A Unified Sentence Space for Categorical Distributional-Compositional Semantics: Theory and Experiments

Dimitri KARTSAKLIS   Mehrnoosh SADRZADEH   Stephen PULMAN[*]
DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF OXFORD
Wolfson Building, Parks Road, Oxford OX1 3QD, UK
`firstname.lastname@cs.ox.ac.uk`

ABSTRACT
This short paper summarizes a faithful implementation of the categorical framework of Coecke et al. (2010), the aim of which is to provide compositionality in distributional models of lexical semantics. Based on Frobenius Algebras, our method enable us to (1) have a unifying meaning space for phrases and sentences of different structure and word vectors, (2) stay faithful to the linguistic types suggested by the underlying type-logic, and (3) perform the concrete computations in lower dimensions by reducing the space complexity. We experiment with two different parameters of the model and apply the setting to a verb disambiguation and a term/definition classification task with promising results.

KEYWORDS: semantics, compositionality, distributional models, category theory, Frobenius algebra, vector space models, disambiguation, definition classification.

[*] Support by EPSRC grant EP/F042728/1 is gratefully acknowledged by the first two authors.

Proceedings of COLING 2012: Posters, pages 549–558,
COLING 2012, Mumbai, December 2012.

1 Introduction

Distributional models of meaning work by building co-occurrence vectors for every word in a corpus depending on its context, following Firth's intuition that "you should know a word by the company it keeps" (Firth, 1957). In such models, the co-occurrence vector of each word is built by fixing a set of words as the basis of a vector space and a window of size k, then counting how many times the word in question has co-occurred with each base in that window. This approach has been proved useful in many natural language tasks (Curran, 2004; Schütze, 1998; Landauer and Dumais, 1997; Manning et al., 2008), but until now it lacks any means of compositionality that would allow the combination of two word vectors into a new one following some grammar rule. In fact, compositional abilities of distributional models have been subject of much discussion and research in recent years. For example, Mitchell and Lapata (2008) present results for intransitive sentences, Erk and Padó (2004) work on transitive verb phrases, while Baroni and Zamparelli (2010) and Guevara (2010) provide comprehensive analyses of adjective-noun phrases. Despite the experimental strength of these approaches, most of them only deal with phrases and sentences of two words. On the other hand, Socher et al. (2010, 2011) use recursive neural networks in order to produce vectors for sentences of arbitrary length with good results. However, their method is somehow detached from the formal semantics view, paying little attention to the grammatical relations that hold between the words.

Following a different path, Coecke et al. (2010) provide a solution that offers compositional abilities to distributional models while at the same time avoids all the above pitfalls. Based on the abstract setting of category theory, the authors develop a generic mathematical framework whereby the meaning of a sentence of any length and structure can, in principle, be turned into a vector, following the rules of the grammar. Implementations of this model for transitive and intransitive sentences have been provided by Grefenstette and Sadrzadeh (2011a,b). However, although their method outperforms the multiplicative and additive models of Mitchell and Lapata (2008) on simple transitive sentences, it has a non-scalability problem. Specifically, the concrete structures used in the actual computations are not faithful to the linguistic types of the underlying type-logic, hence the model does not generalize to more complex phrases and sentences where a relational structure can be found nested in another relational structure. Furthermore, the vectors obtained for sentences of different grammatical structures live in different vector spaces: sentences with intransitive verbs live in the same space as context vectors, denoted by N, sentences with transitive verbs in $N^2 = N \otimes N$, and sentences with ditransitive verbs in N^3. A direct consequence of this instantiation is that one cannot compare meanings of sentences unless they have the same grammatical structure.

In this work we outline a solution to the above problems by instantiating the sentence space to be the same space as one in which context vectors live, namely we stipulate that $S = N$. As a result of this decision, we become able to compare lexical meanings of words with compositional meanings of phrases and sentences. We show how the theoretical computations of Coecke et al. (2010) instantiate in this concrete setting, and how the Frobenius Algebras, originating from group theory (Frobenius, 1903) and later extended to vector spaces (Coecke et al., 2008), allow us to not only represent meanings of words with complex roles, such as verbs, adjectives, and prepositions, in an intuitive relational manner, but also to stay faithful to their original linguistic types. Equally as importantly, this model enables us to realize the concrete computations in lower dimensional spaces, thus reduce the space complexity of the implementation.

We experiment in two different tasks with promising results: First, we repeat the disambiguation experiment of Grefenstette and Sadrzadeh (2011a) for transitive verbs. Then we proceed to a novel task: We use The Oxford Junior Dictionary (Sansome et al., 2000), Oxford Concise School Dictionary (Hawkins et al., 2004), and WordNet in order to derive a set of term/definition pairs, measure the similarity of each term with every definition, and use this measurement to classify the definitions to specific terms.

2 An overview of the categorical model

Using the abstract framework of category theory, Coecke et al. (2010) equip the distributional models of meaning with compositionality in a way that every grammatical reduction is in one-to-one correspondence with a linear map defining mathematical manipulations between vector spaces. In other words, given a sentence $s = w_1 w_2 \cdots w_n$ there exists a syntax-driven linear map f from the context vectors of the individual words to a vector for the whole sentence:

$$\overrightarrow{s} = f(\overrightarrow{w_1} \otimes \overrightarrow{w_2} \otimes \cdots \otimes \overrightarrow{w_n}) \tag{1}$$

allowing us to compare the synonymy of two different sentences as if they were words, by constructing their vectors and measuring the distance between them. This result is based on the fact that the base type-logic of the framework, a *pregroup grammar* (Lambek, 2008), shares the same abstract structure with vector spaces, that of a *compact closed category*. If P is the free pregroup generated by such a grammar and **FVect** the category of finite dimensional vector spaces (with linear maps) over $\mathbb{R}$, it is possible then for one to work on the product category **FVect** $\times P$, pairing each grammatical type $\alpha \in P$ with a vector space V to an object (V, α). More importantly, the morphisms of this product category will be pairs of linear maps and pregroup reductions between these objects of the following form:

$$(f, \leq) : (V, p) \to (W, q) \tag{2}$$

leading from the grammatical type p and its corresponding vector space V to type q and the vector space W.

Pregroups A pregroup grammar (Lambek, 2008) is a type-logical grammar built on the rigorous mathematical basis of pregroups, i.e. partially ordered monoids with unit 1, whose each element p has a left adjoint p^l and a right adjoint p^r, that is

$$p^l p \leq 1 \leq p p^l \quad \text{and} \quad p p^r \leq 1 \leq p^r p \tag{3}$$

Each element p represents an atomic type of the grammar, for example n for noun phrases and s for sentences. Atomic types and their adjoints can be combined to form compound types, e.g. $n^r s n^l$ for a transitive verb. The rules of the grammar are prescribed by the mathematical properties of pregroups, and specifically by the inequalities in (3) above. A partial order in the context of a logic denotes implication, so from (3) we derive:

$$p^l p \to 1 \quad \text{and} \quad p p^r \to 1 \tag{4}$$

These cancellation rules to the unit object are called ϵ maps, and linear-algebraically correspond to the inner product between the involved context vectors. It also holds that $1p = p = p1$. We will use the case of a transitive sentence as an example. Here, the subject and the object have the type n, whereas the type of the verb is $n^r s n^l$, denoting that the verb looks for a noun at its left and a noun at its right in order to return an entity of type s (a sentence). The derivation has the form $n(n^r s n^l)n = (nn^r)s(n^l n) \to 1s1 = s$, and corresponds to the morphism $\epsilon_N \otimes 1_S \otimes \epsilon_N : N \otimes N \otimes S \otimes N \otimes N \to S$ which returns a vector living in S.

For details of pregroup grammars and its type dictionary we refer the reader to Lambek (2008). For more information about the compositional-distributional framework, see Coecke et al. (2010); Coecke and Paquette (2011) provide a good introduction to category theory.

3 Instantiating the sentence space

The categorical framework of Coecke et al. (2010) is abstract in the sense that it does not prescribe concrete guidelines for constructing tensors for meanings of words with special roles such as verbs or adjectives. Even more importantly, it does not specify the exact form of the sentence space S, leaving these details as open questions for the implementor.

3.1 Stipulating $S = N \otimes N$

The work of Grefenstette and Sadrzadeh (2011a) was the first large-scale practical implementation of this framework for intransitive and transitive sentences, and thus a first step towards providing some concrete answers to these questions. Following ideas from formal semantics that verbs are actually relations, the authors argue that the distributional meaning of a verb is a weighted relation representing the extent according to which the verb is related to its subjects and objects. In vector spaces, these relations are represented by linear maps, equivalent to matrices for the case of binary relations and to tensors for relations of arity n. Hence transitive verbs can be represented by matrices created by structurally mixing and summing up all the contexts (subject and object pairs) in which the verb appears. More precisely, we have:

$$\overline{verb} = \sum_i (\overrightarrow{sbj_i} \otimes \overrightarrow{obj_i}) \tag{5}$$

where $\overrightarrow{sbj_i}$ and $\overrightarrow{obj_i}$ are the context vectors of subject and object, respectively, and i iterates over all contexts in which the specific verb occurs. This method (which we refer to as "relational") is also extended to other relational words, such as adjectives whose vectors are constructed as the sum of all the nouns that the adjective modifies.

One important design decision was that the meaning of a sentence was represented as a rank-n tensor, where n is the number of arguments for the head word of the sentence. In other words, an intransitive sentence lives in a space $S = N$, a transitive one in $S = N \otimes N$ and so on. Although this approach delivers good results for the disambiguation task on which it was tested, it inherently suffers from two important problems, the most obvious of which is that there is no direct way to compare sentences of different structures, say an intransitive one with a transitive one. Furthermore, the representation of the meaning of a sentence or a phrase as a rank-n tensor with $n > 1$ limits the ability of the model to scale up to larger fragments of the language, where more complex sentences with nested or recursive structure can occur, since the concrete objects used in the actual mathematical operations are not any more faithful to the linguistic types. Finally, the above design decision means that the space complexity of the algorithm is $\Theta(d^n)$, where d is the cardinality of the vector space and n the number of arguments for the head word. This could create certain space problems for complex sentences.

3.2 Stipulating $S = N$

The work presented in this paper stems from the observation that the theory does not impose a special choice of sentence space, in particular it does not impose that tensors for S should have ranks greater than 1. Hence we stipulate that $S = N$ and show how this instantiation works by performing the computations on the example transitive sentence 'dogs chase cats'. Take $\overrightarrow{dog}$ and $\overrightarrow{cat}$ be the context vectors for the subject and the object, both living in N as prescribed by their types. As any vector, these can be expressed as weighted sums of their basis vectors, that is, $\overrightarrow{dog} = \sum_i c_i^{dog} \overrightarrow{n_i}$ and $\overrightarrow{cat} = \sum_k c_k^{cat} \overrightarrow{n_k}$. On the other hand, the type of the verb indicates that this entity should live in N^3, represented by $\overline{chase} = \sum_{ijk} c_{ijk}^{chase} (\overrightarrow{n_i} \otimes \overrightarrow{n_j} \otimes \overrightarrow{n_k})$. By putting everything together, the meaning of the sentence is calculated as follows; this result lives in N, since it is a weighted sum over $\overrightarrow{n_j}$:

$$\epsilon_n^r \otimes 1_s \otimes \epsilon_n^l (\overrightarrow{dog} \otimes \overline{chase} \otimes \overrightarrow{cat}) = \sum_{ijk} c_{ijk}^{chase} \langle \overrightarrow{dog} | \overrightarrow{n_i} \rangle \langle \overrightarrow{n_k} | \overrightarrow{cat} \rangle \overrightarrow{n_j} \tag{6}$$

An important consequence of our design decision is that it enables us to reduce the space complexity of the implementation from $\Theta(d^n)$ (Grefenstette and Sadrzadeh, 2011a) to $\Theta(d)$, making the problem much more tractable. What remains to be solved is a theoretical issue, that in practice the meaning of relational words such as 'chase' as calculated by Equation 5 is a matrix living in N^2—however, the mathematical framework above prescribes that it should be a rank-3 tensor in N^3. The necessary expansions are achieved by using Frobenius algebraic operations, for which the following sections first provide the mathematical definitions and then a linguistic justification.

4 Frobenius Algebras

Frobenius algebras were originally introduced by F. G. Frobenius in group theory (Frobenius, 1903). Since then they have found applications in other fields of mathematics and physics, e.g. see Kock (2003). Carboni and Walters (1987) provided a general categorical definition, according to which a Frobenius algebra over a monoidal category $(\mathscr{C}, \otimes, I)$ is a tuple $(F, \sigma, \iota, \mu, \zeta)$ consisting of an associative coalgebra (σ, ι) and an associative algebra (μ, ζ), respectively given by the following types:

$$\sigma : F \to F \otimes F \qquad \iota : F \to I \qquad \mu : F \otimes F \to F \qquad \zeta : I \to F$$

The above should satisfy the *Frobenius condition*, stating that $(\mu \otimes 1_F) \circ (1_F \otimes \sigma) = (1_F \otimes \mu) \circ (\sigma \otimes 1_F) = \sigma \circ \mu$. For the case of the category **FVect** over a field I (for us $I = \mathbb{R}$), these morphisms become linear maps that form a Frobenius algebra over a vector space N with a fixed set of bases $\{\overrightarrow{n_i}\}_i$, explicitly given as follows (Coecke et al., 2008):

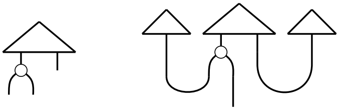
$$\sigma :: \overrightarrow{n_i} \mapsto \overrightarrow{n_i} \otimes \overrightarrow{n_i} \qquad \iota :: \overrightarrow{n_i} \mapsto 1 \qquad \mu :: \overrightarrow{n_i} \otimes \overrightarrow{n_i} \mapsto \overrightarrow{n_i} \qquad \zeta :: 1 \mapsto \overrightarrow{n_i}$$

Since the bases of our vector spaces are orthonormal, these maps moreover form a *special commutative* Frobenius algebra, meaning that they correspond to a uniform copying and uncopying of the basis vectors. When applied to $v \in N$, the copying map σ recovers the bases of v and the unit map ι their corresponding weights. Together, they faithfully encode tensors of a lower dimensional N into a higher dimensional tensor space $N \otimes N$. In linear algebraic terms, $\sigma(v)$ is a diagonal tensor whose diagonal elements consist of weights of v. The uncopying map μ, on the other hand, loses some information when encoding a higher dimensional tensor into a lower dimensional space. For $w \in N \otimes N$, we have that $\mu(w)$ is a tensor consisting only of the diagonal elements of w, hence losing the information encoded in the non-diagonal part.

5 Frobenius parameters in distributional linguistic practice

It would be instructive to see how our decision for taking $S = N$ and the Frobenius constructions affect the meaning of a sentence in practice. We use a pictorial calculus that allows convenient graphical representations of the derivations. In this notation, each tensor is represented by a triangle, and its rank can be determined by the outgoing wires. The tensor product is depicted as juxtaposition of triangles. We also remind to the reader that the relational method for constructing a tensor for the meaning of a verb (Grefenstette and Sadrzadeh, 2011a) provides us with a matrix in N^2. In order to embed this in N^3, as required by the categorical framework, we apply a $\sigma : N^2 \to N^3$ map to it. Now the Frobenius operation σ gives us some options for the form of the resulting tensor, which are presented below:

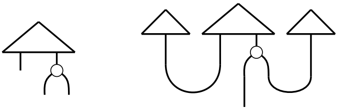
CpSbj The first option is to copy the "row" dimension of the matrix which, according to Equation 5, corresponds to the subject. In Part (a) below we see how σ transforms the verb this way. Once substituted in Equation 1, we obtain the interaction in Part (b). Linear algebraically, the σ map transforms the matrix of the verb in the way depicted on the right:



(a) (b)

$$\sigma :: \sum_{ij} c_{ij}(\overrightarrow{n_i} \otimes \overrightarrow{n_j}) \mapsto \sum_{iij} c_{ii}(\overrightarrow{n_i} \otimes \overrightarrow{n_i} \otimes \overrightarrow{n_j})$$

CpObj Our other option is to copy the "column" dimension of the matrix, i.e. the object dimension (the corresponding σ map again on the right):



(a) (b)

$$\sigma :: \sum_{ij} c_{ij}(\overrightarrow{n_i} \otimes \overrightarrow{n_j}) \mapsto \sum_{ijj} c_{jj}(\overrightarrow{n_i} \otimes \overrightarrow{n_j} \otimes \overrightarrow{n_j})$$

Geometrically, we can think of these two options as different ways for "diagonally" placing a plane into a cube. The diagrams provide us a direct way to simplify the calculations involved, since they suggest a closed form formula for each case. Taking as an example the diagram of the copy-subject method, we see that: (a) the object interacts with the verb; (b) the result of this interaction serves as input for the σ function; (c) one wire of the output of σ interacts with the object, while the other branch delivers the result. In terms of linear algebra, this corresponds to the computation $\sigma(\overline{verb} \times \overrightarrow{obj}) \times \overrightarrow{sbj}$ (where × denotes matrix multiplication), which is equivalent to the following:

$$\overrightarrow{sbj\ verb\ obj} = \overrightarrow{sbj} \odot (\overline{verb} \times \overrightarrow{obj}) \qquad (7)$$

where the symbol $\odot$ denotes component-wise multiplication and × is matrix multiplication. Similarly, the meaning of a transitive sentence for the copy-object case is given by:

$$\overrightarrow{sbj\ verb\ obj} = \overrightarrow{obj} \odot (\overline{verb}^{T} \times \overrightarrow{sbj}) \qquad (8)$$

We should bring to the reader's attention the fact that equipped with the above closed forms we do not need to create or manipulate rank-3 tensors at any point of the computation, something that would cause high computational overhead. Furthermore, note that the nesting problem of Grefenstette and Sadrzadeh (2011a) does not arise here, since the linguistic and concrete types are the same.

6 Experiments

We train our vectors from a lemmatised version of the British National Corpus (BNC), following closely the parameters of the setting described in Mitchell and Lapata (2008), later used by Grefenstette and Sadrzadeh (2011a). Specifically, we use the 2000 most frequent words as the basis for our vector space; this single space will serve as a semantic space for both nouns and sentences. The weights of the vectors are set to the ratio of the probability of the context word given the target word to the probability of the context word overall. As our similarity measure we use the cosine distance.

6.1 Disambiguation

We first test our models against the disambiguation task for transitive sentences described in Grefenstette and Sadrzadeh (2011a). The goal is to assess how well a model can discriminate between the different senses of an ambiguous verb, given the context (subject and object) of that verb. The entries of this dataset consist of a target verb, a subject, an object, and a landmark verb used for the comparison. One such entry for example is, “write, pupil, name, spell”. A good compositional model should be able to understand that the sentence “pupil write name” is closer to the sentence “pupil spell name” than, for example, to “pupil publish name”. On the other hand, given the context “writer, book” these results should be reversed. The dataset contains 200 such entries with verbs from CELEX, hence 400 sentences. The evaluation of this experiment is performed by calculating Spearman’s ρ correlation against the judgements of 25 human evaluators. As our baselines we use an additive (ADDTV) and a multiplicative (MULTP) model, where the meaning of a sentence is computed by adding and point-wise multiplying, respectively, the context vectors of its words.

The results are shown in Table 1. The most successful $S = N$ model for this task is the copy-object model, which is performing really close to the original relational model of Grefenstette and Sadrzadeh (2011a), with the difference to be statistically insignificant. This is a promising result, since it suggests that the lower-dimensional new model performs similarly with the richer structure of the old model for transitive sentences, while at the same time allows generalisation to even more complex sentences[1]. More importantly, note that the categorical models are the only ones that respect the word order and grammatical structure of sentences; a feature completely dismissed in the simple multiplicative model.

	Upper-bound	ADDTV	MULTP	CPSBJ	CPOBJ
ρ	0.620	0.050	0.163	0.143	**0.172**

Table 1: Disambiguation results. Upper-bound denotes the inter-annotator agreement.

6.2 Definition classification

The ability of reliably comparing the meaning of single words with larger textual fragments, e.g. phrases or even sentences, can be an invaluable tool for many challenging NLP tasks, such as definition classification, paraphrasing, sentiment analysis, or even the simple everyday search on the Internet. In this task we examine the extent to which our models can correctly match a number of terms (single words) with a number of definitions (phrases). To our knowledge, this is the first time a compositional distributional model is tested for its ability to match words with phrases. Our dataset consists of 112 terms (72 nouns and 40 verbs) and their main definitions, extracted from The Oxford Junior Dictionary (Sansome et al., 2000). For each term, and in order to get a richer dataset, we added two more definitions that expressed the same or an

[1] The original relational model of Grefenstette and Sadrzadeh (2011a) with $S = N^2$, provided a ρ of 0.21. When computed with our program with the exact same parameters (without embedding them in the $S = N$ model), we obtained a ρ of 0.195. The differences between both of these and our best model are statistically insignificant. In Grefenstette and Sadrzadeh (2011b), a direct non-relational model was used to compute verb matrices; this provided a ρ of 0.28. However, as explained by the authors themselves, this method is not general and for instance cannot be used for intransitive verbs.

Term	**Main definition**	**Def. 2**	**Def. 3**
blaze	large strong fire	huge potent flame	substantial heat
husband	married man	partner of a woman	male spouse
apologise	say sorry	express regret or sadness	acknowledge shortcoming or failing
embark	get on a ship	enter boat or vessel	commence trip

Table 2: Sample of the dataset for the term/definition comparison task.

alternative meaning, using the entries from the Oxford Concise School Dictionary (Hawkins et al., 2004) or by paraphrasing with the WordNet synonyms of the words in the definitions. So in total we obtained three definitions per term. In all cases a definition for a noun-term is a noun phrase, whereas the definitions for the verb-terms consist of verb phrases. For the latter case, we construct our verb vectors by summing over all context vectors of objects with which the verb appears in the corpus in a verb phrase; that is, we use $\overline{verb} = \sum_i \overrightarrow{obj_i}$. A sample of the dataset is shown in Table 2; the complete dataset will be made available online.

We approach the evaluation problem as a classification task, where the terms have the role of the classes. Specifically, we calculate the distance between each definition and every term in the dataset, and the definition is assigned to the term that gives the higher similarity. We evaluate the results by calculating accuracy (Table 3). Our model is referred to as the copy-object model (CpObj), and is compared with the multiplicative and additive models. The copy-object and multiplicative models perform similarly, with the former to have slightly better performance for nouns and the latter to be slightly better for verbs. We speculate that this lesser ability of our model in verbs terms is due to data sparsity, since the cases of pure verb phrases (from which we build the verb vectors for this task) are limited in BNC and not every verb of our dataset had a well-populated vector representation.

	CpObj	Multp	Addt	Cont
Nouns	**0.24**	0.22	0.17	0.09
Verbs	0.28	**0.30**	0.25	0.07

Table 3: Accuracy results for the term/definition comparison task.

7 Conclusion

The contribution of this work is that it provides a faithful implementation of the general categorical compositional distributional model of Coecke et al. (2010), with three important advantages compared to previous attempts: (1) it makes possible to compare phrases and sentences with different structures, up to the extreme case of comparing a sentence with a single word; (2) it follows the types suggested by the type-logical approaches, hence enables us to build concrete vectors for nested relational phrases; and (3) drastically reduces the space complexity of previous implementations. We achieved this using operations of Frobenius Algebras over vector spaces to expand and shrink the dimensions of the concrete tensors involved in the actual computations. This theoretical result stands on its own right, since it provides a framework that can be used in conjunction with various compositional-distributional settings and techniques. For example, one could populate the relational matrices using machine-learning techniques, as Baroni and Zamparelli (2010) tried for adjective-noun pairs, and then apply the categorical framework for the composition as described in this paper. As a proof of concept for the viability of our method, we presented experimental results in two tasks involving disambiguation and definition classification.

References

Baroni, M. and Zamparelli, R. (2010). Nouns are Vectors, Adjectives are Matrices. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Carboni, A. and Walters, R. (1987). Cartesian Bicategories I. *Journal of Pure and Applied Algebra*, 49.

Coecke, B. and Paquette, E. (2011). Categories for the Practicing Physicist. In Coecke, B., editor, *New Structures for Physics*, pages 167–271. Springer.

Coecke, B., Pavlovic, D., and Vicary, J. (2008). A New Description of Orthogonal Bases. *Mathematical Structures in Computer Science*, 1.

Coecke, B., Sadrzadeh, M., and Clark, S. (2010). Mathematical Foundations for Distributed Compositional Model of Meaning. Lambek Festschrift. *Linguistic Analysis*, 36:345–384.

Curran, J. (2004). *From Distributional to Semantic Similarity*. PhD thesis, School of Informatics, University of Edinburgh.

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

Firth, J. (1957). A synopsis of linguistic theory 1930–1955. In *Studies in Linguistic Analysis*.

Frobenius, F. (1903). Theorie der Hyperkomplexen Gröβen. *Sitzung der Phys.-Math*, pages 504–538.

Grefenstette, E. and Sadrzadeh, M. (2011a). Experimental Support for a Categorical Compositional Distributional Model of Meaning. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Grefenstette, E. and Sadrzadeh, M. (2011b). Experimenting with Transitive Verbs in a DisCoCat. In *Proceedings of Workshop on Geometrical Models of Natural Language Semantics (GEMS)*.

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

Hawkins, J., Delahunty, A., and McDonald, F. (2004). *Oxford Concise School Dictionary*. Oxford University Press.

Kock, J. (2003). Frobenius Algebras and 2D Topological Quantum Field Theories. In *London Mathematical Society Student Texts*. Cambridge University Press.

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

Landauer, T. and Dumais, S. (1997). A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquision, Induction, and Representation of Knowledge. *Psychological Review*.

Manning, C., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Mitchell, J. and Lapata, M. (2008). Vector-based Models of Semantic Composition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 236–244.

Sansome, R., Reid, D., and Spooner, A. (2000). *The Oxford Junior Dictionary*. Oxford University Press.

Schütze, H. (1998). Automatic Word Sense Discrimination. *Computational Linguistics*, 24:97–123.

Socher, R., Huang, E., Pennington, J., Ng, A., and Manning, C. (2011). Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. *Advances in Neural Information Processing Systems*, 24.

Socher, R., Manning, C., and Ng, A. (2010). Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*.