[5] (Multiply) performing models, as well as the previous second-best performing model (Kintsch). The additive and multiplicative models are simply applications of vector addition and component-wise multiplication. We invite the reader to consult (Mitchell and Lapata, 2008) for the description of Kintsch's additive model and parametric choices.

Model Parameters   To provide the most accurate comparison with the existing multiplicative model, and exploiting the aforementioned feature that the categorical model can be built "on top of" existing lexical distributional models, we used the parameters described by Mitchell and Lapata (2008) to reproduce the vectors evaluated in the original experiment as our noun vectors. All vectors were built from a lemmatised version of the BNC. The noun basis was the 2000 most common context words, basis weights were the probability of context words given the target word divided by the overall probability of the context word. Intransitive verb function-vectors were trained using the procedure presented in §4. Since the dataset only contains intransitive verbs and nouns, we used $S = N$. The cosine measure of vectors was used as a similarity metric.

First Experiment Results   In Table 3 we present the comparison of the selected models. Our categorical model performs significantly better than the existing second-place (Kintsch) and obtains a ρ quasi-identical to the multiplicative model, indicating significant correlation with the annotator scores.

There is not a large difference between the mean High score and mean Low score, but the distribution in Figure 6 shows that our model makes a non-negligible distinction between high similarity phrases and low similarity phrases, despite the absolute scores not being different by more than a few percentiles.

---

[5] The multiplicative model presented here is what is qualified as best in (Mitchell and Lapata, 2008). However, they also present a slightly better performing ($\rho = 0.19$) model which is a combination of their multiplicative model and a weighted additive model. The difference in ρ is qualified as "not statistically significant" in the original paper, and furthermore the mixed model requires parametric optimisation hence was not evaluated against the entire test set. For these reasons, we chose not to include it in the comparison.

Model	High	Low	ρ
Baseline	0.27	0.26	0.08
Add	0.59	0.59	0.04
Kintsch	0.47	0.45	0.09
Multiply	0.42	0.28	0.17
Categorical	**0.84**	**0.79**	**0.17**
UpperBound	4.94	3.25	0.40

Table 3: Selected model means for High and Low similarity items and correlation coefficients with human judgements, first experiment (Mitchell and Lapata, 2008). $p < 0.05$ for each ρ.
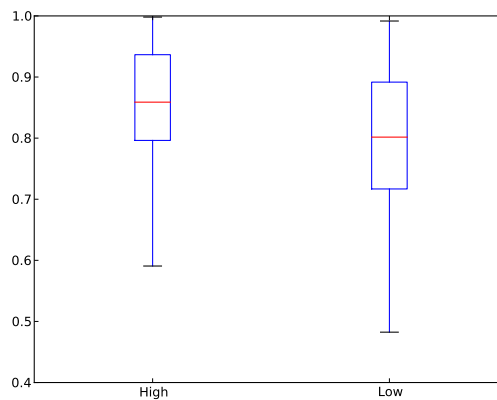

Figure 6: Distribution of predicted similarities for the categorical distributional model on High and Low similarity items.

Second Dataset Description   The second dataset[6], developed by the authors, follows the format of the (Mitchell and Lapata, 2008) dataset used for the first experiment, with the exception that the target and landmark verbs are transitive, and an object noun is provided in addition to the subject noun, hence forming a small transitive sentence. The dataset comprises 200 entries consisting of sentence pairs (hence a total of 400 sentences) constructed by following the procedure outlined in §4 of (Mitchell and Lapata, 2008), using transitive verbs from CELEX[7]. For examples of these sentences, see Table 4. The dataset was split into four sections of 100 entries each, with guaranteed 50% exclusive overlap with

---

[6] http://www.cs.ox.ac.uk/activities/CompDistMeaning/GS2011data.txt
[7] http://celex.mpi.nl/

exactly two other datasets. Each section was given to a group of evaluators, with a total of 25, who were asked to form simple transitive sentence pairs from the verbs, subject and object provided in each entry; for instance 'the table showed the result' from 'table show result'. The evaluators were then asked to rate the semantic similarity of each verb pair within the context of those sentences, and offer a score between 1 and 7 for each entry. Each entry was given an arbitrary classification of HIGH or LOW by the authors, for the purpose of calculating mean high/low scores for each model. For example, the first two pairs in table 4 were classified as HIGH, whereas the second two pairs as LOW.

Sentence 1	Sentence 2
table show result	table express result
map show location	map picture location
table show result	table picture result
map show location	map express location

Table 4: Example entries from the transitive dataset without annotator score, second experiment.

Evaluation Methodology The evaluation methodology for the second experiment was identical to that of the first, as are the scales for means and scores. Here also, Spearman's ρ is deemed a more rigorous way of determining how well a model tracks difference in meaning. This is both because of the imprecise nature of the classification of verb pairs as HIGH or LOW; and since the objective similarity scores produced by a model that distinguishes sentences of different meaning from those of similar meaning can be renormalised in practice. Therefore the delta between HIGH means and LOW mean cannot serve as a definite indication of the practical applicability (or lack thereof) of semantic models; the means are provided just to aid comparison with the results of the first experiment.

Model Parameters As in the first experiment, the lexical vectors from (Mitchell and Lapata, 2008) were used for the other models evaluated (additive, multiplicative and baseline)[8] and for the noun vec-

---

[8]Kintsch was not evaluated as it required optimising model parameters against a held-out segment of the test set, and we could not replicate the methodology of Mitchell and Lapata

tors of our categorical model. Transitive verb vectors were trained as described in §4 with $S = N \otimes N$.

Second Experiment Results The results for the models evaluated against the second dataset are presented in Table 5.

Model	High	Low	ρ
Baseline	0.47	0.44	0.16
Add	0.90	0.90	0.05
Multiply	0.67	0.59	0.17
Categorical	**0.73**	**0.72**	**0.21**
UpperBound	4.80	2.49	0.62

Table 5: Selected model means for High and Low similarity items and correlation coefficients with human judgements, second experiment. $p < 0.05$ for each ρ.

We observe a significant (according to $p < 0.0.5$) improvement in the alignment of our categorical model with the human judgements, from 0.17 to 0.21. The additive model continues to make little distinction between senses of the verb during composition, and the multiplicative model's alignment does not change, but becomes statistically indistinguishable from the non-compositional baseline model.

Once again we note that the high-low means are not very indicative of model performance, as the difference between high mean and the low mean of the categorical model is much smaller than that of the both the baseline model and multiplicative model, despite better alignment with annotator judgements.

7 Discussion

In this paper, we described an implementation of the categorical model of meaning (Coecke et al., 2010), which combines the formal logical and the empirical distributional frameworks into a unified semantic model. The implementation is based on building matrices for words with relational types (adjectives, verbs), and vectors for words with atomic types (nouns), based on data from the BNC. We then show how to apply verbs to their subject/object, in order to compute the meaning of intransitive and transitive sentences.

(2008) with full confidence.

1402

Other work uses matrices to model meaning (Baroni and Zamparelli, 2010; Guevara, 2010), but only for adjective-noun phrases. Our approach easily applies to such compositions, as well as to sentences containing combinations of adjectives, nouns, verbs, and adverbs. The other key difference is that they learn their matrices in a top-down fashion, *i.e.* by regression from the composite adjective-noun context vectors, whereas our model is bottom-up: it learns sentence/phrase meaning compositionally from the vectors of the compartments of the composites. Finally, very similar functions, for example a verb with argument alternations such as 'break' in 'Y breaks' and 'X breaks Y', are not treated as unrelated. The matrix of the intransitive 'break' uses the corpus-observed information about the subject of break, including that of 'Y', similarly the matrix of the transitive 'break' uses information about its subject and object, including that of 'X' and 'Y'. We leave a thorough study of these phenomena, which fall under providing a modular representation of passive-active similarities, to future work.

We evaluated our model in two ways: first against the word disambiguation task of Mitchell and Lapata (2008) for intransitive verbs, and then against a similar new experiment for transitive verbs, which we developed.

Our findings in the first experiment show that the categorical method performs on par with the leading existing approaches. This should not surprise us given that the context is so small and our method becomes similar to the multiplicative model of Mitchell and Lapata (2008). However, our approach is sensitive to grammatical structure, leading us to develop a second experiment taking this into account and differentiating it from models with commutative composition operations.

The second experiment's results deliver the expected qualitative difference between models, with our categorical model outperforming the others and showing an increase in alignment with human judgements in correlation with the increase in sentence complexity. We use this second evaluation principally to show that there is a strong case for the development of more complex experiments measuring not only the disambiguating qualities of compositional models, but also their syntactic sensitivity, which is not directly measured in the existing experiments.

These results show that the high level categorical distributional model, uniting empirical data with logical form, can be implemented just like any other concrete model. Furthermore it shows better results in experiments involving higher syntactic complexity. This is just the tip of the iceberg: the mathematics underlying the implementation ensures that it uniformly scales to larger, more complicated sentences and enables it to compare synonymity of sentences that are of different grammatical structure.

8 Future Work

Treatment of function words such as 'that', 'who', as well as logical words such as quantifiers and conjunctives are left to future work. This will build alongside the general guidelines of Coecke et al. (2010) and concrete insights from the work of Widdows (2005). It is not yet entirely clear how existing set-theoretic approaches, for example that of discourse representation and generalised quantifiers, apply to our setting. Preliminary work on integration of the two has been presented by Preller (2007) and more recently also by Preller and Sadrzadeh (2009).

As mentioned by one of the reviewers, our pre-group approach to grammar flattens the sentence representation, in that the verb is applied to its subject and object at the same time; whereas in other approaches such as CCG, it is first applied to the object to produce a verb phrase, then applied to the subject to produce the sentence. The advantages and disadvantages of this method and comparisons with other systems, in particular CCG, constitutes ongoing work.

9 Acknowledgement

References

H. Alshawi (ed). 1992. *The Core Language Engine*. MIT Press.

M. Baroni and R. Zamparelli. 2010. *Nouns are vectors, adjectives are matrices*. Proceedings of Conference

on Empirical Methods in Natural Language Processing (EMNLP).

S. Clark and S. Pulman. 2007. *Combining Symbolic and Distributional Models of Meaning*. Proceedings of AAAI Spring Symposium on Quantum Interaction. AAAI Press.

B. Coecke, and E. Paquette. 2011. *Categories for the Practicing Physicist. New Structures for Physics*, 167-271. B. Coecke (ed.). Lecture Notes in Physics **813**. Springer.

B. Coecke, M. Sadrzadeh and S. Clark. 2010. *Mathematical Foundations for Distributed Compositional Model of Meaning*. Lambek Festschrift. Linguistic Analysis **36**, 345–384. J. van Benthem, M. Moortgat and W. Buszkowski (eds.).

J. Curran. 2004. *From Distributional to Semantic Similarity*. PhD Thesis, University of Edinburgh.

K. Erk and S. Padó. 2004. *A Structured Vector Space Model for Word Meaning in Context*. Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP), 897–906.

G. Frege 1892. *Über Sinn und Bedeutung*. Zeitschrift für Philosophie und philosophische Kritik 100.

J. R. Firth. 1957. *A synopsis of linguistic theory 1930-1955*. Studies in Linguistic Analysis.

E. Grefenstette, M. Sadrzadeh, S. Clark, B. Coecke, S. Pulman. 2011. *Concrete Compositional Sentence Spaces for a Compositional Distributional Model of Meaning*. International Conference on Computational Semantics (IWCS'11). Oxford.

G. Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer.

E. Guevara. 2010. *A Regression Model of Adjective-Noun Compositionality in Distributional Semantics*. Proceedings of the ACL GEMS Workshop.

Z. S. Harris. 1966. *A Cycling Cancellation-Automaton for Sentence Well-Formedness*. International Computation Centre Bulletin **5**, 69–94.

R. Hudson. 1984. *Word Grammar*. Blackwell.

J. Lambek. 2008. *From Word to Sentence*. Polimetrica, Milan.

T. Landauer, and S. Dumais. 2008. *A solution to Platos problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge*. Psychological review.

C. D. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to information retrieval*. Cambridge University Press.

J. Mitchell and M. Lapata. 2008. *Vector-based models of semantic composition*. Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, 236–244.

R. Montague. 1974. *English as a formal language*. Formal Philosophy, 189–223.

J. Nivre. 2003. *An efficient algorithm for projective dependency parsing*. Proceedings of the 8th International Workshop on Parsing Technologies (IWPT).

A. Preller. *Towards Discourse Representation via Pregroup Grammars*. Journal of Logic Language Information **16** 173–194.

A. Preller and M. Sadrzadeh. *Semantic Vector Models and Functional Models for Pregroup Grammars*. Journal of Logic Language Information. DOI: 10.1007/s10849-011-9132-2. to appear.

J. Saffron, E. Newport, R. Asling. 1999. *Word Segmentation: The role of distributional cues*. Journal of Memory and Language **35**, 606–621.

H. Schuetze. 1998. *Automatic Word Sense Discrimination*. Computational Linguistics **24**, 97–123.

P. Smolensky. 1990. *Tensor product variable binding and the representation of symbolic structures in connectionist systems*. Computational Linguistics **46, 1–2**, 159–216.

M. Steedman. 2000. *The Syntactic Process*. MIT Press.

D. Widdows. 2005. *Geometry and Meaning*. University of Chicago Press.

L. Wittgenstein. 1953. *Philosophical Investigations*. Blackwell.