Translating and Evolving: Towards a Model of Language Change in DisCoCat

Bradley, T.-D.; Lewis, M.; Master, J.; Theilman, B.

DOI
10.4204/EPTCS.283.4
Publication date
2018
Document Version
Final published version
Published in
Electronic Proceedings in Theoretical Computer Science
License
CC BY

Link to publication

Citation for published version (APA):
Bradley, T-D., Lewis, M., Master, J., & Theilman, B. (2018). Translating and Evolving: Towards a Model of Language Change in DisCoCat. *Electronic Proceedings in Theoretical Computer Science*, *283*, 50-61. https://doi.org/10.4204/EPTCS.283.4

Translating and Evolving: Towards a Model of Language Change in DisCoCat

Tai-Danae Bradley
Graduate Center, CUNY
tbradley@gradcenter.cuny.edu

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

Jade Master
Dept. Mathematics, UC Riverside
jmast003@ucr.edu

Brad Theilman
Gentner Lab, UC San Diego
btheilma@ad.ucsd.edu

The categorical compositional distributional (DisCoCat) model of meaning developed by [7] has been successful in modeling various aspects of meaning. However, it fails to model the fact that language can change. We give an approach to DisCoCat that allows us to represent language models and translations between them, enabling us to describe translations from one language to another, or changes within the same language. We unify the product space representation given in [7] and the functorial description in [16], in a way that allows us to view a language as a catalogue of meanings. We formalize the notion of a lexicon in DisCoCat, and define a dictionary of meanings between two lexicons. All this is done within the framework of monoidal categories. We give examples of how to apply our methods, and give a concrete suggestion for compositional translation in corpora.

1 Introduction

Language allows us to communicate, and to compose words in a huge variety of ways to obtain different meanings. It is also constantly changing. The compositional distributional model of [7] describes how to use compositional methods within a vector space model of meaning. However, this model, and others that are similar [4, 20], do not have a built in notion of language change, or of translation between languages.

In contrast, many statistical machine translation systems currently use neural models, where a large network is trained to be able to translate words and phrases [21, 8]. This approach does not make use of the grammatical structure which allows you to build translations of phrases from the translations of individual words. In this paper we define a notion of translation between two compositional distributional models of meaning which constitutes a first step towards unifying these two approaches.

Modeling translation between two languages also has intrinsic value, and doing so within the DisCoCat framework means that we can use its compositional power. In section 3.1, we provide a categorical description of translation between two languages that encompasses both updating or amending a language model and translating between two distinct natural languages.

In order to provide this categorical description, we must first introduce some preliminary concepts. In section 3.2 we propose a unification of the product space representation of a language model of [7] and the functorial representation of [16]. This allows us to formalize the notion of lexicon in section 3.3 which had previously been only loosely defined in the DisCoCat framework. We then show how to build a dictionary between two lexicons and give an example showing how translations can be used to model an update or evolution of a compositional distributional model of meaning. In section 3.4 we give a concrete suggestion for automated translation between corpora in English to corpora in Spanish.

2 Background

2.1 Categorical Compositional Distributional Semantics

Categorical compositional distributional models [7] successfully exploit the compositional structure of natural language in a principled manner, and have outperformed other approaches in Natural Language Processing (NLP) [9, 14]. The approach works as follows. A mathematical formalization of grammar is chosen, for example *Lambek's pregroup grammars* [18], although the approach is equally effective with other categorial grammars [6]. Such a categorial grammar allows one to verify whether a phrase or a sentence is grammatically well-formed by means of a computation that establishes the overall grammatical type, referred to as *a type reduction*. The meanings of *individual* words are established using a distributional model of language, where they are described as vectors of co-occurrence statistics derived automatically from corpus data [19]. The categorical compositional distributional programme unifies these two aspects of language in a compositional model where grammar mediates composition of meanings. This allows us to derive the meaning of sentences from their grammatical structure, and the meanings of their constituent words. The key insight that allows this approach to succeed is that both pregroup grammars and the category of vector spaces carry the same abstract structure [7], and the same holds for other categorial grammars since they typically have a weaker categorical structure.

The categorical compositional approach to meaning uses the notion of a *monoidal category*, and more specifically a *compact closed* category to understand the structure of grammar and of vector spaces. For reasons of space, we do not describe the details of the compositional distributional approach to meaning. Details can be found in [7, 16], amongst others. We note only that instead of using a pregroup as our grammar category, we use the free compact closed category $J = \mathscr{C}(\mathscr{B})$ generated over a set of types $\mathscr{B}$, as described in [24, 23].

3 Translating and Evolving

The categorical model has proved successful in a number of natural language processing tasks [9, 14], and is flexible enough that it can be extended to include ambiguity [22] and changes of the semantic category [5, 1]. These formalisms have allowed for connections between semantic meanings. By representing words as density matrices, a variant of Löwner ordering has been used to measure the degree of entailment between two words [25, 3]. A more simple notion of similarity has been implemented in the distributional model by using dot product [7]. However, these notions of similarity are not built into the formalism of the model. This section defines the notion of a categorical language model which keeps track of internal relationships between semantic meanings.

So far the implementation of these models has been static. In this section, we define a notion of translation which comprises a first step into bringing dynamics into these models of meaning. We show how a language model can be *lexicalized*, i.e. how vocabulary can be attached to types and vectors and introduce a category of lexicons and translations between them. This allows dictionary between phrases in one language model and the phrases in another.

3.1 Categorical Language Models and Translations

Definition 3.1. Let J be a category which is freely monoidal on some set of grammatical types. A **distributional categorical language model** or **language model** for short is a strong monoidal functor

$$F\colon (J,\cdot) \to (\mathsf{FVect},\otimes)$$

If J is compact closed then the essential image of F inherits a compact closed structure. All of the examples we consider will use the canonical compact closed structure in FVect. However, this is not a requirement of the general approach, and other grammars that are not compact closed my be used, such as Chomsky grammars [11] or Lambek monoids [6].

Distributional categorical language models do not encapsulate everything about a particular language. In fact, there are many possible categorical language models for the same language and there is a notion of translation between them.

Definition 3.2. A **translation** $T = (j, \alpha)$ from a language model $F\colon J \to \mathsf{FVect}$ to a language model $F'\colon J' \to \mathsf{FVect}$ is a monoidal functor $j\colon J \to J'$ and a monoidal natural transformation $\alpha\colon F \Rightarrow F' \circ j$. Pictorially, (j, α) is the following 2-cell

$$\begin{tikzcd} J \arrow[r, "F"] \arrow[d, "j"'] & \mathsf{FVect} \\ J' \arrow[ur, "F'"'] \arrow[u, phantom, shift left=6, "\Downarrow\alpha"] & \end{tikzcd}$$

Given another language model $F''\colon J'' \to \mathsf{FVect}$ and a translation $T' = (j', \alpha')$ the composite translation is computed pointwise. That is, $T' \circ T$ is the translation $(j' \circ j, \alpha' \circ \alpha)$ where $\alpha' \circ \alpha$ is the vertical composite of the natural transformations α and α'.

Definition 3.3. Let $\mathsf{DisCoCat}$ be the category with distributional categorical language models as objects, translations as morphisms, and the composition rule described above.

This category allows us to define ways of moving between languages. The most obvious application of this is that of translation between two languages such as English and Spanish. However, the translation could also be from a simpler language to a more complex language, which we think of as learning, and it could be within a shared language, where we see the language evolving.

3.2 The Product Space Representation

In [7] the *product space representation* of language models was introduced as a way of linking grammatical types with their instantiation in FVect. The idea is that the meaning computations take place in the category $J \times \mathsf{FVect}$ where J is a pregroup or free compact closed category. Let p be a sentence, that is a sequence of words $\{w_i\}$ whose grammatical types reduce to the sentence type s. To compute the meaning of p you:

- Determine both the grammatical type g_i and distributional meaning $\mathbf{v}_i$ in V_{g_i} where V_{g_i} is a meaning space for the grammatical type g_i.
- Using the monoidal product and tensor product, obtain the element $g_1 \cdot \ldots \cdot g_n \in J$ and $\mathbf{v}_1 \otimes \cdots \otimes \mathbf{v_n} \in V_{g_1} \otimes \ldots \otimes V_{g_n}$.
- Let $r\colon g_1 \cdot \cdots \cdot g_n \to s$ be a type reduction in J. There is a linear transformation $M\colon V_{g_1} \otimes \cdots \otimes V_{g_n} \to V_s$ given by matching up the compact closed structure in J with the canonical compact closed structure in FVect. Apply M to the vector $\mathbf{v}_1 \otimes \ldots \otimes \mathbf{v}_n$ to get the distributional meaning of your sentence.

The product space $J \times \mathsf{FVect}$ provides a setting in which the meaning computations take place but it does not contain all of the information required to compute compositional meanings of sentences. To do this requires an assignment of every grammatical type to a vector space and every type reduction to a linear transformation in a way which preserves the compact closed structure of both categories. This suggests that there is a compact closed functor $F\colon J \to \mathsf{FVect}$ lurking beneath this approach. With this in mind we introduce a new notion of the product space representation using the Grothendieck construction [12]. In order to use the Grothendieck construction we first need to interpret vector spaces as categories.

In this paper, we will do this in two mostly trivial ways which do not take advantage of the vector space structure in FVect. The first way we will turn vector spaces into categories is via the discrete functor

$$D\colon \mathsf{FVect} \to \mathsf{Cat}$$

D assigns each vector space V to the discrete category of its underlying set. For a linear transformation $M\colon V \to W$, $D(M)$ is the unique functor from $D(V)$ to $D(W)$ which agrees with M on the elements of V.

There is another way to generate free categories from sets.

Definition 3.4. Let V be a finite dimensional real vector space. Then, the **free chaotic category** on V, denoted $C(V)$, is a category where

- objects are elements of V and,
- for all $\mathbf{u}$,$\mathbf{v}$ in V we include a unique arrow $d(\mathbf{u},\mathbf{v})\colon \mathbf{u} \to \mathbf{v}$ labeled by the Euclidean distance $d(\mathbf{u},\mathbf{v})$ between $\mathbf{u}$ and $\mathbf{v}$.

This construction extends to a functor $C\colon \mathsf{FVect} \to \mathsf{Cat}$. For $M\colon V \to W$, define $C(M)\colon C(V) \to C(W)$ to be the functor which agrees with M on objects and sends arrows $d(\mathbf{u},\mathbf{v})$ to $d(M\mathbf{u},M\mathbf{v})$.

The morphisms in $C(V)$ for a vector space V allow us to keep track of the relationships between different words in V.

We now give a definitions of the product space representation in terms of the Grothendieck construction which depends on a choice of functor $K\colon \mathsf{FVect} \to \mathsf{Cat}$.

Definition 3.5. Let $F\colon J \to \mathsf{FVect}$ be a language model and let $K\colon \mathsf{FVect} \to \mathsf{Cat}$ be a faithful functor. The **product space representation** of F with respect to K, denoted $\mathtt{PS}_K(F)$, is the Grothendieck construction of $K \circ F$. Explicitly, $\mathtt{PS}_K(F)$ is a category where

- an object is a pair $(g,\mathbf{u})$ where g is an object of J and $\mathbf{u}$ is an object of $K \circ F(g)$
- a morphism from $(g,\mathbf{u})$ to $(h,\mathbf{v})$ is a tuple (r,f) where $r\colon g \to h$ is a morphism in J and $f\colon K \circ F(r)(\mathbf{u}) \to \mathbf{v}$ is a morphism in $K \circ F(h)$
- the composite of $(r',f')\colon (g,\mathbf{u}) \to (h,\mathbf{v})$ and $(r,f)\colon (h,\mathbf{v}) \to (i,\mathbf{x})$ is defined by

$$(r,f)\circ(r',f') = (r \circ r', f \circ (K \circ F)(r)(f'))$$

Remark 3.6. Because K is faithful, it is an equivalence of categories onto its essential image in Cat. Because monoidal structures pass through equivalences $K \circ F\colon J \to \operatorname{Im} K \circ F$ is a monoidal functor where $\operatorname{Im} K \circ F$ denotes the essential image of $K \circ F$.

When K is equal to the discrete category functor D, then the product space representation is the **category of elements of** F. This is a category where

- objects are pairs $(g,\mathbf{u})$ where g is a grammatical type and $\mathbf{u}$ is a vector in $F(g)$.
- a morphism $r\colon (g,\mathbf{u}) \to (h,\mathbf{v})$ is a type reduction $r\colon g \to h$ such that $F(r)(\mathbf{u}) = \mathbf{v}$

In this context we can compare the product space representation in Definition 3.5 with the representation introduced in [7] to see that they are not the same. One difference is that $\mathtt{PS}_D(F)$ only includes the linear transformations that correspond to type reductions and not arbitrary linear transformations. This narrows down the product space representation to a category that characterizes the meaning computations which can occur. Also, the meaning reductions in $\mathtt{PS}_D(F)$ correspond to morphisms in the product space representation whereas before they occurred within specific objects of the product space. Using this definition of the product space representation, we are able to formally introduce a lexicon into the model and understand how these lexicons are affected by translations.

When $K = C$ as in Definition 3.4 the product space representation is as follows:

- objects are pairs $(g, \mathbf{u})$ where g is a grammatical type and $\mathbf{u}$ is a vector in $F(g)$.
- a morphism $(r,d)\colon (g,\mathbf{u}) \to (h,\mathbf{v})$ is:
  - a type reduction $r\colon g \to h$
  - a positive real number $d\colon C \circ F(r)(\mathbf{u}) \to \mathbf{v}$

Now, objects in $\mathtt{PS}_C(F)$ are pairs of grammatical types and *vectors*, rather than vector spaces. We can therefore see $\mathtt{PS}_C(F)$ as a catalogue of all possible meanings associated with grammatical types. The linear transformations available in $\mathtt{PS}_C(F)$ are only those that are derived from the grammar category.

Proposition 3.7 ($\mathtt{PS}_K(F)$ is monoidal)**.** For $K = C$ and $K = D$, $\mathtt{PS}_K(F)$ is a monoidal category with monoidal product given on objects by

$$(g,\mathbf{u}) \otimes (h,\mathbf{v}) = (g \cdot h, \Phi_{g,h}(\mathbf{u} \otimes \mathbf{v}))$$

and on morphisms by

$$(r,f) \otimes (r',f') = (r \cdot r', \Phi_{g,h}(f \otimes f'))$$

where $\Phi_{g,h}\colon K \circ F(g) \otimes K \circ F(h) \to K \circ F(g \cdot h)$ is the natural isomorphism included in the data of the monoidal functor $K \circ F$.

Proof. Adapted from Theorem 38 of [2]. □

The fact that $\mathtt{PS}_K(F)$ is monoidal enables us to use the powerful graphical calculus available for monoidal categories. Previously the monoidal graphical calculus has only been used to pictorially reason about grammatical meanings. Because the elements of the product space representation represent both the syntactic and semantic meaning, this proposition tells us that we can reason graphically about the entire meaning of our phrase.

The product space construction also applies to translations:

Proposition 3.8 (Translations are monoidal)**.** Let $K\colon \mathsf{FVect} \to \mathsf{Cat}$ be a fully faithful functor. Then there is a functor $\mathtt{PS}_K\colon \mathsf{DisCoCat} \to \mathsf{MonCat}$, where MonCat is the category where objects are monoidal categories and morphisms are strong monoidal functors, that sends

- language models $F\colon J \to \mathsf{Cat}$ to the monoidal category $\mathtt{PS}_K(F)$
- translations $T = (j,\alpha)$ to the strong monoidal functor $\mathtt{PS}_K(T)\colon \mathtt{PS}_K(F) \to \mathtt{PS}_K(F')$ where the functor $\mathtt{PS}_K(T)$ acts as follows:
  - On objects, $\mathtt{PS}_K(T)$ sends $(g,\mathbf{u})$ to $(j(g), \alpha_g \mathbf{u})$.
  - Suppose $(r,f)\colon (g,\mathbf{u}) \to (h,\mathbf{v})$ is a morphism in $\mathtt{PS}_K(F)$ so that $r\colon g \to h$ is a reduction in J and $f\colon F(r)(\mathbf{u}) \to \mathbf{v}$ is a morphism in $F(h)$. Then $\mathtt{PS}_K(T)$ sends (r,f) to the pair $(j(r), \alpha_h \circ f)$.

Proof. Adapted from Theorem 39 in [2]. □

3.3 Lexicons

Using our definition of the product space representation, we are able to formally introduce a lexicon into the model and describe how these lexicons are affected by translations. In what follows we fix $K = C$ in all product space constructions and denote $\mathtt{PS}_C(F)$ as $\mathtt{PS}(F)$. We also use the notation F_C for $C \circ F$.

Definition 3.9. Let F be a categorical language model and let W be a finite set of words, viewed as discrete category. Then a **lexicon for** F is a functor $\ell \colon W \to \mathtt{PS}(F)$. This corresponds to a function from W into the objects of $\mathtt{PS}(F)$.

Lexicons can be extended to arbitrary phrases in the set of words W. Phrases are finite sequences of words $v_1 \dots v_n \in W^*$ where W^* is the free monoid on W. The function ℓ assigns to each $v_i \in W$ the pair $(g_i, \mathbf{v_i})$ corresponding to its grammatical type g and its semantic meaning $\mathbf{v_i} \in F(g_i)$. Because W^* is free, this defines a unique object in $\mathtt{PS}(F)$:

$$(g, \mathbf{v}) := \otimes_{i=1}^{n} \ell(v_i) = (g_1, \mathbf{v}_1) \otimes \dots \otimes (g_n, \mathbf{v}_n) = (g_1 \cdots g_n, \mathbf{v}_1 \otimes \dots \otimes \mathbf{v}_n)$$

where g_i is the grammatical type of v_i and $\mathbf{v}_i$ is the semantic meaning of v_i for $i \in \{1, ..., n\}$. The extension of ℓ to W^* will be denoted by

$$l^* \colon W^* \to \mathtt{PS}(F).$$

Example 3.10. Let $J = \mathscr{C}(\{n, s\})$ be the free compact closed category on the grammatical types of nouns and sentences. Then, for the phrase $u =$ *Rose likes Rosie*, a lexicon for F gives the unique element

$$\ell(u) = (g, \mathbf{u}) = (n, \mathbf{Rose}) \otimes (n^r s n^l, \mathbf{likes}) \otimes (n, \mathbf{Rosie})$$

In $\mathtt{PS}(F)$, the grammar type $n^r s n^l$ reduces to s via the morphism $r = \varepsilon_n \; 1_s \; \varepsilon_n$ and so we get a reduction

$$(n, \mathbf{Rose}) \otimes (n^r s n^l, \mathbf{likes}) \otimes (n, \mathbf{Rosie}) \xrightarrow{(r, F(r))} (s, \mathbf{Rose\ likes\ Rosie})$$

To fully specify a translation between two lexicons it is not necessary to manually match the words in each corpora. This is because a relation between the phrases in the corpora can be derived from a translation between the language models.

Definition 3.11. Let $\ell \colon W \to \mathtt{PS}(F)$ and $m \colon V \to \mathtt{PS}(G)$ be lexicons and let T be a translation from F to G. Then, the F-G **dictionary** with respect to T is the comma category

$$(\mathtt{PS}(T) \circ \ell^*) \downarrow m^*$$

denoted by Dict_T. Since W and V are discrete categories, $(\mathtt{PS}(T) \circ \ell^*) \downarrow m^*$ is a set of triples $(p, (r, d), q)$ where $p \in W^*$, $q \in V^*$ and $(r, d) \colon (\mathtt{PS}(T) \circ \ell)(p) \to m(q)$ is a morphism in $\mathtt{PS}(G)$. Explicitly, let

$$\ell(p) = (g, \mathbf{p}) \quad \text{and} \quad m(q) = (g', \mathbf{q})$$

then (r, d) is

- a type reduction $r \colon j(g) \to g'$ in the grammar category J
- a morphism d in $C \circ G(g')$. Recall from Definition 3.4 that this corresponds to a real number $d(\mathbf{p}', \mathbf{q})$ denoting the distance between $\mathbf{p}'$ and $\mathbf{q}$ in $G(g')$. Here, $\mathbf{p}'$ is the vector that results from applying the translation and any grammatical reductions. Namely, $\mathbf{p}' = (C \circ G(r) \circ \alpha_g)\mathbf{p}$ i.e., we firstly translate the vector $\mathbf{p}$ into $\mathtt{PS}(F')$, then apply the linear map corresponding to the reduction r, and finally send the resulting vector to its corresponding object in the chaotic category.

Dict_T and allows us to keep track of the distances between phrases in W to phrases in W' in a compositional way; similarities between phrases are derived from similarities between the constituent words.

Let k be a positive real number. Then define $\mathsf{Dict}_{T,k}$ to be the relation which pairs two words in Dict_T if the distance d between their semantic meanings is less than or equal to k. The purpose of this is to say that we are interested in pairs of words and phrases which do not have to be identical, but whose meaning is sufficiently close.

Example 3.12 (Syntactic simplification). We give an example of a translation from a language with several noun types, accounting for singular n_s and plural n_p nouns to a language with one noun type. To start, suppose W^* is the free monoid on the set $\{$*Rosie, wears, boots, shoes, bikini*$\}$ and set $V = W$. Suppose $\mathscr{B} = \{n_s, n_p, n, s\}$ and $\mathscr{B}' = \{n, s\}$ so that $J = \mathscr{C}(\mathscr{B})$ and $J' = \mathscr{C}(\mathscr{B}')$, and let $T = (j, \alpha)$ be a translation from F to F', where the language model F has $F(n) = N$ and $F(n_s) = F(n_p) = N'$ where $N \cong \mathbb{R}^3$ is generated by $\{\mathbf{boots}, \mathbf{shoes}, \mathbf{bikini}\}$ and $N' = N \times \mathbb{R}$ where the extra dimension records the quantity conveyed by the noun. Let $F(s) = S$ be a one-dimensional space spanned by $\mathbf{1}$, which denotes *surprise*. For the purposes of this example we will normalize all non-zero values in S to the vector $\mathbf{1}$. This gives only two attainable values in S; $\mathbf{s} = \mathbf{1}$ meaning that the sentence is surprising, and $\mathbf{s} = \mathbf{0}$ meaning that the sentence is not surprising. The language model F' agrees with F on both n and s, that is $F'(n) = N$ and $F'(s) = S$. The functor j is given by $j(n) = j(n_s) = j(n_p) = n$ and $j(s) = s$ and finally, the components of α are by the identity on every space except for N' where it is defined as the canonical projection onto the first three coordinates.

$$\begin{tikzcd}
W^* \arrow[d, "\ell"] & W^* \arrow[d, "\ell'"] \\
\mathtt{PS}(F) \arrow[r, "\mathtt{PS}(T)"'] & \mathtt{PS}(F')
\end{tikzcd}$$

defined as follows:

$$\begin{aligned}
W &\xrightarrow{\quad\ell\quad} \mathtt{PS}(F) \\
\textit{Rosie} &\longmapsto (n_s, (2,5,3,1)^T) \\
\textit{boots} &\longmapsto (n_p, (1,0,0,2)^T) \\
\textit{a boot} &\longmapsto (n_p, (1,0,0,1)^T) \\
\textit{wears} &\longmapsto \left(n_m^r s n_m^l, \begin{pmatrix} 1 & 1 & 1 & 0 \\ -1 & -1 & -1 & 0 \\ 1 & 1 & 1 & 0 \\ -2 & -2 & -1 & 1 \end{pmatrix} \right)
\end{aligned}$$

$$\begin{aligned}
W &\xrightarrow{\quad\ell'\quad} \mathtt{PS}(F') \\
\textit{Rosie} &\longmapsto (n, (2,5,3)^T) \\
\textit{boots} &\longmapsto (n, (1,0,0)^T) \\
\textit{wears} &\longmapsto \left(n^r s n^l, \begin{pmatrix} 1 & 1 & 1 \\ -1 & -1 & -1 \\ 1 & 1 & 1 \end{pmatrix} \right)
\end{aligned}$$

where we use n_m to denote either of n_s or n_p. The type of *wears* is polymorphic - it can take singular, plural, or mass nouns as subject and object. Here, we consider types with singular or plural arguments.

The functor $\mathtt{PS}(T)$ uses j and α to assign a tuple on the right to each to each tuple on the left, i.e.

$$\begin{tikzcd}
\mathtt{PS}(F) \arrow[r, "\mathtt{PS}(T)"] & \mathtt{PS}(F') \\
(n_s,(2,5,3,1)^T) \arrow[r, mapsto] & (n,(2,5,3)^T) \\
(n_p,(1,0,0,2)^T) \arrow[r, mapsto] & (n,(1,0,0)^T)
\end{tikzcd}$$

$$\left(n_m^r s n_m^l, \begin{pmatrix} 1 & 1 & 1 & 0 \\ -1 & -1 & -1 & 0 \\ 1 & 1 & 1 & 0 \\ -2 & -2 & -1 & 1 \end{pmatrix}\right) \longmapsto \left(n^r s n^l, \begin{pmatrix} 1 & 1 & 1 \\ -1 & -1 & -1 \\ 1 & 1 & 1 \end{pmatrix}\right)$$

In particular, each item on the right hand side is of the form $(\mathtt{PS}(T)\circ\ell)(u)$ where u is an element of W. Now, in F, we have:

$$\begin{aligned}
\textbf{Rosie wears boots} &= F(\varepsilon_n \cdot 1_s \cdot \varepsilon_n)(\textbf{Rosie} \otimes \textbf{wears} \otimes \textbf{boots}) \\
&= (\varepsilon_n \otimes 1_s \otimes \varepsilon_n)\left(\begin{pmatrix} 2 \\ 5 \\ 3 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 & 1 & 0 \\ -1 & -1 & -1 & 0 \\ 1 & 1 & 1 & 0 \\ -2 & -2 & -1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \\ 0 \\ 2 \end{pmatrix}\right) = 0, \text{ i.e. unsurprising}
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\textbf{Rosie wears a boot} &= F(\varepsilon_n \cdot 1_s \cdot \varepsilon_n)(\textbf{Rosie} \otimes \textbf{wears} \otimes \textbf{a boot}) \\
&= (\varepsilon_n \otimes 1_s \otimes \varepsilon_n)\left(\begin{pmatrix} 2 \\ 5 \\ 3 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 & 1 & 0 \\ -1 & -1 & -1 & 0 \\ 1 & 1 & 1 & 0 \\ -2 & -2 & -1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}\right) = -1 = 1 \text{ after normalization, i.e. surprising}
\end{aligned}$$

We translate *Rosie wears boots* by computing

$$\begin{aligned}
(\mathtt{PS}(T)\circ\ell)(\textit{Rosie wears boots}) &= \mathtt{PS}(T)((n_s, \textbf{Rosie}) \otimes (n_m^r s n_m^l, \textbf{wears}) \otimes (n_p, \textbf{boots}) \\
&= ((j(n_s), \alpha_{n_s}(\textbf{Rosie})) \otimes (j(n_m^r s n_m^l), \alpha_{n_m^r s n_m^l}(\textbf{wears})) \otimes (j(n_p), \alpha_{n_p}(\textbf{boots})) \\
&= \left(n, \begin{pmatrix} 2 \\ 5 \\ 3 \end{pmatrix}\right) \otimes \left(n^r s n^l, \begin{pmatrix} 1 & 1 & 1 \\ -1 & -1 & -1 \\ 1 & 1 & 1 \end{pmatrix}\right) \otimes \left(n, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}\right) = (\mathtt{PS}(T)\circ\ell)(\textit{Rosie wears a boot})
\end{aligned}$$

and applying the relevant reduction morphisms, we obtain:

$$\begin{aligned}
\textbf{Rosie wears boot}' &= F'(\varepsilon_n \cdot 1_s \cdot \varepsilon_n)(\textbf{Rosie}' \otimes \textbf{wears}' \otimes \textbf{boot}') \\
&= (\varepsilon_n \otimes 1_s \otimes \varepsilon_n)\left(\begin{pmatrix} 2 \\ 5 \\ 3 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 & 1 \\ -1 & -1 & -1 \\ 1 & 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}\right) = 0, \text{ i.e. unsurprising}
\end{aligned}$$

To translate the sentence *Rosie wears a boot* we perform the same matrix multiplication to obtain a value of 0 as well. In the language model F, these two sentences had distinct meanings. However, because F' cannot detect the quantity of a noun, their translations are both unsurprising.

This example shows how we can map a language with one grammatical structure onto another with a differing grammatical structure. In this case we have simplified the grammar, but we could also provide a translation (j,α) that maps the simpler grammar into the more complex grammar by identities and inclusion. The phenomenon of grammatical simplification is one that has been observed in various languages [27, 17]. This provides us with the beginnings of a way to describe these kinds of language evolution.

3.4 Translating Between English and Spanish

In this section we construct a partial translation from English and Spanish. The relationship between English grammar and Spanish grammar is not functional; there are multiple types in Spanish that a single type in English should correspond to and vice versa.

Let $J_E = \mathscr{C}(\{a_E, n_E\})$ be the grammar category for English and let $J_S = \mathscr{C}(\{a_S, n_S\})$ be the grammar category for Spanish. In English, adjectives multiply on the left to get a reduction

$$r : a_E n_E \to n_E$$

and in Spanish adjectives multiply on the right to get a reduction

$$q : n_S a_S \to n_S$$

Suppose there is a strict monoidal functor $j\colon J_E \to J_S$ which makes the assignment $j(a_E) = a_S$. We also wish to map the reduction r to the reduction q. This requires that $j(a_E n_E) = n_S a_S$. By monoidality this means that $j(a_E) = n_S$ and $j(n_E) = a_S$. A monoidal functor cannot capture this relationship because it must be single valued. However, if we choose to only translate either adjectives or nouns we can construct a translation.

Example 3.13 (Translation at the phrase level). In this example we choose to translate the fragment of English and Spanish grammar which includes nouns but not adjectives. We can also translate intransitive verbs from English to Spanish while keeping the functor between grammar categories single-valued.

Let $J_{En} = \mathscr{C}(\{n_E, s_E\})$ be the free compact closed category on the noun and sentence types in English and let J_S be the isomorphic category $\mathscr{C}\{n_S, s_S\}$ generated by the corresponding types in Spanish. Consider distributional categorical language models

$$F_{En}\colon J_{En} \to \mathsf{Cat} \text{ and } F_{Sp}\colon J_{Sp} \to \mathsf{Cat}$$

for English and Spanish respectively. Consider a fragment of these languages consisting of only nouns and intransitive verbs. Let $F_{En}(n) = N_{En}$, $F_{En}(s) = S_{En}$, $F_{Sp}(n) = N_{Sp}$ and $F_{Sp}(s) = S_{Sp}$. Lexicons for the two languages can be populated by learning representations from text.

To specify the translation $\mathtt{PS}(T)$ we set j to be the evident functor which sends English types to their corresponding type in Spanish. To define a natural transformation $\alpha : F_{En} \to F_{Sp} \circ j$ it suffices to define α on the basic grammatical types which are not the nontrivial product of any two other types. Because α is a monoidal natural transformation, we have that $\alpha_{gh} = \alpha_g \otimes \alpha_h$ for every product type gh.

If there were only one grammatical type g, then the language models would have no grammatical content and the translation would consist of a single linear transformation between words in English to words in Spanish. Learning this transformation is in fact a known method for implementing word-level machine translation, as outlined in [21, 13].

However, in general we need the natural transformation α to commute with the type reductions in $\mathscr{C}(\{n_E, s_E\})$. Indeed, consider **dogs** $\in N_{En}$, **run** $\in N_{En} \otimes S_{En}$, **perros** $\in N_{Sp}$, **corren** $\in N_{Sp} \otimes S_{Sp}$. We require that

$$\begin{tikzcd}
N_{En} \otimes N_{En} \otimes S_{En} \arrow[r, "\varepsilon_{N_{En}} \otimes 1_{En}"] \arrow[d, "\alpha_{n \cdot n^l \cdot s}"] & S_{En} \arrow[d, "\alpha_s"] \\
N_{Sp} \otimes N_{Sp} \otimes S_{Sp} \arrow[r, "\varepsilon_{N_{Sp}} \otimes 1_{Sp}"'] & S_{Sp}
\end{tikzcd}$$

commutes i.e. that if we first reduce **dogs** $\otimes$ **run** to obtain **dogs run** and then translate to **perros corren**, we get the same as if we translate each word first, sending **dogs** $\otimes$ **run** to **perros** $\otimes$ **corren** and then reduce to **perros corren**. Because these meaning reductions are built using dot products, this requirement is equivalent to the components of α being unitary linear transformations. In general, a linear transformation learned from a corpora will not be unitary. In this case we can replace α_g with the unitary matrix which is closest to it. This is a reasonable approximation because translations should preserve relative similarities between words in the same language.

4 Future Work

We have defined a category DisCoCat which contains the categorical compositional distributional semantics of [7] as objects and ways in which they can change as morphisms. We then outlined how this category can be used to translate, update or evolve different distributional compositional models of meaning.

There is a wide range of future work on this topic that we would like to explore. Some of the possible directions are the following:

- In this paper, we failed to construct a complete translation from English to Spanish using the definitions in this paper. The difficulty arose from the lack of a functional relationship between the two languages. To accommodate this, translations between language models can be upgraded by replacing functors with profunctors. This would include replacing the grammar transformation j with a monoidal profunctor between the grammar categories. Because relationships between semantic meanings are also multi-valued we plan on replacing the components of the natural transformation with profunctors as well.
- This model can be improved to take advantage of the metric space structure of vector spaces to form dictionaries in a less trivial way. This would give a more intelligent way of forming translation dictionaries between two languages
- Whilst we have taken care to ensure that the category of language models we use is monoidal, we have not yet taken advantage of the diagrammatic calculus that is available to us. This is something we would like to do in future work.
- We can better understand how language users negotiate a shared meaning space as in [26], by modeling this as translations back and forth between agents. This will enhance the field of evolutionary linguistics by giving a model of language change that incorporates categorial grammar.
- We would like to use the methods here to implement computational experiments by creating compositional translation matrices for corpora in two different languages. These models of translation may also be used to make the previous computational experiments such as[10] [15] more flexible.

5 Acknowledgments

Thanks go to Joey Hirsh for enlightening and fruitful discussions of the issues in this paper. We would also like to thank John Baez for catching a mistake in an early draft of this paper. We gratefully acknowledge support from the Applied Category Theory 2018 School at the Lorentz Center at which the research was conceived and developed, and funding from KNAW. Martha Lewis gratefully acknowledges support from NWO Veni grant 'Metaphorical Meanings for Artificial Agents'.

References

[1] Yaared Al-Mehairi, Bob Coecke & Martha Lewis (2016): *Categorical Compositional Cognition*. In: *International Symposium on Quantum Interaction*, Springer, pp. 122–134, doi:10.1007/978-3-319-52289-0_10.

[2] John C Baez, John Foley, Joseph Moeller & Blake S Pollard (2017): *Network Models*. *arXiv preprint arXiv:1711.00037*.

[3] Desislava Bankova, Bob Coecke, Martha Lewis & Daniel Marsden (2016): *Graded entailment for compositional distributional semantics*. *arXiv preprint arXiv:1601.04908*.

[4] Marco Baroni & Roberto Zamparelli (2010): *Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space*. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 1183–1193.

[5] Joe Bolt, Bob Coecke, Fabrizio Genovese, Martha Lewis, Dan Marsden & Robin Piedeleu (2017): *Interacting Conceptual Spaces I: Grammatical Composition of Concepts*. *arXiv preprint arXiv:1703.08314*.

[6] Bob Coecke, Edward Grefenstette & Mehrnoosh Sadrzadeh (2013): *Lambek vs. Lambek: Functorial vector space semantics and string diagrams for Lambek calculus*. *Ann. Pure Appl. Logic* 164(11), pp. 1079–1100, doi:10.1016/j.apal.2013.05.009.

[7] Bob Coecke, Mehrnoosh Sadrzadeh & Stephen Clark (2010): *Mathematical foundations for a compositional distributional model of meaning*. *arXiv preprint arXiv:1003.4394*.

[8] Jianfeng Gao, Xiaodong He, Wen-tau Yih & Li Deng (2014): *Learning continuous phrase representations for translation modeling*. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1, pp. 699–709, doi:10.3115/v1/P14-1066.

[9] Edward Grefenstette & Mehrnoosh Sadrzadeh (2011): *Experimental support for a categorical compositional distributional model of meaning*. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 1394–1404.

[10] Edward Grefenstette & Mehrnoosh Sadrzadeh (2011): *Experimental support for a categorical compositional distributional model of meaning*. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 1394–1404.

[11] Jules Hedges & Mehrnoosh Sadrzadeh (2016): *A generalised quantifier theory of natural language in categorical compositional distributional semantics with bialgebras*. *arXiv preprint arXiv:1602.01635*.

[12] Peter Johnstone (2002): *Sketches of an Elephant Vol. 1*. Oxford Science Publications.

[13] Armand Joulin, Piotr Bojanowski, Tomas Mikolov & Edouard Grave (2018): *Improving Supervised Bilingual Mapping of Word Embeddings*. *arXiv preprint arXiv:1804.07745*.

[14] Dimitri Kartsaklis & Mehrnoosh Sadrzadeh (2013): *Prior disambiguation of word tensors for constructing sentence vectors*. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1590–1601.

[15] Dimitri Kartsaklis, Mehrnoosh Sadrzadeh & Stephen Pulman (2012): *A Unified Sentence Space for Categorical Distributional-Compositional Semantics: Theory and Experiments*. In: *Proceedings of COLING 2012: Posters*, pp. 549–558.

[16] Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, Stephen Pulman & Bob Coecke (2013): *Reasoning about meaning in natural language with compact closed categories and Frobenius algebras*. *Logic and Algebraic Structures in Quantum Computing*, p. 199, doi:10.1017/CBO9781139519687.011.

[17] Anthony S Kroch (1989): *Reflexes of grammar in patterns of language change*. *Language variation and change* 1(3), pp. 199–244, doi:10.1017/S0954394500000168.

[18] Joachim Lambek (2001): *Type grammars as pregroups*. *Grammars* 4(1), pp. 21–39, doi:10.1023/A:1011444711686.

[19] Kevin Lund & Curt Burgess (1996): *Producing high-dimensional semantic spaces from lexical co-occurrence*. *Behavior research methods, instruments, & computers* 28(2), pp. 203–208, doi:10.3758/BF03204766.

[20] Jean Maillard, Stephen Clark & Edward Grefenstette (2014): *A type-driven tensor-based semantics for CCG*. In: *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pp. 46–54.

[21] Tomas Mikolov, Quoc V Le & Ilya Sutskever (2013): *Exploiting similarities among languages for machine translation*. *arXiv preprint arXiv:1309.4168*.

[22] Robin Piedeleu, Dimitri Kartsaklis, Bob Coecke & Mehrnoosh Sadrzadeh (2015): *Open system categorical quantum semantics in natural language processing*. *arXiv preprint arXiv:1502.00831*, doi:10.4230/LIPIcs.CALCO.2015.270.

[23] Anne Preller (2013): *From Logical to Distributional Models*. In: *Proceedings of the 10th International Workshop on Quantum Physics and Logic, QPL 2013, Castelldefels (Barcelona), Spain, July 17-19, 2013.*, pp. 113–131, doi:10.4204/EPTCS.171.11.

[24] Anne Preller & Joachim Lambek (2007): *Free compact 2-categories*. *Mathematical Structures in Computer Science* 17(2), pp. 309–340, doi:10.1017/S0960129506005901.

[25] Mehrnoosh Sadrzadeh, Dimitri Kartsaklis & Esma Balkir (2018): *Sentence entailment in compositional distributional semantics*. *Ann. Math. Artif. Intell.* 82(4), pp. 189–218, doi:10.1007/s10472-017-9570-x.

[26] Luc Steels (2003): *Evolving grounded communication for robots*. *Trends in cognitive sciences* 7(7), pp. 308–312, doi:10.1016/S1364-6613(03)00129-3.

[27] Remi van Trijp (2013): *Linguistic Assessment Criteria for Explaining Language Change: A Case Study on Syncretism in German Definite Articles*. *Language Dynamics and Change* 3(1), pp. 105–132.