Open System Categorical Quantum Semantics in Natural Language Processing

Robin Piedeleu[1], Dimitri Kartsaklis[2], Bob Coecke[1], and Mehrnoosh Sadrzadeh[2]

1 **Department of Computer Science, University of Oxford**
Parks Road, Oxford OX1 3QD, UK
{robin.piedeleu;bob.coecke}@cs.ox.ac.uk

2 **School of Electronic Engineering and Computer Science, Queen Mary University of London**
Mile End Road, London E1 4NS, UK
{d.kartsaklis;m.sadrzadeh}@qmul.ac.uk

Abstract

Originally inspired by categorical quantum mechanics (Abramsky and Coecke, LiCS'04), the categorical compositional distributional model of natural language meaning of Coecke, Sadrzadeh and Clark provides a conceptually motivated procedure to compute the meaning of a sentence, given its grammatical structure within a Lambek pregroup and a vectorial representation of the meaning of its parts. Moreover, just like CQM allows for varying the model in which we interpret quantum axioms, one can also vary the model in which we interpret word meaning.

In this paper we show that further developments in categorical quantum mechanics are relevant to natural language processing too. Firstly, Selinger's CPM-construction allows for explicitly taking into account lexical ambiguity and distinguishing between the two inherently different notions of homonymy and polysemy. In terms of the model in which we interpret word meaning, this means a passage from the vector space model to density matrices. Despite this change of model, standard empirical methods for comparing meanings can be easily adopted, which we demonstrate by a small-scale experiment on real-world data. Secondly, commutative classical structures as well as their non-commutative counterparts that arise in the image of the CPM-construction allow for encoding relative pronouns, verbs and adjectives, and finally, iteration of the CPM-construction, something that has no counterpart in the quantum realm, enables one to accommodate both entailment and ambiguity.

1998 ACM Subject Classification I.2.7 Natural Language Processing

Keywords and phrases category theory, density matrices, distributional models, semantics

Digital Object Identifier 10.4230/LIPIcs.CALCO.2015.267

1 Introduction

Language serves to convey meaning. From this perspective, the ultimate and long-standing goal of any computational linguist is to capture and adequately represent the meaning of an utterance in a computer's memory. At word level, *distributional semantics* offers an effective way to achieve that goal; following the *distributional hypothesis* [11] which states that the meaning of a word is determined by its context, words are represented as vectors of co-occurrence statistics with all other words in the vocabulary. While models following this paradigm have been found very useful in a number of natural language processing tasks, they do not scale up to the level of phrases or sentences. This is due to the capacity of natural language to generate an infinite number of structures (phases and sentences) from finite means (words); no text corpus, regardless of its size, can provide reliable distributional

© Robin Piedeleu, Dimitri Kartsaklis, Bob Coecke and Mehrnoosh Sadrzadeh;
licensed under Creative Commons License CC-BY
6th International Conference on Algebra and Coalgebra in Computer Science (CALCO'15).
Editors: Larry Moss and Paweł Sobociński; pp. 267–286

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

statistics for a multi-word sentence. On the other hand, type-logical approaches conforming to the tradition of Lambek [16], Montague and other pioneers of language, are compositional and deal with the sentence at a more abstract level based on the syntactical rules that hold between the different text constituents, but in principle they do not provide a convincing model for word meaning.

The categorical compositional distributional model of Coecke, Sadrzadeh and Clark [7] addresses the challenge of combining these two orthogonal models of meaning in a unified setting. The model is based on the observation that a grammar expressed as a pregroup [15] shares the same structure with the category of finite dimensional vector spaces and linear maps, that of a *compact closed category* [14]. In principle, this offers a canonical way to express a grammatical derivation as a morphism that defines linear-algebraic manipulations between vector spaces, resulting in a sentence vector. The main characteristic of the model is that the grammatical type of a word determines the vector space in which it lives. Words with atomic types, such as nouns, are represented by vectors living in some basic vector space N; on the contrary, relational words such as verbs and adjectives live in tensor product spaces of higher order. An adjective, for example, is an element of $N \otimes N$, while a transitive verb lives in $N \otimes S \otimes N$. The relational tensors act on their argument by *tensor contraction*, a generalization of the familiar notion of matrix multiplication to higher order tensors.

Ambiguity is a dominant feature of language. At the lexical level, one can distinguish between two broad types of ambiguity: *homonymy* refers to cases in which, due to some historical accident, words that share exactly the same spelling and pronunciation are used to describe completely distinct concepts; such an example is 'bank', meaning a financial institution and a land alongside a river. On the other hand, the senses of a *polysemous* word are usually closely related with only small deviations between them; as an example, think of 'bank' again as a financial institution and the concrete building where that institution is accommodated. These two notions of ambiguity are inherently different; while a polysemous word still retains a certain level of semantic coherence, a homonymous word can be seen as an incoherent mixing due to coincidence. The issue of lexical ambiguity and the different levels of it is currently ignored from almost all attempts that aim to equip distributional models of meaning with compositionality.

The purpose of this paper is to provide the theoretical foundations for a compositional distributional model of meaning capable of explicitly dealing with lexical ambiguity. In the proposed model we exploit the observation that the compact closed structure on which the original model of Coecke et al. [7] was based provides an abstraction of the Hilbert space formulation used in the quantum theory, in terms of pure quantum states as vectors, which is known under the umbrella of categorical quantum mechanics [1]. In fact, the original model of Coecke et al. was itself greatly inspired by quantum theory, and in particular, by quantum protocols such as quantum teleportation. Importantly, vectors in a Hilbert space represent the states of a closed quantum system, also called *pure* states. Selinger's *CPM-construction* [21], which maps any dagger compact closed category on another one, then adjoins *open system* states, also called *mixed states*. In the new model, these allow for a lack of knowledge on part of the system under consideration, which may be about an extended part of the quantum system, or uncertainty (read: ambiguity) regarding the preparation procedure.

The crucial distinction between homonymous and polysemous words is achieved as follows: while a polysemous word corresponds to a *pure* quantum state, a homonymous word is given by a *mixed* state that essentially embodies a probability distribution over all potential meanings of that word. Mathematically, a mixed states is expressed as a *density matrix*: a self-adjoint, positive semi-definite operator with trace one. The new formulation offers many opportunities for interesting and novel research. For instance, by exploiting the notion of *Von Neumann entropy* one can measure how ambiguity evolves from individual words to

larger text constituents; we would expect that the level of ambiguity in word 'bank' is higher than that of the compound 'river bank'.

Furthermore, the richness of the new category in which the meanings of words now live offers interesting alternative design options. In the past, for example, Sadrzadeh, Kartsaklis and colleagues [19, 12] enriched the categorical compositional model with elements of classical processing, exploiting the fact that any basis of a finite-dimensional vector space induces a *commutative Frobenius algebra* over this space, which allows the uniform copying or deleting of the information relative to this basis [6]. As we will see in Sect. 4, the dagger compact closed categories arising from the CPM-construction also accommodate canonical non-commutative Frobenius algebras which have the potential to account for the non-commutativity of language.

Finally, we discuss how iterated application of the CPM-construction, which gives rise to states that have no interpretation in quantum theory, does have a natural application in natural language processing. It allows for simultaneous semantic representation of more than one language feature that can be represented by density matrices, for example, lexical entailment in conjunction with ambiguity.

Related work The issue of lexical ambiguity in categorical compositional models of meaning has been previously experimentally investigated by Kartsaklis and Sadrzadeh [13], who present evidence that the introduction of an explicit disambiguation step on the word vectors prior to composition improves the performance of the models. Furthermore, the research presented here is not the only one that uses density matrices for linguistic purposes. Balkır [2] uses a form of density matrices in order to provide a similarity measure that can be used for evaluating hyponymy-hypernymy relations. In Sect. 5 we indicate how these two uses of density matrices can be merged into one. Finally, Blacoe et al. [3] describe a distributional (but not compositional) model of meaning based on density matrices created by grammatical dependencies.

2 Background

The field of *category theory* aims at identifying and studying connections between seemingly different forms of mathematical structures. A very representative example of its potency is the compositional categorical framework of Coecke et al. [7], which shows that a grammatical derivation defining the structure of a sentence is homomorphic to a linear-algebraic formula acting on a semantic space defined by a distributional model. The framework offers a concrete manifestation of the *rule-to-rule hypothesis* and a mathematical counterpart to the formal semantics perspective on language. As noted above, the main idea is based on the fact that both the type-logic of the model, a pregroup grammar, and the semantic category, namely **FHilb**, possess a compact-closed structure. Recall that a *compact closed category* is a monoidal category in which every object A has a left and right adjoint, denoted as A^l, A^r respectively, for which the following special morphisms exist:

$$\eta^l : I \to A \otimes A^l \qquad \eta^r : I \to A^r \otimes A \qquad \epsilon^l : A^l \otimes A \to I \qquad \epsilon^r : A \otimes A^r \to I \tag{1}$$

These maps need to satisfy certain conditions (known as *yanking equations*) which ensure that all relevant diagrams commute:

$$\begin{aligned} &(1_A \otimes \epsilon^l_A) \circ (\eta^l_A \otimes 1_A) = 1_A \qquad (\epsilon^r_A \otimes 1_A) \circ (1_A \otimes \eta^r_A) = 1_A \\ &(\epsilon^l_A \otimes 1_{A^l}) \circ (1_{A^l} \otimes \eta^l_A) = 1_{A^l} \qquad (1_{A^r} \otimes \epsilon^r_A) \circ (\eta^r_A \otimes 1_{A^r}) = 1_{A^r} \end{aligned} \tag{2}$$

Finally, the passage from syntax to semantics is carried out by a *strong monoidal functor* and, as a result, preserves the compact closed structure. Before we proceed to expand on the above constructions, we refer the reader to App. A for a brief introduction to the graphical calculus of monoidal categories which will be used throughout our exposition.

2.1 Pregroup grammars

A *pregroup algebra* [15] is a partially ordered monoid with unit 1, whose each element p has a left adjoint p^l and a right adjoint p^r, conforming to the following inequalities:

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

A *pregroup grammar* is a pregroup algebra freely generated over a set of basic types $\mathcal{B}$ including a designated end type and a type dictionary that assigns elements of the pregroup to the vocabulary of a language. For example, it is usually assumed that $\mathcal{B} = \{n, s\}$, where n is the type assigned to a noun or a well-formed noun phrase, while s is a designated type kept for a well-formed sentence. Atomic types can be combined in order to provide types for relational words; for example, an adjective has type $n \cdot n^l$, reflecting the fact that it is something that expects for a noun at its right-hand side in order to return another noun. Similarly, a transitive verb has type $n^r \cdot s \cdot n^l$, denoting something that expects two nouns (one at each side) in order to return a sentence. Based on (3), for this latter case the pregroup derivation gets the following form:

$$n \cdot (n^r \cdot s \cdot n^l) \cdot n = (n \cdot n^r) \cdot s \cdot (n^l \cdot n) \leq 1 \cdot s \cdot 1 \leq s \tag{4}$$

Let $\mathbf{C_F}$ denote the *free compact closed category* derived from the pregroup algebra of a pregroup grammar [18]; then, according to (1), the above type reduction corresponds to the morphism $\epsilon_n^r \cdot 1_s \cdot \epsilon_n^l : n \cdot n^r \cdot s \cdot n^l \cdot n \to s$ in $\mathbf{C_F}$.

2.2 From syntax to semantics

The type-logical approach presented in Sect. 2.1 is compositional, but unable to distinguish between words of the same type; even more importantly, the only information that a derivation such as the one in (4) can provide to us is whether the sentence is well-formed or not. Distributional models of meaning offer a solution to the first of these problems, by representing a word in terms of its distributional behaviour in a large corpus of text. While the actual methods for achieving this can vary (see App. D for a concrete implementation), the goal is always the same: to represent words as points of some metric space, where differences in semantic similarity can be detected and precisely quantified. The prime intuition is that words appearing in similar contexts must have a similar meaning [11]. The word vectors typically live in a highly dimensional semantic space with a fixed orthonormal basis, the elements of which correspond to content-bearing words. The values in the vector of a target word w_t express co-occurrence statistics extracted from some large corpus of text, showing how strongly w_t is associated with each one of the basis words. For a concise introduction to distributional models of meaning see [23].

We take $(\mathbf{FHilb}, \otimes)$, the category of finite dimensional Hilbert spaces and linear maps over the scalar field I, to be the semantic counterpart of $\mathbf{C_F}$ which, as we saw before, accommodates the grammar. **FHilb** is a *dagger compact closed* category (or, †-compact closed); that is, a *symmetric* compact closed category (so that $A^r \cong A^l = A^*$ for all A) equipped with an involutive contravariant functor $\dagger : \mathbf{FHilb} \to \mathbf{FHilb}$ that is the identity on objects. Concretely, in **FHilb**, for a morphism $f : A \to B$, its dagger $f^\dagger : B \to A$ is simply its adjoint. Furthermore, $\epsilon_A = \eta_A^\dagger \circ \sigma_{A^*,A}$ for all A.

Taking $|\psi\rangle$ and $|\phi\rangle$ to be two vectors in a Hilbert space $\mathcal{H}$, $\epsilon_A : A^* \otimes A \to I$ is the pairing $\epsilon_A(\langle\psi|, |\phi\rangle) = \langle\psi|(|\phi\rangle) = \langle\psi|\phi\rangle$ and $\eta_A = \epsilon_A^\dagger$. This allows the inner product to be categorically defined as $\langle\psi|\phi\rangle : I \xrightarrow{\psi} \mathcal{H} \xrightarrow{\phi^\dagger} I$. In practice it is often necessary to normalise in order to obtain the cosine of the angle between vectors as a measure of semantic similarity.

2.3 Quantizing the grammar

We now proceed to present a solution to the second problem posed above, that of providing a quantified semantic representation for a sentence by composing the representations of the words therein: in this paper we follow [17] and [12] and we achieve the transition from syntax to semantics via a *strong monoidal functor* $Q : \mathbf{C_F} \to \mathbf{FHilb}$ which can be shown to also preserve the compact structure so that $Q(p^l) = Q(p)^l$ and $Q(p^r) = Q(p)^r$ for p an object of $\mathbf{C_F}$. Since each object in **FHilb** is its own dual we also have $Q(p^l) \cong Q(p) \cong Q(p^r)$. Moreover, for basic types, we let $Q(n) = N$ and $Q(s) = S$. Note that since Q is strongly monoidal, complex types are mapped to tensor product of vector spaces:

$$Q(n \cdot n^r) = Q(n) \otimes Q(n^r) = N \otimes N \qquad Q(n^r \cdot s \cdot n^l) = Q(n^r) \otimes Q(s) \otimes Q(n^l) = N \otimes S \otimes N$$

Finally, each morphism in $\mathbf{C_F}$ is mapped to a linear map in **FHilb**. Equipped with such a functor, we can now define the meaning of a sentence as follows:

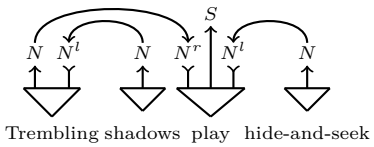
▶ **Definition 1.** Let $|w_i\rangle$ be a vector $I \to Q(p_i)$ corresponding to word w_i with type p_i in a sentence $w_1 w_2 \dots w_n$. Given a type-reduction $\alpha : p_1 \cdot p_2 \cdot \dots \cdot p_n \to s$, the meaning of the sentence is defined as:

$$|w_1 w_2 \dots w_n\rangle := Q(\alpha)(|w_1\rangle \otimes \dots \otimes |w_n\rangle) \tag{5}$$

Take as an example the sentence "Trembling shadows play hide-and-seek", with the standard types $n \cdot n^l$ and $n^r \cdot s \cdot n^l$ assigned to adjectives and verbs, respectively. Then the adjective 'trembling' will be a morphism $I \to Q(n \cdot n^l) = I \to N \otimes N$, that is, a state in the tensor product space $N \otimes N$. Note that this matrix defines a linear map $N \to N$, an interpretation that is fully aligned with the formal semantics perspective: an adjective is a function that takes a noun as input and returns a modified version of it. Similarly, the verb 'play' lives in $N \otimes S \otimes N$ or, equivalently, is a bi-linear map $N \otimes N \to S$ (with a subject and an object as arguments) which returns a sentence. In contrast to those two relational words, the nouns 'shadows' and 'hide-and-seek' are plain vectors in N. The syntax of the sentence conforms to the following type reduction:

$$(\epsilon_n^r \cdot 1_s) \circ (1_n \cdot \epsilon_n^l \cdot 1_{n^r} \cdot 1_s \cdot \epsilon_n^l) : n \cdot n^l \cdot n \cdot n^r \cdot s \cdot n^l \cdot n \to s \tag{6}$$

which, when transferred to **FHilb** via Q, yields the following diagrammatic derivation:



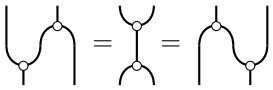
(7)

2.4 Using Frobenius algebras in language

Compact closed categories on their own do not have much structure. The expressive power of these categories can be increased using *Frobenius algebras*. Recall from [4] that a Frobenius algebra in a monoidal category is a quintuple $(A, \Delta, \iota, \mu, \zeta)$ such that:

- (A, μ, ζ) is a monoid, that is we have $\mu : A \otimes A \to A$ and $\zeta : I \to A$ satisfying associativity and unit conditions,
- (A, Δ, ι) is a co-monoid, so that $\Delta : A \to A \otimes A$ and $\iota : A \to I$ satisfy co-associativity and co-unit conditions;

- furthermore, Δ and μ adhere to the following *Frobenius condition*:

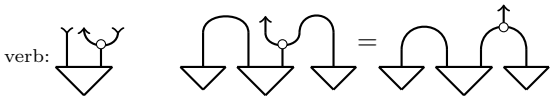


(8)

In a monoidal †-category, a †-Frobenius algebra is a Frobenius algebra whose co-monoid is adjoint to the monoid. As shown in [6], every finite dimensional Hilbert space $\mathcal{H}$ with orthonormal basis $\{|i\rangle\}$ has a †-Frobenius algebra associated to it, the co-multiplication and multiplication of which correspond to uniformly *copying* and *uncopying* the basis as follows:

$$\Delta :: |i\rangle \mapsto |i\rangle \otimes |i\rangle \quad \iota :: |i\rangle \mapsto 1 \quad \mu :: |i\rangle \otimes |j\rangle \mapsto \delta_{ij}|i\rangle := \begin{cases} |i\rangle & i = j \\ |0\rangle & i \neq j \end{cases} \qquad \zeta :: 1 \mapsto \sum_i |i\rangle$$

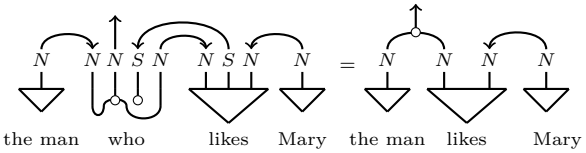
Abstractly, this enables us to copy and delete the (classical) information relative to the given basis. Concretely, the copying Δ-map amounts to encoding faithfully the components of a vector in $\mathcal{H}$ as the diagonal elements of a matrix in $\mathcal{H} \otimes \mathcal{H}$, while the "uncopying" operation μ picks out the diagonal elements of a matrix and returns them as a vector in $\mathcal{H}$. Kartsaklis et al. [12] use the Frobenius co-multiplication in order to faithfully encode tensors of lower order to higher order ones, thus restoring the proper functorial relation. An adjective, for example, is given as $\Delta(\sum_i |noun_i\rangle)$, where $|noun_i\rangle$ is a noun modified by the specific adjective in a training corpus. Furthermore, given a transitive verb constructed as $|verb\rangle = \sum_i |subj_i\rangle \otimes |obj_i\rangle$ [10], we can encode it to a tensor in $\mathcal{H} \otimes \mathcal{H} \otimes \mathcal{H}$ by either copying the row dimension (responsible for the interaction of the verb with the subject noun) or the column dimension (responsible for the interaction with the object). For the latter case, referred to by Copy-Object, the composition becomes as follows:



(9)

The composition for the case of copying the subject dimension proceeds similarly on the left-hand side. In practice, empirical work has shown that objects have stronger influence on the meaning of a transitive sentence than subjects [12], which suggests that the Frobenius structure of the Copy-Object approach is a more effective model of sentential compositionality.

Finally, Sadrzadeh et al. [19] exploit the abilities of Frobenius algebras in order to model relative pronouns. Specifically, *copying* is used in conjunction with *deleting* in order to allow the head noun of a relative clause to interact with its modifier verb phrase from the far left-hand side of the clause to its right-hand side. For the case of a relative clause modifying a subject this is achieved as follows:



(10)

3 Encoding ambiguity

The previous compositional model relies on a strong monoidal functor from a compact closed category, representing syntax, to **FHilb**, modelling a form of distributional semantics. In this section we will modify the functor to a new codomain category. To achieve our goal, we will explore a categorical construction, inspired from quantum physics and originally due to Selinger [21], in the context of the categorical model of meaning developed in the previous sections.

3.1 Mixing in FHilb

Although seemingly unrelated, quantum mechanics and linguistics share a common link through the framework of †-compact closed categories, an abstraction of the Hilbert space formulation, and have been used in the past [1] to provide structural proofs for a class of quantum protocols, essentially recasting the vector space semantics of quantum mechanics in a more abstract way. Shifting the perspective to the field of linguistics, we saw how the same formalism proposes a description of the semantic interactions of words at the sentence level. Here we make the connection between the two fields even more explicit, taking advantage of the fact that the ultimate purpose of quantum mechanics is to deal with uncertainty—and this is essentially what we need to achieve here in the context of language.

We start by observing that, in quantum physics, the Hilbert space model is insufficient to incorporate the epistemic state of the observer in its formalism: what if one does not have knowledge of a quantum system's initial state and can only attribute a probability distribution to a set of possible states? The answer is by considering a *statistical ensemble* of pure states: for example, one may assign a $\frac{1}{2}$ probability that the state vector of a system is $|\psi_1\rangle$ and a $\frac{1}{2}$ probability that it is in state $|\psi_2\rangle$. We say that this system is in a *mixed state*. In the Hilbert space setting, such a state cannot be represented as a vector. In fact, any normalised sum of pure states is again a pure state (by the vector space structure). Note that the state $(\psi_1 + \psi_2)/\sqrt{2}$ is a *quantum superposition* and not the mathematical representation of the mixed state above.

This situation is similar to the issue we face when trying to model ambiguity in distributional semantics: given two different meanings of a homonymous word and their relative weights (given as probabilities), simply looking at the convex composition of the associated vectors collapses the ambiguous meaning to a single vector, thereby fusing together the two senses of the word. The mathematical response to this problem is to move the focus away from states in a Hilbert space to a specific kind of operators on the same space: more specifically, to *density operators*, i.e., positive semi-definite, self-adjoint operators of trace one. The density operator formalism is our means to express a probability distribution over the potential meanings of a homonymous word in a distributional model (see App. C for a more detailed linguistic intuition). We formally define this as follows:

▶ **Definition 2.** Let a distributional model be given in the form of a Hilbert space M, in which every word w_t is represented by a statistical ensemble $\{(p_i, |w_t^i\rangle)\}_i$—where $|w_t^i\rangle$ is a vector corresponding to a specific unambiguous meaning of the word that can occur with probability p_i. The distributional meaning of the word is defined as:

$$\rho(w_t) = \sum_i p_i |w_t^i\rangle\langle w_t^i| \tag{11}$$

Note that for the case of a non-homonymous word, the above formula reduces to $|w_t\rangle\langle w_t|$, with $|w_t\rangle$ corresponding to the state vector assigned to w_t. Now, if mixed states are density operators, we need a notion of morphism that preserves this structure, i.e., that maps states to states. In the Hilbert space model, the morphisms were simply linear maps. The corresponding notion in the mixed setting is that of *completely positive maps*, that is, positive maps that respect the monoidal structure of the underlying category.

To constitute a compositional model of meaning, our construction also needs to respect our stated goals: specifically, the category of operator spaces and completely positive maps must be a †-compact closed category; furthermore, we need to identify the morphism that plays the part of the Frobenius algebra of the previous model. We start working towards these goals by describing a construction that builds a similar category, not only from **FHilb**, but, more abstractly, from any †-compact closed category.

3.2 Doubling and complete positivity

The category that we are going to build was originally introduced by Selinger [21] as a generalisation of the corresponding construction on Hilbert spaces. Conceptually, it corresponds to shifting the focus away from vectors or morphisms of the form $I \to A$ to operators on the same space or morphisms of type $A \to A$. We will formalise this idea by first introducing the category $\mathbf{D}(\mathcal{C})$ on a compact closed category $\mathcal{C}$, which can be perhaps better understood in its diagrammatic form as a *doubling* of the wires. In this context, we obtain a duality between states of $\mathbf{D}(\mathcal{C})$ and operators of $\mathcal{C}$, pictured by simple wire manipulations. As we will see, $\mathbf{D}(\mathcal{C})$ retains the compact closedness of $\mathcal{C}$ and is therefore a viable candidate for a semantic category in our compositional model of meaning. However, at this stage, states of $\mathbf{D}(\mathcal{C})$ do not yet admit a clear interpretation in terms of mixing. This is why we need to introduce the notion of completely positive morphisms, of which positive operators on a Hilbert space (mixed states in quantum mechanics) are a special case. This will allow us later to define the subcategory $\mathbf{CPM}(\mathcal{C})$ of $\mathbf{D}(\mathcal{C})$.

3.2.1 The D construction (doubling)

First, given a †-compact closed category[1] $\mathcal{C}$ we define:

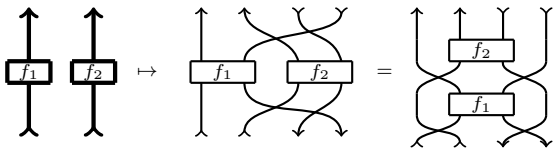
▶ **Definition 3.** The category $\mathbf{D}(\mathcal{C})$ with

- the same objects as $\mathcal{C}$;
- morphisms between objects A and B of $\mathbf{D}(\mathcal{C})$ are morphisms $A \otimes A^* \to B \otimes B^*$ of $\mathcal{C}$.
- composition and dagger are inherited from $\mathcal{C}$ via the embedding $E : \mathbf{D}(\mathcal{C}) \hookrightarrow \mathcal{C}$ defined by $A \mapsto A \otimes A^*$ on objects and $f \mapsto f$ on morphisms.

In addition, we can endow the category $\mathbf{D}(\mathcal{C})$ of a monoidal structure by defining the tensor $\otimes_{\mathbf{D}}$ as $A \otimes_{\mathbf{D}} B = A \otimes B$ on objects A and B, and for morphisms $f_1 : A \otimes A^* \to B \otimes B^*$ and $f_2 : C \otimes C^* \to D \otimes D^*$, by:

$$f_1 \otimes_{\mathbf{D}} f_2 : A \otimes C \otimes C^* \otimes A^* \xrightarrow{\cong} A \otimes A^* \otimes C \otimes C^* \xrightarrow{f_1 \otimes f_2} B \otimes B^* \otimes D \otimes D^* \xrightarrow{\cong} B \otimes D \otimes D^* \otimes B^* \quad (12)$$

Or graphically by,



(13)

where the arrow $\mapsto$ represents the functor E and we use the convention of depicting morphisms in $\mathbf{D}(\mathcal{C})$ with thick wires and boxes to avoid confusion. Note that the intuitive alternative of simply juxtaposing the two morphisms as we would in $\mathcal{C}$ fails to produce a completely positive morphism in general, as will become clearer when we define completely positivity in this context. This category carries all the required structure. We refer the reader to [21] for a proof of the following:

[1] The construction works on any monoidal category with a dagger, i.e., an involution, but we will not need the additional generality.

▶ Proposition 3.1. The category $\mathbf{D}(\mathcal{C})$ inherits a †-compact closed structure from $\mathcal{C}$ via the strict monoidal functor $M : \mathcal{C} \to \mathbf{D}(\mathcal{C})$ defined inductively by

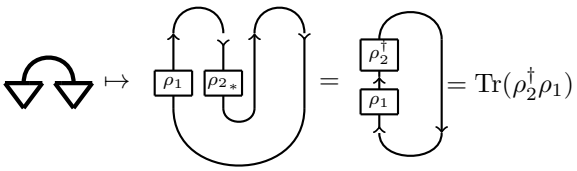
$$\begin{cases} f_1 \otimes f_2 & \mapsto M(f_1) \otimes_{\mathbf{D}} M(f_2) \quad ; \\ A & \mapsto A \qquad\qquad\qquad\quad \text{on objects;} \\ f & \mapsto f \otimes f_* \qquad\qquad \text{on morphisms.} \end{cases}$$

where $f_* = (f^\dagger)^*$ by definition.

The functor M shows that we are not losing any expressive power since unambiguous words (represented as maps of $\mathcal{C}$) still admit a faithful representation in doubled form. For reference, the reader can find in App. B a dictionary that translates useful diagrams from one category to the other. Now, notice that we have a bijective correspondence between states of $\mathbf{D}(\mathcal{C})$, i.e., morphisms $I \to A$ and operators on A in $\mathcal{C}$. Explicitly, the map $\mathcal{C}(A, A) \to \mathcal{C}(I, A \otimes A^*)$ is, for an operator $\rho : A \to A$,

$$\rho \mapsto \ulcorner \rho \urcorner = (\rho \otimes 1_{A^*}) \circ \eta_{A^*} = \tag{14}$$

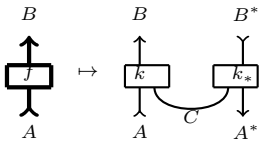
that is easily seen to be an isomorphism by bending back the rightmost wire (by application of the yanking equations (2)). In the special case of states, the generalised inner product generated by the dagger functor can be computed in terms of the canonical trace induced by the compact closed structure (and reduces to the usual inner product on a space of operators in **FHilb**):



$$= \mathrm{Tr}(\rho_2^\dagger \rho_1) \tag{15}$$

3.2.2 The CPM construction (complete positivity)

▶ **Definition 4.** A morphism $f : A \to B$ of $\mathbf{D}(\mathcal{C})$ is completely positive if there exists an object C and a morphism $k : C \otimes A \to B$, in $\mathcal{C}$, such that f embeds in $\mathcal{C}$ as $(k \otimes k_*) \circ (1_A \otimes \eta_{C^*} \otimes 1_{A^*})$ or, pictorially,

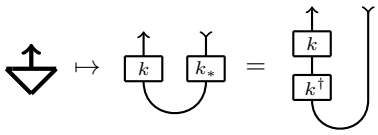


(16)

From this last representation, we easily see that the composition of two completely positive maps is completely positive. Similarly, the tensor product of two completely positive maps is completely positive. Therefore, we can define:

▶ **Definition 5.** The category $\mathbf{CPM}(\mathcal{C})$ is the subcategory of $\mathbf{D}(\mathcal{C})$ whose objects are the same and morphisms are completely positive maps.

$\mathbf{CPM}(\mathcal{C})$ is monoidal and $\otimes_{\mathbf{CPM}} = \otimes_{\mathbf{D}}$. We easily recover the usual notion of positive operator from this definition:



$$= \ulcorner k \circ k^{\dagger} \urcorner \tag{17}$$

with pure states corresponding to the disconnected case. Finally, from Def. 4 it is clear that, for a morphism f of $\mathcal{C}$, $M(f) = f \otimes f_*$ is completely positive. Thus,

▶ Proposition 3.2. *M factors through the embedding $I : \mathbf{CPM}(\mathcal{C}) \hookrightarrow \mathbf{D}(\mathcal{C})$, i.e., there exists a strictly monoidal functor $\tilde{M} : \mathcal{C} \to \mathbf{CPM}(\mathcal{C})$ such that $M = I\tilde{M}$.*

3.3 Categorical model of meaning: Reprise

We are now ready to put together all the concepts introduced above in the context of a compositional model of meaning. Our aim in this section is to reinterpret the previous model of [7] as a functor from a compact closed grammar to the category $\mathbf{CPM}(\mathcal{C})$, *for any compact closed category* $\mathcal{C}$. Given semantics in the form of a strong monoidal functor $Q : \mathbf{C_F} \to \mathcal{C}$, our model of meaning is defined by the composition:

$$\tilde{M}Q : \mathbf{C_F} \to \mathcal{C} \to \mathbf{CPM}(\mathcal{C}) \tag{18}$$

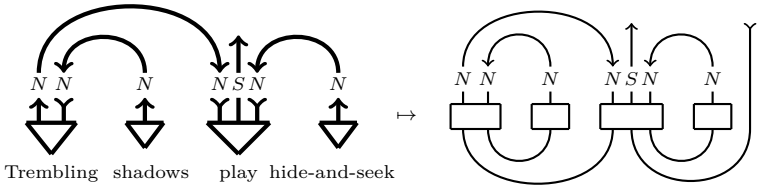
Since $\tilde{M}$ sends an object A to the same A in $\mathbf{CPM}(\mathcal{C})$, the mapping of atomic types, their duals and relational types of the grammar occurs in exactly the same fashion as in the previous model. Furthermore, note that Q is strongly monoidal and $\tilde{M}$ is strictly monoidal, so the resulting functor is strongly monoidal and, in particular, preserves the compact structure. Thus, we can perform type reductions in $\mathbf{CPM}(\mathcal{C})$ according to the grammatical structure dictated by the category $\mathbf{C_F}$.

Note that we have deliberately abstracted the model to highlight its richness—the category $\mathcal{C}$ could be any compact closed category: **FHilb**, the category **Rel** of sets and relations (in which case we recover a form of Montague semantics) or, as we will see in Sect. 5, even another iteration of the **CPM** construction.

▶ **Definition 6.** Let $\rho(w_i)$ be a meaning state $I \to \tilde{M}Q(p_i)$ corresponding to word w_i with type p_i in a sentence $w_1 \dots w_n$. Given a type-reduction $\alpha : p_1 \cdot \dots \cdot p_n \to s$, the meaning of the sentence is defined as:

$$\rho(w_1 \dots w_n) := \tilde{M}Q(\alpha)\big(\rho(w_1) \otimes_{\mathbf{CPM}} \dots \otimes_{\mathbf{CPM}} \rho(w_n)\big)$$

For example, assigning density matrix representations to the words in the previous example sentence "trembling shadows play hide and seek", we obtain the following meaning representation:



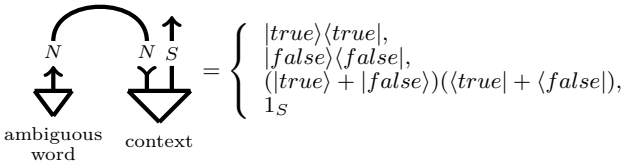
Diagrammatically, it is clear that in the new setting the partial trace implements meaning composition. Note that diagrams as the above illustrate the flow of ambiguity or information between words. The question of how does ambiguity evolve when composing words to

form sentences is very hard to answer precisely in full generality. The key message is that (unambiguous) meaning emerges in the interaction of a word with its context, through the wires. This process of disambiguation is perhaps better understood by studying very simple examples, as we are going to do in the next section.

3.4 Introducing ambiguity in formal semantics

Here, we will work in the category **CPM**(**Rel**). We recall that **Rel** is the †-compact category of sets and relations. The tensor product is the Cartesian product and the dagger associates to a relation its opposite. Let our sentence set be $S = \{true, false\}$. In **Rel**, this means that we are only interested in the truth of a sentence, as in Montague semantics. In this context, nouns are subsets of attributes. Given a context to which we pass the meaning of a word, the meaning of the resulting sentence can be either $|false\rangle$, $|true\rangle$ or $|false\rangle + |true\rangle$, the latter representing superposition, i.e., the case for which the context is insufficient to determine the truth of all the attributes of the word (classically, this can be identified with $false$).

On the other hand, in the internal logic of **CPM**(**Rel**), mixing adds a second dimension that can be interpreted as ambiguous meaning, regardless of truth. The possible values are:



where the identity on S represents ambiguity. Note that we use Dirac notation in **Rel** rather than set theoretic union and cartesian product, since elements in finite sets can be seen as basis vectors of free modules over the semi-ring of Booleans; a binary relation can be expressed as an adjacency matrix. The trace of a square matrix picks out the elements for which the corresponding relation is reflexive.

Consider the phrase 'queen rules'. We allow a few highly simplifying assumptions: first, we restrict our set of nouns to the rather peculiar 'Freddy Mercury', 'Brian May', 'Elisabeth II', 'chess', 'England' and the empty word ϵ. Moreover, we consider the verb 'rule', supposed to have the following unambiguous meaning:

$$|rule\rangle = |band\rangle \otimes |true\rangle \otimes |\epsilon\rangle + |chess\rangle \otimes |false\rangle \otimes |\epsilon\rangle + |elisabeth\rangle \otimes |true\rangle \otimes |england\rangle$$

with the obvious $|band\rangle = |freddy\rangle + |brian\rangle$. This definition reflects the fact that a band can rule (understand "be the best") as well as a monarch. Finally, the ambiguous meaning of 'queen' is represented by the following operator:

$$\rho(queen) = |elisabeth\rangle\langle elisabeth| + |band\rangle\langle band| + |chess\rangle\langle chess|$$

A computation of the meaning of the sentence in algebraic form yields, $\mathrm{Tr}_N(|rule\rangle\langle rule| \circ (\mathrm{Tr}_{N'}(\rho(queen)) \otimes 1'_N)) = 1_S$. In other words, the meaning of the sentence is neither $true$ nor $false$ but still ambiguous. This is because the context that we pass to 'queen' is insufficient to disambiguate it (the band *or* the monarch can rule). Now, if we consider 'queen rules England', the only matching pattern in the definition of $|rule\rangle$ is $|elisabeth\rangle$ which corresponds to a *unique* and therefore unambiguous meaning of $\rho(queen)$. Hence, a similar calculation yields $\mathrm{Tr}_N(|rule\rangle\langle rule| \circ (\mathrm{Tr}_{N'}(\rho(queen)) \otimes |england\rangle\langle england|)) = |true\rangle\langle true|$ and the sentence is not only true but unambiguous. In this case, the context was sufficient to disambiguate the meaning of the word 'queen'.

3.5 Measuring ambiguity with real data

While a large-scale experiment is out of the scope of this paper, in this section we present some preliminary witnessing results that showcase the potential of the model. Using 2000-dimensional meaning vectors created by the procedure described in App. D, we show how ambiguity evolves for five ambiguous nouns when they are modified by an adjective or a relative clause. For example, 'nail' can appear as 'rusty nail' or 'nail that grows'; in both cases the modifier resolves part of the ambiguity, so we expect that the entropy of the larger compound would be lower than that of the original ambiguous noun. Both types of composition use the Frobenius framework described in Sect. 2.4; We further remind that for a density matrix ρ with eigen-decomposition $\rho = \sum e_i |e_i\rangle\langle e_i|$, Von Neumann entropy is given as $S(\rho) = -\operatorname{Tr}(\rho \ln \rho) = -\sum_i e_i \ln e_i$.

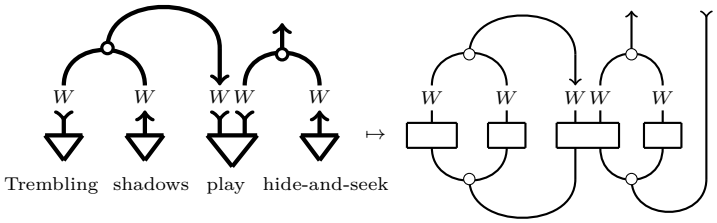
As Table 1 shows, the entropy of the compounds is always lower than that of the ambiguous noun. Even more interestingly, for some cases (e.g 'vessel that sails') the context is so strong that is capable to almost *purify* the meaning of the noun. This demonstrates an important aspect of the proposed model: *disambiguation = purification.*

	Relative Clauses			Adjectives		
noun: v_1/v_2	*noun*	*n* that v_1	*n* that v_2	adj_1/adj_2	adj_1 *n*	adj_2 *n*
organ: enchant/ache	0.18	0.11	0.08	music/body	0.10	0.13
vessel: swell/sail	0.25	0.16	**0.01**	blood/naval	0.05	0.07
queen: fly/rule	0.28	0.14	0.16	fair/chess	0.05	0.16
nail: gleam/grow	0.19	0.06	0.14	rusty/finger	**0.04**	0.11
bank: overflow/loan	0.21	0.19	0.18	water/financial	0.20	0.16

Table 1 Computing entropy for nouns modified by relative clauses and adjectives.

3.6 Flow of information with †-Frobenius algebras

In the above examples we used the assumption that a verb tensor had been faithfully constructed according to its grammatical type. However, as we saw in Sect. 2.4, concrete constructions might yield operators on a space of tensor order lower than the space to which the functor $\tilde{MQ}$ maps their grammatical type. As before, †-Frobenius algebras can be used to solve this type mismatch and encode the information carried by an operator into tensors of higher order. Specifically, we will first consider the †-Frobenius algebra whose copying map is $M(\Delta)$ and whose deleting map is $M(\iota)$, as doubling preserves both operations. In addition, the monoid operation is clearly completely positive. In more concrete terms, the monoid operation is precisely the point-wise (sometimes called Hadamard) product of matrices. Assuming we have a distributional model in the form of a vector space W with a distinguished basis and density matrices on W (to represent the meaning of our nouns and adjectives) and on $W \otimes W$ (for verbs), our example sentence is given by:



4 Non-commutativity

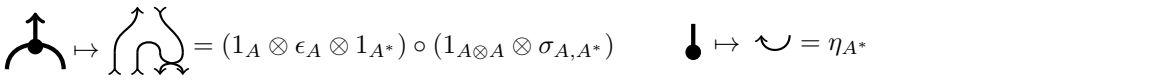
If the last section was concerned with applications of the CPM-construction to model ambiguity, here we discuss the role of the D-construction for the same purpose. Frobenius algebras on objects of $\mathbf{D}(\mathcal{C})$ are not necessarily commutative and thus their associated monoid

is not a completely positive morphism. In the quantum physical literature, non-completely positive maps are not usually considered since they are not physically realisable. However, in linguistics, free from these constraints, we could theoretically venture outside of the subcategory $\mathbf{CPM}(\mathcal{C})$, deep into $\mathbf{D}(\mathcal{C})$.

For example, Coecke, Heunen and Kissinger [8] introduced the category $\mathbf{CP}^*(\mathcal{C})$ of †-Frobenius algebras (with additional technical conditions) and completely positive maps, over an arbitrary †-compact category $\mathcal{C}$, in order to study the interaction of classical and quantum systems in a single categorical setting: classical systems are precisely the commutative algebras and completely positive maps are quantum channels, that is, physically realisable processes between systems. Interestingly, in accordance with the content of the no-broadcasting theorem for quantum systems the multiplication of a commutative algebra is a completely positive morphism while the multiplication of a non-commutative algebra is not. It is clear that the meaning composition of words in a sentence is only commutative in exceptional cases; the non commutativity of the grammatical structure reflects this. However, in earlier methods of composition, this complexity was lost in translation when passing to semantics.

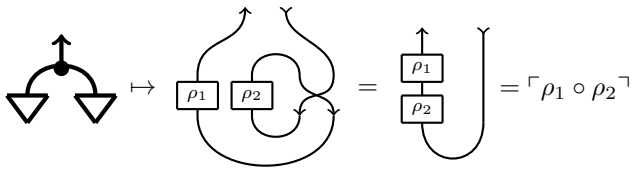
With linguistic applications in mind, the $\mathbf{CP}^*$ construction suggests various ways of composing the meaning of words, each corresponding to a specific Frobenius algebra operation. Conceptually, this idea makes sense since a verb does not compose with its subject in the same way that an adjective composes with the noun phrase to which it applies. The various ways of composing words may also offer a theoretical base for the introduction of logic in distributional models of natural language. This is where the richness of $\mathbf{D}(\mathcal{C})$ reveals itself: algebras in this category are more complex and, in particular, allow us to study the action of non-commutative structures—a topic of great interest to formal linguistics where the interaction of words is highly non-commutative. Hereafter we introduce a non-commutative †-Frobenius algebra that is not the doubled image of any algebra in $\mathcal{C}$.

▶ **Definition 7.** For every object A of $D(\mathcal{C})$, the morphisms of $D(\mathcal{C})$, $\mu : A \otimes_{\mathbf{D}} A \to A$ and $\iota : I \to A$ defined by the following diagrams in $\mathcal{C}$:

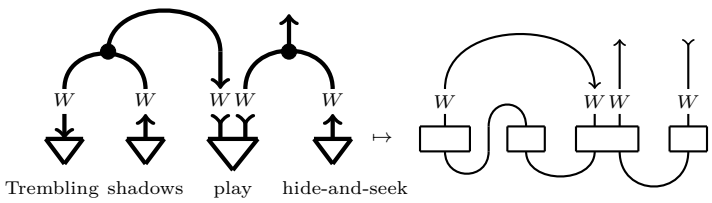


are the multiplication and unit of a †-Frobenius algebra $\mathcal{F}_{\mathbf{D}}$—where σ is the natural swap isomorphism in $\mathcal{C}$.

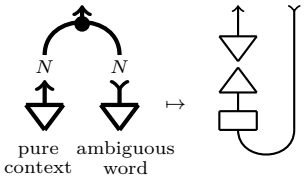
Proof that the above construction is indeed a †-Frobenius algebra can be found in [9]. The action of the Frobenius multiplication μ on states $I \to A$ of $\mathbf{D}(\mathcal{C})$ is particularly interesting; in fact, it implements the composition of operators of $\mathcal{C}$, in $\mathbf{D}(\mathcal{C})$:



The meaning of the "trembling shadows..." sentence using the algebra $\mathcal{F}_{\mathbf{D}}$ becomes:

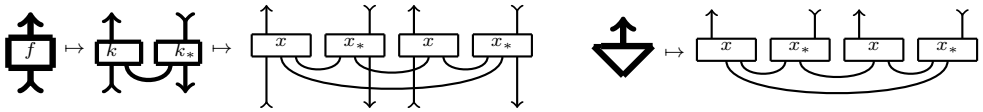


How does composition with the new algebra affect the flow of ambiguity in the simple case of an ambiguous word to which we pass an unambiguous context? Given a projection onto a one-dimensional subspace $|w\rangle\langle w|$ and a density operator ρ, the composition $|w\rangle\langle w|\rho$ is a (*not necessarily orthogonal*) projection. In a sense, the meaning of the pure word determines that of the ambiguous word as evidenced by the disconnected topology of the following diagram:

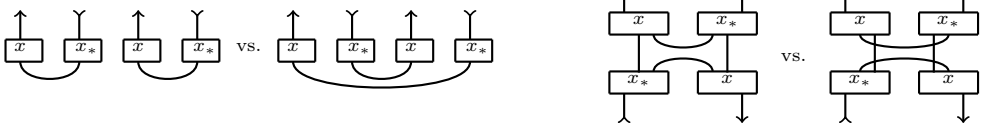


5 Adding lexical entailment

We now demonstrate the advantage of the fact that the CPM-construction is an abstract construction, and hence can be applied to any suitable (i.e. living in a †-compact closed category) model of word meaning. Besides ambiguity, another feature of language which is not captured by the distributional model is the fact that the meaning of one word (= *hypernym*) generalises that of another word (= *hyponym*). This points at a partial ordering of word meanings. For example, 'painter' generalises 'Brueghel'. Density matrices can be endowed with a partial ordering which could play that role, e.g. the *Bayesian ordering* [5]. This raises the question of how to accommodate both features together in a model of natural language meaning. Since $\mathbf{CPM}(\mathcal{C})$ is always †-compact closed, a canonical solution is obtained by iterating the CPM-construction:



Given a word/phrase/sentence meaning as above, lack of any ambiguity or generality correspond to distinct diagrams, respectively (left), and can be measured by taking the von Neumann entropy of different operators (right):



6 Conclusion and future work

In this paper we detailed a compositional distributional model of meaning capable of explicitly handling lexical ambiguity. We discussed its theoretical properties and demonstrated its potential for real-world natural language processing tasks by a small-scale experiment. A large-scale evaluation will be our challenging next step, aiming to provide empirical evidence regarding the effectiveness of the model in general and the performance of the different Frobenius algebras in particular. On the theoretical side, the logic of ambiguity in $\mathbf{CPM}(\mathbf{Rel})$, the non-commutative features of the D-construction as well as further exploration of nested levels of CPM, each deserve a separate treatment. In addition, one important weakness of distributional models is the representation of words that serve a purely logical role, like logical connectives or negation. Density operators support a form of logic whose distributional and compositional properties could be examined, potentially providing a solution to this long-standing problem of compositional distributional models.

References

1 Samson Abramsky and Bob Coecke. A categorical semantics of quantum protocols. In *19th Annual IEEE Symposium on Logic in Computer Science*, pages 415–425, 2004.

2 Esma Balkır. Using density matrices in a compositional distributional model of meaning. Master's thesis, University of Oxford, 2014.

3 William Blacoe, Elham Kashefi, and Mirella Lapata. A quantum-theoretic approach to distributional semantics. In *Proceedings of NACL 2013*, pages 847–857. Association for Computational Linguistics, June 2013.

4 A. Carboni and R.F.C. Walters. Cartesian Bicategories I. *Journal of Pure and Applied Algebra*, 49, 1987.

5 B. Coecke and K. Martin. A partial order on classical and quantum states. In B. Coecke, editor, *New Structures for Physics*, Lecture Notes in Physics, pages 593–683. Springer, 2011.

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

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

8 Bob Coecke, Chris Heunen, and Aleks Kissinger. Categories of quantum and classical channels. *arXiv preprint arXiv:1305.3821*, 2013.

9 Bob Coecke and Robert W Spekkens. Picturing classical and quantum Bayesian inference. *Synthese*, 186(3):651–696, 2012.

10 E. Grefenstette and M. Sadrzadeh. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the EMNLP 2011*, 2011.

11 Z. Harris. *Mathematical Structures of Language*. Wiley, 1968.

12 D. Kartsaklis, M. Sadrzadeh, S. Pulman, and B. Coecke. Reasoning about meaning in natural language with compact closed categories and Frobenius algebras. *arXiv preprint arXiv:1401.5980*, 2014.

13 Dimitri Kartsaklis and Mehrnoosh Sadrzadeh. Prior disambiguation of word tensors for constructing sentence vectors. In *Proceedings of EMNLP 2013*, pages 1590–1601, 2013.

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

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

16 Joachim Lambek. The mathematics of sentence structure. *American mathematical monthly*, pages 154–170, 1958.

17 A. Preller and M. Sadrzadeh. Bell states and negative sentences in the distributed model of meaning. In P. Selinger B. Coecke, P. Panangaden, editor, *Electronic Notes in Theoretical Computer Science, Proceedings of the 6th QPL Workshop on Quantum Physics and Logic*. University of Oxford, 2010.

18 Anne Preller and Joachim Lambek. Free compact 2-categories. *Mathematical Structures in Computer Science*, 17(02):309–340, 2007.

19 M. Sadrzadeh, S. Clark, and B. Coecke. The Frobenius anatomy of word meanings I: subject and object relative pronouns. *Journal of Logic and Computation*, Advance Access, October 2013.

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

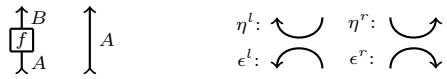
21 Peter Selinger. Dagger compact closed categories and completely positive maps. *Electronic Notes in Theoretical Computer Science*, 170:139–163, 2007.

22 Peter Selinger. A survey of graphical languages for monoidal categories. In Bob Coecke, editor, *New structures for physics*, pages 289–355. Springer, 2011.

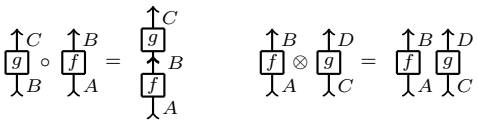
23 Peter D Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, 2010.

A Graphical calculus

Monoidal categories are complete with regard to a graphical calculus [22] which depicts derivations in their internal language very intuitively, thus simplifying the reading and the analysis. Objects are represented as labelled wires, and morphisms as boxes with input and output wires. The η- and ϵ-maps are given as half-turns.

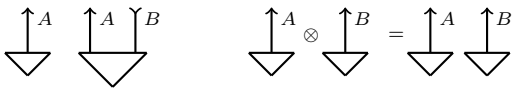


Composing morphisms amounts to connecting outputs to inputs, while the tensor product is simply juxtaposition:



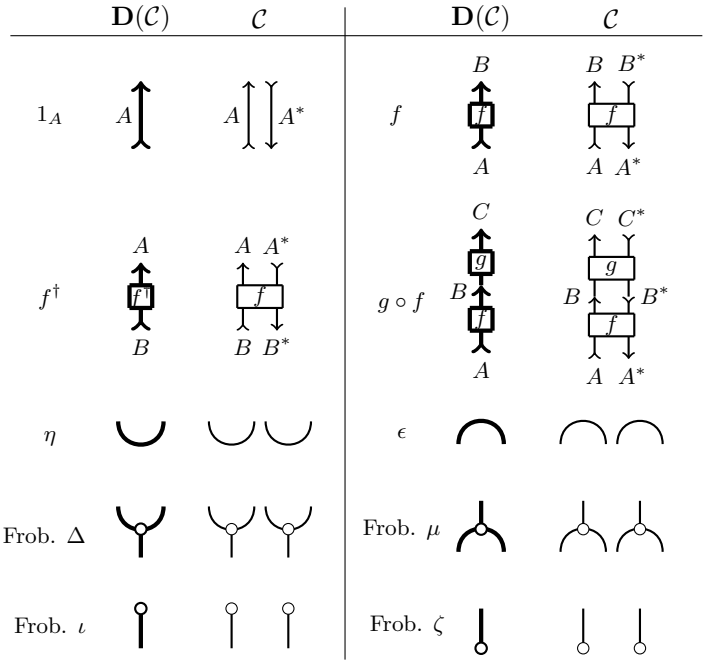
In this language, the yanking equations (2) get an intuitive visual justification (here for the first two identities):

For a given object A, we define a *state* of A to be a morphism $I \to A$. If A denotes a vector space, we can think of a state as a specific vector living in that space. In our graphical language the unit object I can be omitted, leading to the following representation of states:



Note that the second diagram from the left depicts an *entangled* state of $A \otimes B$; product states (such as the rightmost one) are simple juxtapositions of two states.

B Translation from $\mathcal{C}$ to $\mathbf{D}(\mathcal{C})$



C Linguistic intuition

In order to deal with lexical ambiguity we firstly need to understand its nature. In other words, we are interested to study in what way an ambiguous word differs from an unambiguous one, and what is the defining quality that makes this distinction clear. On the surface, the answer to these questions seems straightforward: an ambiguous word is one with more than one lexicographic entries in the dictionary. However, this definition fits well only to homonymous cases, in which due to some historical accident words that share the same spelling and pronunciation refer to completely unrelated concepts. Indeed, while the number of meanings of a homonymous word such as 'bank' is almost fixed across different dictionaries, the same is not true for the small (and overlapping) variations of senses that might be listed under a word expressing a polysemous case.

The crucial distinction between homonymy and polysemy is that in the latter case a word still expresses a coherent and self-contained concept. Recall the example of the polysemous use of 'bank' as a financial institution and the building where the services of the institution are offered; when we use the sentence 'I went to the bank' (with the financial meaning of the word in mind) we essentially refer to *both* of the polysemous meanings of 'bank' at the same time—at a higher level, the word 'bank' expresses an abstract but concise concept that encompasses all of the available polysemous meanings. On the other hand, the fact that the same name can be used to describe a completely different concept (such as a river bank or a number of objects in a row) is nothing more than an unfortunate coincidence expressing lack of specification. Indeed, a listener of the above sentence can retain a small amount of uncertainty regarding the true intentions of the sayer; although her first guess would be that 'bank' refers to the dominant meaning of financial institution (including *all related polysemous meanings*), a small possibility that the sayer has actually visited a river bank still remains. Therefore, in the absence of sufficient context, the meaning of a homonymous word is more reliably expressed as a *probabilistic mixing* of the unrelated individual meanings.

In a distributional model of meaning where a homonymous word is represented by a single vector, the ambiguity in meaning has been collapsed into a convex combination of the relevant sense vectors; the result is a vector that can be seen as the average of all senses, inadequate to reflect the meaning of any of them in a reliable way. We need a way to avoid that. In natural language, ambiguities are resolved with the introduction of context (recall that meaning is use), which means that for a compositional model of meaning the resolving mechanism is the compositional process itself. We would like to retain the ambiguity of a homonymous word when needed (i.e. in the absence of appropriate context) and allow it to collapse only when the context defines the intended sense, during the compositional process.

In summary, we seek an appropriate model that will allows us: (a) to express homonymous words as probabilistic mixings of their individual meanings; (b) to retain the ambiguity until the presence of sufficient context that will eventually resolve it during composition time; (c) to achieve all the above in the multi-linear setting imposed by the vector space semantics of our original model.

D From Theory to Practice

The purpose of this appendix is to show how the theoretical ideas presented in this paper can take a concrete form using standard natural language processing techniques. The setting we present below has been used for the mini-experiments in Sect. 3.5. We approach the creation of density matrices as a three-step process: (a) we first produce an ambiguous semantic space; (b) we apply a word sense induction method on it in order to associate each word with a set of sense vectors; and finally (c) we use the sense vectors in order to create a density matrix for each word. These steps are described in separate sections below.

D.1 Creating a Concrete Semantic Space

We train our basic vector space using ukWaC, a corpus of English text with 2 billion words (100 million sentences). The basis of the vector space consists of the 2,000 most frequent content words (nouns, verbs, adjectives, and adverbs), excluding a list of *stop words*.[2] Furthermore, the vector space is lemmatized and unambiguous regarding syntactic information; in other words, each vector is uniquely identified by a (*lemma,pos-tag*) pair, which means for example that 'book' as a noun and 'book' as a verb are represented by different meaning vectors. The weights of each vector are set to the ratio of the probability of the context word c_i given the target word t to the probability of the context word overall, as follows:

$$v_i(t) = \frac{p(c_i|t)}{p(c_i)} = \frac{\text{count}(c_i, t) \cdot \text{count}(total)}{\text{count}(t) \cdot \text{count}(c_i)}$$

where $\text{count}(c_i, t)$ refers to how many times c_i appears in the context of t (that is, in a 5-word window at either side of t) and $\text{count}(total)$ is the total number of word tokens in the corpus.

D.2 Word Sense Induction

The notion of word sense induction, that is, the task of detecting the different meanings under which a word appears in a text, is intimately connected with that of distributional hypothesis—that the meaning of a word is always context-dependent. If we had a way to

[2] That is, very common words with low information content, such as the verbs 'get' and 'take' or adverbs like 'really' and 'always'.

Meaning 1: 24070 contexts
port owner cargo fleet sailing ferry craft Navy merchant cruise navigation officer metre voyage authority deck coast launch fishery island charter Harbour pottery radio trip pay River Agency Scotland sell duty visit fish insurance skipper Roman sink War shore sail town Coastguard assistance Maritime registration call rescue bank Museum captain incident customer States yacht mooring barge comply landing Ireland sherd money Scottish tow tug maritime wreck board visitor tanker freight purchase lifeboat
Meaning 2: 5930 contexts
clot complication haemorrhage lymph stem VEGF Vitamin glucose penis endothelium retinopathy spasm antibody clotting AMD coagulation marrow lesion angina blindness medication graft vitamin vasoconstriction virus proliferation Ginkgo diabetic ventricle thickening tablet anaemia thrombus Vein leukocyte scleroderma stimulation degeneration homocysteine Raynaud breathe mediator Biloba Diabetes LDL metabolism Gene infiltrate atheroma arthritis lymphocyte lobe C's histamine melanoma gut dysfunction vitro triglyceride infarction lipoprotein

Table 2 Derived Meanings for Word 'Vessel'.

create a vectorial representation for the contexts in which a specific word occurs, then, a clustering algorithm could be applied in order to create groupings of these contexts that hopefully reveal different usages of the word—*different meanings*—in the training corpus.

This intuitive idea was first presented by Schütze [20] in 1998, and more or less is the cornerstone of every unsupervised word sense induction and disambiguation method based on semantic word spaces up to today. The approach we use is a direct variation of this standard technique. For what follows, we assume that each word in the vocabulary has already been assigned to an *ambiguous* semantic vector by following typical distributional procedures, for example similar to the setting described in Sect. D.1.

We assume for simplicity that the context is defined at the sentence level. First, each context for a target word w_t is represented by a *context vector* of the form $\frac{1}{n}\sum_{i=1}^{n}|w_i\rangle$, where $|w_i\rangle$ is the semantic vector of some other word $w_i \neq w_t$ in the same context. Next, we apply hierarchical agglomerative clustering on this set of vectors in order to discover the latent senses of w_t. Ideally, the contexts of w_t will vary according to the specific meaning in which this word has been used. Table 2 provides a visualization of the outcome of this process for the ambiguous word 'vessel'. Each meaning is visualized as a list of the most dominant words in the corresponding cluster, ranked by their TF-IDF values.

We take the centroid of each cluster as the vectorial representation of the corresponding sense/meaning. Thus, each word w is initially represented by a tuple $(|w\rangle, S_w)$, where $|w\rangle$ is the ambiguous semantic vector of the word as created by the usual distributional practice, and S_w is a set of *sense vectors* (that is, centroids of context vectors clusters) produced by the above procedure.

Note that our approach takes place at the vector level (as opposed to tensors of higher order), so it provides a natural way to create sets of meaning vectors for "atomic" words of the language, that is, for nouns. It turns out that the generalization of this to tensors of higher order is straightforward, since the clustering step has already equipped us with a number of sets consisting of context vectors, each one of which stands in one-to-one correspondence with a set of contexts reflecting a different semantic usage of the higher-order word. One then can use, for example, the argument "tensoring and summing" procedure of [10] (briefly described in Sect. 2.4) in order to compute the meaning of the ith sense of a word of arity n

as:

$$|word\rangle_i = \sum_{c \in C_i} \bigotimes_{k=1}^{n} |arg_{k,c}\rangle \tag{19}$$

where C_i is the set of contexts associated with the ith sense, and $arg_{k,c}$ denotes the kth argument of the target word in context c. Of course, more advanced statistical methods could be also used for learning the sense tensors from the provided partitioning of the contexts, as long as these methods respect the multi-linear nature of the model. This completes the word sense induction step.

D.3 Creating Density Matrices

We have now managed to equip each word with a set of sense vectors (or higher-order tensors, depending on its grammatical type). Assigning a probability to each sense is trivial and can be directly derived by the number of times the target word occurs under a specific sense divided by the total occurrences of the word in the training corpus. This creates a statistical ensemble of state vectors and probabilities that can be used for computing a density matrix for the word according to Def. 2.