The structure of meaning in language: parallel narratives in linear algebra and category theory

Tai-Danae Bradley * Juan Luis Gastaldi † John Terilla ‡

August 5, 2023

Introduction

Categories for AI, an online program about category theory in machine learning, unfolded over several months beginning in the Fall of last year. The "Cats for AI" organizing committee, which included several researchers from industry including two from DeepMind, felt that the machine learning community ought to be using more rigorous compositional language and that category theory has "great potential to be a cohesive force" in science in general and in Artificial Intelligence in particular. While this article is by no means a comprehensive report on that event, the popularity of "Cats for AI" — the five introductory lectures on `cats.for.ai` have been viewed thousands of times — signals the growing prevalence of category theoretic tools in AI.

One place where category theory is gaining traction in machine learning is by providing a formal way to discuss how learning systems can be put together. This article has a different and somewhat narrow focus. It's about how a fundamental piece of AI technology used in language modeling can be understood, with the aid of categorical thinking, as a process that extracts structural features of language from purely syntactical input. The idea that structure arises from form may not be a surprise for many readers — category theoretic ideas have been a major influence in pure mathematics for three-generations — but there are consequences for linguistics that are relevant for some of the ongoing debates about artificial intelligence. We include a section that argues that the mathematics in these pages rebut some widely accepted ideas in contemporary linguistic thought and support a return to a structuralist approach to language.

The article begins with a fairly pedantic review of linear algebra which sets up a striking parallel with the relevant category theory. The linear algebra is then used to review how to understand word embeddings, which are at the root of large language models. When the linear algebra is replaced, Mad Libs style, with the relevant category theory, the output becomes not word embeddings but a lattice of formal concepts. The category theory that gives rise to the concept lattice is a particularly simplified piece of enriched category theory and suggests that by simplifying a little less, even more of the structure of language could be revealed.

Objects versus Functions on Objects

When considering a mathematical object X that has little or incomplete structure, one can replace X by something like "functions on X" which will have considerably more structure than X. Usually, there is a natural embedding $X \to Fun(X)$ so that when working with $Fun(X)$, one is working with all of X and more.

*Tai-Danae Bradley is a research mathematician at SandboxAQ and a visiting faculty member at The Master's University. Her email address is tai.danae@math3ma.com.

†Juan Luis Gastaldi is a researcher at ETH Zürich. His email address is juan.luis.gastaldi@gess.ethz.ch.

‡John Terilla is a professor of mathematics at CUNY Queens College and Doctoral Faculty at the CUNY Graduate Center. His email address is jterilla@gc.cuny.edu.

The first example that comes to mind is when the functions on a set X are valued in a field k, which forms a vector space. The embedding $X \to k^X$ is defined by sending $x \in X$ to the indicator function of x defined by $y \mapsto \delta_{xy}$. When X is finite, there is a natural inner product on k^X defined by $\langle f|g\rangle = \sum_{x\in X} f(x)g(x)$ making k^X into a Hilbert space. The physics "ket" notation $|x\rangle$ for the indicator function of $x \in X$ nicely distinguishes the element $x \in X$ from the vector $|x\rangle \in k^X$ and reminds us that there is an inner product. The image of X in k^X defines an orthonormal spanning set and if the elements X are ordered, then the vectors $\{|x\rangle : x \in X\}$ become an ordered basis for k^X and each basis vector $|x\rangle$ has a *one-hot coordinate vector*: a column vector consisting of all zeroes except for a single 1 in the x entry. This, by the way, is the starting point of quantum information theory. Classical bits $\{0, 1\}$ are replaced by quantum bits $\{|0\rangle, |1\rangle\}$ which comprise an orthonormal basis of a two dimensional complex Hilbert space $\mathbb{C}^{\{0,1\}}$. There might not be a way to add elements in the set X, or average two of them for example, but those operations can certainly be performed in k^X. In coordinates, for instance, if $x \neq y$, then the sum $|x\rangle + |y\rangle$ will have all zeroes with 1s in both the x- and y- entries and the sum $|x\rangle + |x\rangle$ has all zeroes and a 2 in the x-entry.

When the ground field is the field with two elements $k = \{0, 1\}$, the vector space structure seems a little weak. Scalar multiplication is trivial, but there are other notable structures on $\{0, 1\}^X$. Elements of $\{0, 1\}^X$ can be thought of as subsets of X, the correspondence being between characteristic functions and the sets on which they are supported. So $\{0, 1\}^X$ has all the structure of a Boolean algebra: the join $v \vee w$ and the meet $v \wedge w$ of two vectors correspond to the union and intersection of the two subsets defined by v and w, and neither the meet nor the join coincide with vector addition. Every vector has a "complement" defined by interchanging $0 \leftrightarrow 1$, the vectors in $\{0, 1\}^X$ are partially ordered by containment, every vector has a cardinality defined by the number of nonzero entries, and so on.

Another closely related example comes from category theory. By replacing a category C by $\mathsf{Set}^{\mathsf{C}^{\mathrm{op}}}$, the set-valued functors on C, one obtains a category with significantly more structure. Here the "op" indicates the variance of the functors in question (contravariant, in this case), a technical point that isn't very important here, but is included for accuracy. It's common to call a functor F in $\mathsf{Set}^{\mathsf{C}^{\mathrm{op}}}$ a *presheaf* on C. The Yoneda lemma provides an embedding $\mathsf{C} \to \mathsf{Set}^{\mathsf{C}^{\mathrm{op}}}$ of the original category as a full subcategory of $\mathsf{Set}^{\mathsf{C}^{\mathrm{op}}}$. Given an object x in C, Grothendieck used h^x to denote a *representable presheaf* $h^x := \mathsf{C}(-, x)$ which is defined by mapping an object y to the set $\mathsf{C}(y, x)$ of morphisms from y to the object x represting the presheaf. In this notation, the Yoneda embedding is defined on objects by $x \mapsto h^x$. And just as the vector space k^X has more structure than X, the category $\mathsf{Set}^{\mathsf{C}^{\mathrm{op}}}$ has more structure than C. For any category C, the category $\mathsf{Set}^{\mathsf{C}^{\mathrm{op}}}$ of presheaves is complete and cocomplete, meaning that the categorical limits and colimits of small diagrams exist in the category, and more.[1] It is also an example of what's called a *topos* which is a natural place in which to do geometry and logic.

As an illustration, consider a finite set X, which can be viewed as a discrete category X, that is, a category whose only morphisms are identity morphisms. In this case $\mathsf{X} = \mathsf{X}^{\mathrm{op}}$, and a presheaf F on X assigns a set to every object x in X. If the elements of X are ordered, then F can be thought of as a column vector whose entries are sets, with the set $F(x)$ in the x-entry. The representable functor h^x can be thought of as a column vector whose entries are all the empty set except for a one-point set $*$ in the x-entry. If we use 0 for $\emptyset$ and 1 for $*$, then these are the same arrays as the one-hot basis vectors that span the vector space k^X. Notably, the categorical coproduct $x \coprod y$ does not exist in the category X, but the coproduct $h^x \coprod h^y$ of the representable functors h^x and h^y *does* exist in the category of presheaves on X. If $x \neq y$ then $h^x \coprod h^y$ is a column consisting of empty sets except for a one-point set $*$ in the x- and y-entries; the coproduct $h^x \coprod h^x$ consists of all empty sets except for a two point set $* \sqcup *$ in the x-entry. And just as the indicator functions form a basis of the vector space k^X, every functor $\mathsf{X}^{\mathrm{op}} \to \mathsf{Set}$ is constructed

[1] Like "op," the word "small" is a technicality that isn't so important here but is included for accuracy.

from representable functors. When X is a finite set, every vector in k^X is a linear combination of basis vectors, and analogously every presheaf in $\mathsf{Set}^{\mathsf{X}^{\mathrm{op}}}$ is a colimit of representables.

In this article, it will be helpful to consider *enriched category theory*, which is the appropriate version of category theory to work with when the set $\mathsf{C}(y, x)$ of morphisms between two objects is no longer a *set*. That is, it may be a partially-ordered set, or a group, or a topological space, or something else. So, enriched category theory amounts to replacing Set with a different category. This is analogous to changing the base field of the vector space k^X, and if the new base category has sufficiently nice structure, then most everything said about replacing C by $\mathsf{Set}^{\mathsf{C}^{\mathrm{op}}}$ goes over nearly word-for-word. For example, replacing Set by the category $\mathsf{2}$, which is a category with two objects 0 and 1 and one non-identity morphism $0 \to 1$, results in the category $\mathsf{2}^{\mathsf{C}^{\mathrm{op}}}$ of $\mathsf{2}$-valued presheaves on C. In the case when C is a set X viewed as a discrete category $\mathsf{X} = \mathsf{X}^{\mathrm{op}}$, the presheaves in $\mathsf{2}^{\mathsf{X}}$ are exactly the same as $\{0, 1\}$-valued functions on X, which are the same as subsets of X. The structure on $\mathsf{2}^{\mathsf{X}}$ afforded by it being a category of $\mathsf{2}$-enriched presheaves is the Boolean algebra structure on the subsets of X previously described. The categorical coproduct of $\mathsf{2}$-enriched presheaves f and g is the join (union) $f \vee g$, and the categorical product is the meet (intersection) $f \wedge g$. So for any set X, the set of functions $\{0, 1\}^X$ can either be viewed as a vector space over $F_2 = \{0, 1\}$, the field with two elements, or as enriched presheaves on X valued in $\mathsf{2} = \{0, 1\}$, depending on whether we think of $\{0, 1\}$ as a field, or as a category $0 \to 1$, and we get different structures depending on which point of view is taken.

Now, before going further, notice that replacing an object by a free construction on that object can't immediately reveal much about the underlying object. Whether it's the free vector space on a finite set X resulting in k^X or the free cocompletion of a category C resulting in $\mathsf{Set}^{\mathsf{C}^{\mathrm{op}}}$, the structures one obtains are *free* and employ the underlying object as little more than an indexing set. The structures on the "functions" on X are owed, essentially, to the structure of what the functions are valued *in*. For example, the source of the completeness and cocompleteness of $\mathsf{Set}^{\mathsf{C}^{\mathrm{op}}}$ is the completeness and cocompleteness of the category of sets. Similarly, vector addition and scalar multiplication in k^X arise from addition and multiplication in the field k. The point is that passing to a free construction on X provides some extra room in which to investigate X, and in the theory of vector spaces, for instance, things become interesting when linear transformations are involved. As another example, passing from a finite group G to the free vector space $\mathbb{C}^G$ doesn't tell you much about the group, until that is, you involve the elements of the group as operators $\mathbb{C}^G \to \mathbb{C}^G$. The result is the regular representation for G, which among its many beautiful properties, decomposes into the direct sum of irreducible representations with every irreducible representation of G included as a term with meaningful multiplicity.

This brings us back to the strategy suggested in the first line of this section. When only a little bit is known about the internal structure of an object X, an approach to learn more is to replace X by something like functions on X and study how the limited known structure of X interacts with the freely-defined structures on the functions of X. A choice is required of what, specifically, to value the functions in and how mathematically that target is viewed. The remaining sections of this article can be interpreted simply as working through the details in an example with linguistic importance for a couple of natural choices of what to value the functions in.

Embeddings in Natural Language Processing

In the last decade, researchers in the field of computational linguistics and natural language processing (NLP) have taken the step of replacing words, which at the beginning only have the structure of a set, ordered alphabetically, by vectors. One gets the feeling that there is structure in words — words appear to be used in language with purpose and meaning; dictionaries relate each word to other words; words can be labelled with parts of speech; and so on — though the precise mathematical nature of the structure of words and their usage is not clear. Language would

appear to represent a significant real world test of the strategy to uncover structure described in the previous section. While the step of replacing words by vectors constituted one of the main drivers of current advances in the field of artificial intelligence, it is not readily recognizable as an instance of replacing a set by functions on that set. This is because replacing words by vectors is typically performed implicitly by NLP tools, which are mathematically obscured by their history — a history which we now briefly review.

Following the surprisingly good results obtained in domains such as image and sound processing, researchers working to process natural language in the 2010s became interested in the application of deep neural network (DNN) models. As a reminder, in its most elementary form, a DNN can be described as a function $f : \mathbb{R}^n \to \mathbb{R}^m$ explicitly expressed as a composition:

$$f\colon \mathbb{R}^n \xrightarrow{f_1} \mathbb{R}^{n_1} \xrightarrow{f_2} \mathbb{R}^{n_2} \xrightarrow{f_3} \cdots \cdots \xrightarrow{f_K} \mathbb{R}^{n_K} \xrightarrow{g} \mathbb{R}^m \quad (1)$$

$$f_i(\mathbf{x}) = a(\mathbf{M}_i\mathbf{x} + b_i) \quad (2)$$

where the $\mathbf{M}_i$ are $n_i \times n_{i-1}$ matrices, the $b_i \in \mathbb{R}^{n_i}$ are "biases", the a is a (non-linear) "activation" function, and g is an output function. After fixing the activation and output functions, a DNN lives in a moduli space of functions parametrized by the entries of the matrices M_i and the bias vectors b_i. Training a DNN is the process of searching through the moduli space for suitable M_i and b_i to find a function that performs a desired task, usually by minimizing a cost function defined on the moduli space. In the particular case of natural language tasks, this typically requires feeding linguistic data into the model and setting the optimization objective to the minimization of the error between the actual and intended outputs, with the optimization performed by a form of gradient descent.

Significantly, in this setting, linguistic data (typically words) would be represented as vectors—the domain of a DNN is a vector space. A natural first choice for practitioners was then to represent words as one-hot vectors. Thus, if one has a vocabulary D consisting of, say, $30,000$ words, then $D \to \mathbb{R}^D$ embeds words as the standard basis vectors in a $30,000$ dimensional real vector space.

$$\begin{aligned} \texttt{aardvark} &\mapsto (1,0,0,0,\ldots,0,0) \\ \texttt{aardwolf} &\mapsto (0,1,0,0,\ldots,0,0) \\ &\vdots \\ \texttt{zyzzyva} &\mapsto (0,0,0,0,\ldots,0,1). \end{aligned}$$

Likewise, if the output is to be decoded as a word, then the output function g is to be interpreted as a probability distribution over the target vocabulary, which is usually the same $\mathbb{R}^D$, though it could certainly be different in applications like translation.

At the time, a DNN was thought of as an end-to-end process: whatever happens between the input and the output was treated as a black-box and left for the optimization algorithm to handle. However, a surprising circumstance arose. If the first layer (namely, f_1 in Equation (1)) of a model trained for a given linguistic task was used as the first layer of another DNN aimed at a different linguistic task, there would be a significant increase in performance of the second task. It was not long thereafter that researchers began to train that single layer independently as the unique hidden layer of a model that predicts a word given other words in the context. Denoting that single layer σ, one can then obtain

$$D \hookrightarrow \mathbb{R}^D \xrightarrow{\sigma} \mathbb{R}^{n_1} \quad (3)$$

which embeds D in a vector space of much lower dimension, typically two or three hundred. Take, for instance, the vector representations made available by [ea], where the word `aardvark` is mapped to a vector with 200 components:

$$\texttt{aardvark} \mapsto (0.632, 0.370, -0.620, \ldots, -0.475).$$

In this way, the images of the initial one-hot basis vectors under the map σ could be used as low-dimensional dense word vector representations to be fed as inputs across multiple DNN models. The word "dense" here is not a technical term but is used in contrast to the original one-hot word vectors in $\mathbb{R}^D$, which are "sparse" in the sense that all entries were

0 except for a single 1. On the other hand, the word vectors in $\mathbb{R}^{n_1}$ generally have nonzero entries.

The set of word vector representations produced in this way — also known as "word embeddings" — were found to have some surprising capabilities. Not only did the performance of models across different tasks increase substantially, but also unexpected linguistic significance was found in the vector space operations of the embedded word vectors. In particular, the inner product between two vectors shows a high correlation with semantic similarity. As an example, the vector representations for `tubulidentata`, `orycteropus`, `anteaters`, `shrews`, and `pangolins` are among the ones with the largest inner product with that of `aardvark`. Even more surprisingly, addition and subtraction of vectors in the embedding space correlate with analogical relations between the words they represent. For instance, the vector for `Berlin` minus the vector for `Germany` is numerically very near the vector for `Paris` minus the vector for `France` [MSC+13], all of which suggests that the word vectors live in something like a *space of meanings* into which individual words embed as points. Subtracting the vector for `Germany` from the vector for `Berlin` does not result in a vector that corresponds to any dictionary word. Rather, the difference of these vectors is more like the concept of a "capital city", not to be confused with the vector for the word `capital`, which is located elsewhere in the meaning space.

Word vector representations as the one described are now the standard input to current neural linguistic models, including Large Language Models (LLMs) which are currently the object of so much attention. And the fact, as suggested by these findings, that semantic properties can be extracted from the formal manipulation of pure syntactic properties — that meaning can emerge from pure form — is undoubtedly one of the most stimulating ideas of our time. We will later show that such an idea is not new but has, in fact, been present in linguistic thought for at least a century.

But first it is important to understand why word embeddings illustrate the utility of passing from X to $Fun(X)$ introduced in the previous section. The semantic properties of word embeddings are not present in the one-hot vectors embedded in $\mathbb{R}^D$. Indeed, the inner product of the one-hot vectors corresponding to `aardvark` and `tubulidentata` is zero, as it is for any two orthogonal vectors, and the vector space operations in $\mathbb{R}^D$ are not linguistically meaningful. The difference between the one-hot vectors for `Germany` and `Berlin` is as far away from the difference between the one-hot vectors for `France` and `Paris` as it is from from the difference between the one-hot vectors for `salami` and `therefore` or any other two one-hot vectors. Linguistically significant properties emerge only after composing with the embedding map $\sigma\colon \mathbb{R}^D \to \mathbb{R}^{n_1}$ in (3) that was obtained through neural optimization algorithms, which are typically difficult to interpret.

In the specific case of word embeddings, however, the algorithm has been scrutinized and shown to be performing an implicit factorization of a matrix comprised of information about how words are used in language. To elaborate, the optimization objective can be shown to be equivalent to factorizing a $|D| \times |D|$ matrix M, where the i-j-entry is a linguistically relevant measure of the term-context association between words w_i and w_j. That measure is based on the pointwise mutual information between both words, which captures the probability that they appear near each other in a textual corpus [LG14]. The map σ is then an optimal low-rank approximation of M. That is, one finds σ' and σ of sizes $|D| \times d$ and $d \times |D|$ respectively, with $d \ll |D|$, such that $\|M - \sigma'\sigma\|$ is mimimal. The upshot is that neural embeddings are just low-dimensional approximations to the columns of M. Therefore, *the surprising properties exhibited by embeddings are less the consequence of some magical attribute of neural models than the algebraic structure underlying linguistic data found in corpora of text.* Indeed, it has since been shown that one can obtain results comparable to those of neural word embeddings by directly using the columns of M, or a low-dimensional approximation thereof, as explicit word-vector representations [LGD15]. Interestingly, the other factor σ', which is readable as the second layer of the trained DNN is typically discarded although it does contain relevant linguistic information.

In summary, the math story of word embeddings

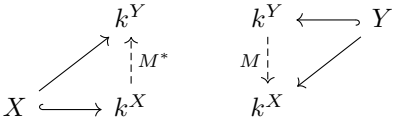
goes like this: first, pass from the set D of vocabulary words to the free vector space $\mathbb{R}^D$. While there is no meaningful linguistic information in $\mathbb{R}^D$, it provides a large, structured setting in which a limited amount of information about the structure of D can be placed. Specifically, this limited information is a $|D| \times |D|$ matrix M consisting of rough statistical data about how words go with other words in a corpus of text. Now, the columns of M, or better yet, the columns of a low-rank factorization of M, then interact with the vector space structure to reveal otherwise hidden syntactic and semantic information in the set D of words. Although a matrix of statistical data seems more mathematically casual than, say, a matrix representing the multiplication table of a group, it has the appeal of assuming nothing about the structure that D might possess. Rather, it is purely a witness of how D has been used in a particular corpus. It's like a set of observations is the input, and a more formal structure is an output.

So, if word embeddings achieved the important step of finding a linguistically meaningful *space* in which words live, then the next step is to better understand what is the *structure* underlying that space. Post facto realizations about vector subtraction reflecting certain semantic analogies hint that even more could be discovered. For this next discussion, it is important to understand that there is an exact solution for the low-rank factorization of a matrix M using the truncated singular value decomposition (SVD), which has a beautiful analogue in category theory. To fully appreciate the analogy, it will be helpful to review matrices from an elementary perspective.

From the Space of Meanings to the Structure of Meanings

In this section, keep in mind the comparison between functions on a set X valued in a field k and functors on a category C valued in Set or another enriching category. Now, let's consider matrices.

Given finite sets X and Y, an X-Y matrix valued in a field k is a function $m\colon X \times Y \to k$. By simple currying, m defines functions $X \to k^Y$ and $Y \to k^X$ defined by $x \mapsto m(x,-)$ and $y \mapsto m(-,y)$. Ordering the elements of X and Y, the function m can be represented as a rectangular array of numbers with $|X|$ rows and $|Y|$ columns with the value $m(x,y)$ being the number in the x-th row and y-th column. The function $m(x,-) \in k^Y$ is then identified with the x-th row of the matrix, which has as many entries as the elements of Y and defines a function on Y sending y to the y-th entry in the row. Similarly, the y-th column of m represents the function $m(-,y) \in k^X$. Linearly extending the maps $X \to k^Y$ and $Y \to k^X$ produces linear maps $M^*\colon k^X \to k^Y$ and $M\colon k^Y \to k^X$, which of course are the linear maps associated with the matrix M and its transpose M^*. Here is a diagram:



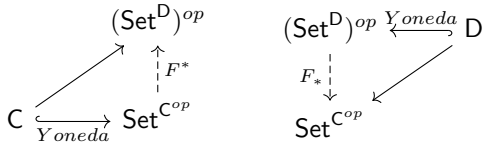
Now, the compositions $MM^*\colon k^X \to k^X$ and $M^*M\colon k^Y \to k^Y$ are linear operators with special properties. If we fix the ground field k to be the real numbers $\mathbb{R}$, we can apply the spectral theorem to obtain orthonormal bases $\{u_1, \ldots, u_m\}$ of k^X and $\{v_1, \ldots, v_n\}$ of k^Y consisting of eigenvectors of MM^* and M^*M respectively with shared non-negative real eigenvalues $\{\lambda_1, \ldots, \lambda_r, 0, \ldots, 0\}$, where $r = \min(m,n)$. This data can be refashioned into a factorization of M as $M = U\Sigma V^*$. This is the so-called singular value decomposition of M. The $\{u_i\}$ are the columns of U, the $\{v_j\}$ are the rows of V^*, and Σ is the $m \times n$ diagonal matrix whose i-th entry is $\sigma_i = \sqrt{\lambda_i}$. The matrices U and V satisfy $U^*U = I$ and $V^*V = I$. In SVD terminology, the non-negative real numbers σ_i are the *singular values* of M, and the vectors $\{u_1, \ldots, u_r\}$ and $\{v_1, \ldots, v_r\}$ are the *left and right singular vectors* of M. In other words, we have pairs of vectors $\{(u_1, v_1), \ldots, (u_r, v_r)\}$ in $k^X \times k^Y$ related to each other as

$$M^* u_j = \sigma_j v_j \text{ and } M v_j = \sigma_j u_j.$$

Moreover, these pairs are ordered with $(u_i, v_i) \leq (u_j, v_j)$ if the corresponding singular values satisfy $\sigma_i \leq \sigma_j$. Finally, it is not difficult to show that the matrix M' with rank at most s that is clos-

est in Frobenius norm to the matrix M is given by $M' = U\Sigma' V^*$ where Σ' is the $m \times n$ diagonal matrix containing only the s greatest non-zero singular values on the diagonal. By eliminating the parts of U, Σ', and V^* that do not, because of all the zeros entries in Σ', participate in the product $U\Sigma' V^*$, one obtains a factorization $M' = U'\Sigma'' V'^* \approx M$ where U' is an $m \times s$ matrix, V'^* is an $s \times n$ matrix, and Σ'' is an $s \times s$ diagonal matrix with the s largest singular values of M on the diagonal. Principal component analysis (PCA) employs this approximate factorization of a matrix into low-rank components for dimensionality reduction of high-dimensional data.

Moving from linear algebra to category theory, one finds a remarkably similar story. Given two categories C and D, the analogy of a C-D matrix is something called a *profunctor*, which is a set-valued functor $f\colon \mathsf{C}^{\mathrm{op}} \times \mathsf{D} \to \mathsf{Set}$. As before, the "op" here and in what follows is used to indicate the variance of functors and is needed for accuracy, but can on first reading be ignored. Experts will surely know of situations in which this op is involved in interesting mathematical dualities, but for the analogy with linear algebra described here, it can be thought of as indicating a sort of transpose between rows and columns, harmlessly moving information around but fixing a convention required for accurate computations. If both domain categories are finite sets viewed as discrete categories, then a profunctor is simply a collection of sets indexed by pairs of elements — that is, a matrix whose entries are sets instead of numbers. Again, by simple currying, a profunctor defines a pair of functors $\mathsf{C} \to \left(\mathsf{Set}^{\mathsf{D}}\right)^{\mathrm{op}}$ and $\mathsf{D} \to \mathsf{Set}^{\mathsf{C}^{\mathrm{op}}}$ defined on objects by $c \mapsto f(c,-)$ and $d \mapsto f(-,d)$. As in the linear algebra setting, the functor $f(c,-)$ can be pictured as the c-th row of sets in the matrix f, which defines a functor $\mathsf{D} \to \mathsf{Set}$ where the j-th object of D is mapped to the j-th set in the row $f(c,-)$. The functor $f(-,d)\colon \mathsf{C}^{\mathrm{op}} \to \mathsf{Set}$ can be similarly be pictured as the d-th column of f. Thinking of a category as embedded in its category of presheaves via the Yoneda (or co-Yoneda) embedding, the functors $\mathsf{C} \to \left(\mathsf{Set}^{\mathsf{D}}\right)^{\mathrm{op}}$ and $\mathsf{D} \to \mathsf{Set}^{\mathsf{C}^{\mathrm{op}}}$ can be extended in a unique way to functors $F^*\colon \mathsf{Set}^{\mathsf{C}^{\mathrm{op}}} \to (\mathsf{Set}^{\mathsf{D}})^{\mathrm{op}}$ and $F_*\colon (\mathsf{Set}^{\mathsf{D}})^{\mathrm{op}} \to \mathsf{Set}^{\mathsf{C}^{\mathrm{op}}}$ that preserve colimits and limits, respectively.



Now, just as the composition of a linear map M and its transpose M^* define linear maps with special properties, the functors F^* and F_* are adjoint functors with special properties. This particular adjunction $F^*\colon \mathsf{Set}^{\mathsf{C}^{\mathrm{op}}} \leftrightarrows (\mathsf{Set}^{\mathsf{D}})^{\mathrm{op}}\colon F_*$ is known as the *Isbell adjunction*, which John Baez recently called "a jewel of mathematics" in a January 2023 column article in this publication [Bae23]. Objects that are fixed up to isomorphism under the composite functors F^*F_* and F_*F^* are called the *nuclei* of the profunctor f and are analogous to the left and right singular vectors of a matrix. One can organize the nuclei into pairs (c_i, d_i) of objects in $\mathsf{C}^{\mathrm{op}} \times \mathsf{D}$ where

$$F^* c_i \cong d_i \text{ and } F_* d_i \cong c_i.$$

The nuclei themselves $\{(c_i, d_i)\}$ have significant structure — they organize into a category that is complete and cocomplete. The pairs $\{(u_i, v_i)\}$ of singular vectors of a matrix have some structure — they are ordered by the magnitude of their singular values, and the magnitudes themselves are quite important. The nuclei $\{(c_i, d_i)\}$ of a profunctor has a different, in some ways more intricate, structure because one can take categorical limits and colimits of diagrams of pairs, allowing the pairs to be combined in various algebraic ways. In the context of linguistics, this is significant because the nuclei are like symbols and the categorical limits and colimits provide ways to manipulate the symbols. This is illustrated concretely in the next section. For now, think word embeddings obtained from the singular vectors of a matrix can be used to overlay the structure of a vector space on meanings. For certain semantic aspects of language, like semantic similarity, a vector space structure is a good fit, but it veils others. The Isbell adjunction can help illuminate other, different structural features of language.

For flexibility, it is useful to look at the Isbell adjunction in the enriched setting. If the base category

is $\mathsf{2}$ instead of Set, then a profunctor r between two finite sets X and Y, viewed as discrete categories enriched over $\mathsf{2}$, is just a function $r\colon X \times Y \to \{0,1\}$, which is the same as a relation on $X \times Y$. The functors $R^*\colon \mathsf{2}^X \to \mathsf{2}^Y$ and $R_*\colon \mathsf{2}^Y \to \mathsf{2}^X$ are known objects in the theory of formal concept analysis [GW99]. The function R^* maps a subset $A \subseteq X$ to the set $R^*(A) = \{y \in Y : R(x,y) = 1 \text{ for all } x \in A\}$ and R_* maps a subset $B \subseteq Y$ to the set $R_*(B) = \{x \in X : R(x,y) = 1 \text{ for all } y \in B\}$. The fixed objects of R^*R_* and R_*R^* are known as *formal concepts*. They are organized into pairs $(A_i, B_i) \subset X \times Y$ with

$$R^*(A_i) = B_i \text{ and } R_*(B_i) = A_i$$

and the set of all formal concepts $\{(A_i, B_i)\}$ is partially ordered with $(A_i, B_i) \leq (A_j, B_j)$ if and only if $A_i \subseteq A_j$ which is equivalent to $B_i \supseteq B_j$. Moreover, $\{(A_i, B_i)\}$ forms a complete lattice, so, like the singular vectors of a matrix, there is a least and a greatest formal concept, and more. The product and coproduct, for example, of formal concepts are defined by $(A_i, B_j) \wedge (A_j, B_j) := (A_i \cap A_j, R^*R_*(B_i \cup B_j))$ and $(A_i, B_j) \vee (A_j, B_j) := (R_*R^*(A_i \cup A_j), B_i \cap B_j)$. The point is that limits and colimits of formal concepts have simple, finite formulas that are similar to, but not exactly, the union and intersection of sets and give an idea of the kind of algebraic structures one would see on the nucleus of a profunctor.

Structures in the Real World

If there's any place where neural techniques have indisputably surpassed more principled approaches to language, it is their capacity to exhibit surprisingly high performance on empirical linguistic data. Whatever the nature of formal language models to come, it will certainly be decisive to judge their quality and relevance in the real world. In this section, we illustrate how the tools of linear algebra and enriched category theory work in practice, and in the conclusion we will share how the empirical capabilities of the enriched category theory presented here can be used to do more.

To start, consider the English Wikipedia corpus comprising all Wikipedia articles in English as of March 2022 [Wik]. If we consider this corpus as a purely syntactic object without assuming any linguistic structure, then the corpus appears as a long sequence of a finite set of independent tokens or characters. To simplify things, let's restrict ourselves to the 40 most frequent characters in that corpus (excluding punctuation), which account for more than 99.7% of occurrences. So, our initial set X contains the following elements:

$$X = \{\texttt{-}, \texttt{/}, \texttt{0}, \texttt{1}, \texttt{2}, \texttt{3}, \texttt{4}, \texttt{5}, \texttt{6}, \texttt{7}, \texttt{8}, \texttt{9}, \texttt{=}, \texttt{a}, \texttt{b}, \texttt{c}, \texttt{d}, \texttt{e}, \texttt{f}, \texttt{g}, \texttt{h}, \texttt{i}, \texttt{j}, \texttt{k}, \texttt{l}, \texttt{m}, \texttt{n}, \texttt{o}, \texttt{p}, \texttt{q}, \texttt{r}, \texttt{s}, \texttt{t}, \texttt{u}, \texttt{v}, \texttt{w}, \texttt{x}, \texttt{y}, \texttt{z}, \texttt{é}\}. \tag{4}$$

Now let $Y = X \times X$ and, in line with what we have seen in the previous sections, consider a matrix $m\colon X \times Y \to \mathbb{R}$ representing some linguistically relevant measure of the association between the elements of X and Y. A straightforward choice for m is the empirical probability that the characters $(y_l, y_r) \in Y$ are the left and right contexts of the character $x \in X$. For instance, we have that $m(\texttt{h}, (\texttt{t}, \texttt{e})) \approx 0.3836$ while $m(\texttt{h}, (\texttt{p}, \texttt{o})) \approx 0.0037$ reflecting that it is over a hundred times more probable to see the sequence `the` than `pho`, given `h` as the center character.

Considered as elements of X, each character is independent and as different as it can be from all the others. However, embedding them $X \to \mathbb{R}^Y$ via the matrix m and leveraging the relationships they exhibit in concrete linguistic practices as reflected by a corpus brings out revealing structural features. Indeed, if we perform an SVD on the induced operator $M^*\colon \mathbb{R}^X \to \mathbb{R}^Y$ we can obtain a vector representation of each character.[2] Figure 1 shows a plot of all characters in X as points in a 3-dimensional space, where the coordinates are given by the singular vectors corresponding to the three largest singular values, scaled by those singular values. We can see how what were originally unrelated elements now appear organized into clusters in the embedding space with identifiable linguistic significance. Namely, the elements are distinguished as vowels, consonants, and digits.

[2] For reasons beyond the scope of this paper, it's convenient to take the square roots of the entries and center the matrix around 0 before performing the SVD.

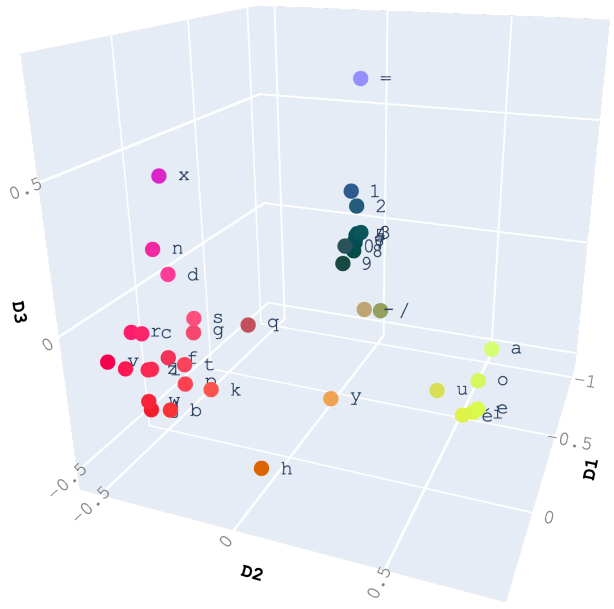


Figure 1: Characters of the Wikipedia corpus as 3-dimensional embeddings obtain through SVD.

What's more, the dimensions of the embedding space defined by the singular vectors of M^* have a little bit of natural structure — there is a canonical order given by the singular values that is endowed with linguistic significance. Looking at the first three singular vectors, we see that the first one discriminates between digits and letters, the second one distinguishes vowels from the rest, and the third one identifies special characters (see Figure 2). The rapid decay of the successive singular values indicates that dimensions beyond three adds only marginal further distinctions.

While the decomposition into singular values and vectors reveals important structural features — each singular vector discriminated between elements in a reasonable way, and the corresponding singular values worked to cluster elements into distinct types — the linear algebra in this narrow context seems to run aground. However, as discussed in the previous sections, we can gain further and different structural insights by considering our sets X and Y as categories or enriched categories. As a primitive illustration, view X and Y as discrete categories enriched over 2. For the next step, a $\{0,1\}$-valued matrix $r\colon X \times Y \to \{0,1\}$ is required. A simple and rather unsophisticated choice is to establish a cutoff value (such as 0.001) to change the same matrix M used in the SVD above into a $\{0,1\}$-valued matrix. All the entries of M less than the cutoff are replaced with 0, and all the entries above the cutoff are replaced with 1.

Then, extend to obtain functors $R^*\colon 2^X \to 2^Y$ and $R_*\colon 2^Y \to 2^X$ and look at the fixed objects of R_*R^* and R^*R_*, which are organized as described earlier into formal concepts $\{(A_i, B_i)\}$ that form a highly structured and complete lattice. Visualizing the lattice in its entirety is challenging in a static two-dimensional image. To give the idea in these pages, one can look at a sublattice defined by selecting a single character and looking at only those concepts $\{(A, B)\}$ for which A contains the selected character. For Figure 4, the characters `a` and `3` are selected and to further reduce the complexity of the images, only nodes representing a large number of contexts, $|B| \geq 20$, are drawn.

Right away the lattices make clear the distinction between digits and letters, but there is also more. Each set of characters A_i is associated to an explicit dual set of contexts B_i suggesting a principle of compositionality — namely, the elements of the corresponding classes can be freely composed to produce a sequence belonging to the corpus. Such composition reveals relevant features from a linguistic viewpoint. While digits tend to compose with other digits or special characters, vowels compose mostly with consonants. Strictly speaking, a similar duality was present in the SVD analysis, since left singular vectors are canonically paired (generically) in a one-to-one fashion with right singular vectors, which discriminate between contexts in Y based on the characters for which they are contexts. The formal concepts, on the other hand, as fixed objects of R_*R^* and R^*R_*, display dualities between large but discrete classes of characters. Numbers, consonants, and letters are all distinguished, but finer distinctions are also made. Moreover, the collection of such dual classes is not just a set of independent elements

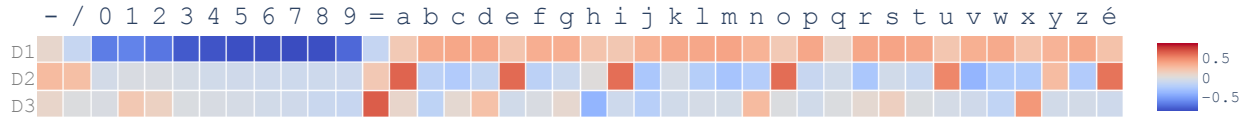


Figure 2: Top three left singular vectors of the X-Y matrix of characters in the Wikipedia corpus, scaled by the corresponding singular values.

but carries the aforementioned operations $\vee$ and $\wedge$ allowing one to perform algebraic operations on the concept level.

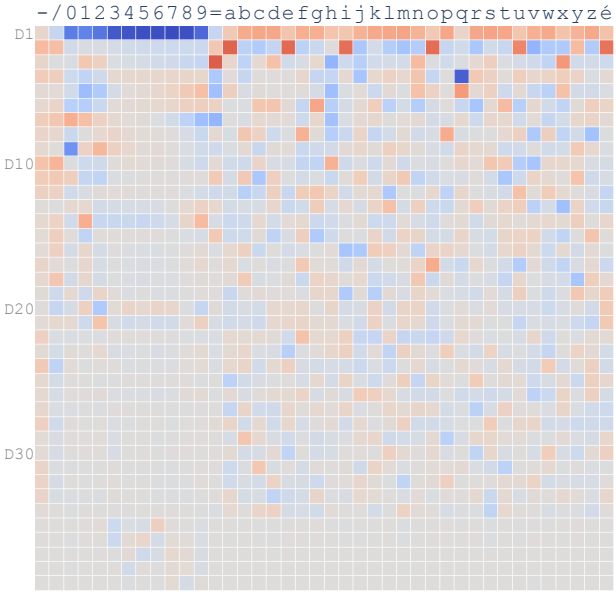


Figure 3: Left singular vectors of the X-Y matrix of characters in the Wikipedia corpus, scaled by their corresponding singular values.

Is it Really Meaning? Content from Form in Linguistic Thought

Upon reflection, it may not be surprising that syntactic features of language can be extracted from a corpus of text, since a corpus — as a sequence of characters — is a syntactic object itself. After all, from a linguistic viewpoint, consonants, vowels, and digits are purely syntactic units devoid of any meaning *per se*. What is true for characters, however, is also true for linguistic units of higher levels. This can be illustrated by letting the set X be the 1000 most frequent words in the British National Corpus [BNC07]. Use the empirical probabilities that the words y_l, y_r are the left and right contexts of word x in that corpus to make an X-Y-matrix M and repeat the same calculations done before. The singular vectors of M corresponding to the ten largest singular values capture all manner of syntactic and semantic features of words, such as nouns, verbs (past and present), adjectives, adverbs, places, quantifiers, numbers, countries, and so on. These ten singular vectors are pictured in descending order in Figure 5 where, for readability, only eight of the 1000 entries are displayed, namely, the four greatest and four least.

Further information about the terms appearing in the singular vectors can be obtained by choosing a cutoff (here, 0.01) to create a Boolean matrix M, just as was done for the character matrix, and a lattice of formal concepts can be extracted for these 1000 words. Figure 6 depicts sublattices for a few words (`France`, `could`, `10`) selected from the entries of the most significant singular vectors pictured in Figure 5. The linear algebra highlights these words as significant and goes some of the way toward clustering them. Choosing a cutoff and passing to the formal concepts reveals the syntactic and semantic classes these words belong to and reveals interesting and more refined structural features.

The broader and more philosophical question remains, though. Is it really *meaning* that has been uncovered, and if so, how is it possible that important aspects of meaning emerge from pure form? In the wake of recent advances in large language mod-

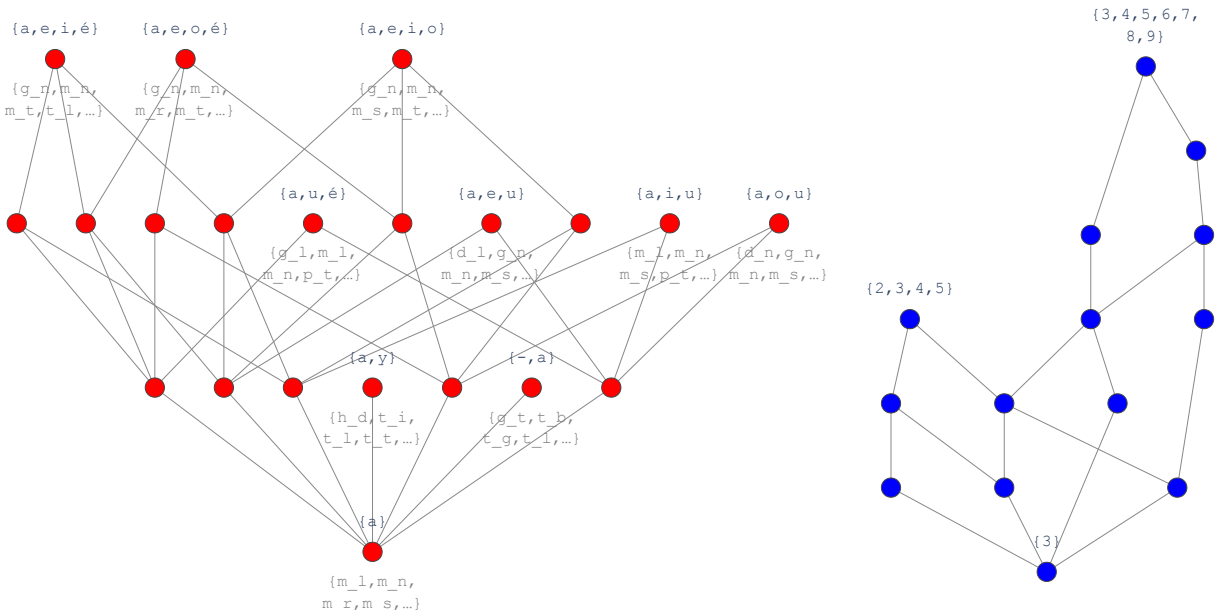


Figure 4: Sublattices of formal concepts for characters in the Wikipedia corpus for the characters `a` and `3` for which there are at least 20 contexts. Only the minimal and maximal nodes are labeled. Contexts not shown for the lattice for `3`.

els, this question has become increasingly important. One idea that's often been repeated is that language models with access to nothing but pure linguistic form (that is, raw text) do not and can not have any relation to meaning. This idea rests upon an understanding of meaning as "the relation between a linguistic form and communicative intent" [BK20, p. 5185]. Given the relevance to the current state of the art, it is worth addressing this idea before concluding this article.

Rather than asserting that formal corpus analysis bears no relation to meaning, one can instead ask the following:

> *What must the relationship between form and meaning be, given that models with access only to linguistic form are capable of identifying features and performing tasks that were generally assumed to require access to meaningful content?*

While these pages are not the place to provide a substantial philosophical treatment of this question, it seems important to point out that the mathematics in these pages supports the idea that meaning is inseparable from the multiple formal dimensions inherent in text data. Moreover, the idea that meaning and form are inseparable is not new, it just is not prevalent in the current debates. From a strictly philosophical standpoint, Kant and Hegel's influential work stood on the principle that form and content are not exclusive, and idea that one can also find at the core of Frege's thought, the father of Analytic philosophy.

Shortly after, the idea that form and meaning are not independent became central in linguistics thanks to the work of Ferdinand de Saussure [Sau59] and the structuralist revolution motivating the emergence of modern linguistics. The key argument is that both form and meaning, signifier and signified, are simultaneously determined by common *structural* features

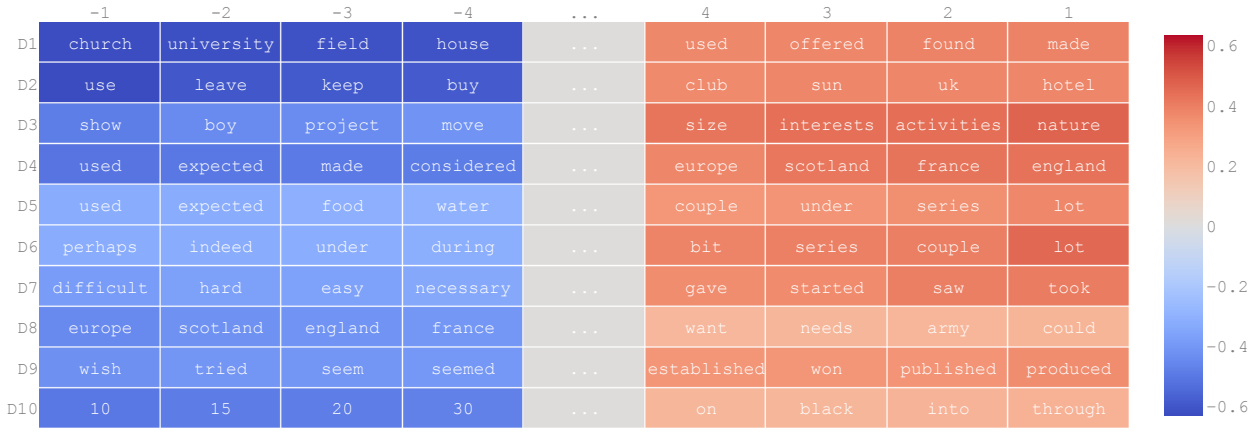


Figure 5: Words in the BNC corpus corresponding to the four greatest and four least values for the first 10 singular vectors in decreasing order.

— structural differences on one side correlate with structural differences on the other. Significantly, one of the main tools to infer that structure in the structuralist theory is the *commutation test*, which tries to establish correlations between pairs of linguistic units at different levels, thus addressing a phenomenon revealed by the mathematical approach presented here. For example, substituting "it" by "they" requires substituting "is" by "are" in the same context, while substituting "it" by "she" does not, although it might necessitate substitution in other units.

Saussure's influential view of language as form dominated linguistics for the first half of the 20th century, advanced by authors like Jakobson and Hjelmslev in Europe and developed further by Harris in America. Often cited as the theoretical foundation of current neural language models, Harris' distributionalism [Har70] maintains that in order for linguistics to be a science, it should account for phenomena through distributional properties alone—that is, how units appear together in a given linguistic corpus. Following this lead, John Firth developed a distributional semantic theory relating meaning to word distribution. As Firth famously wrote, "You shall know a word by the company it keeps!" [Fir57, p. 11]. Halfway through the 20th century, Saussure's idea that linguistic form and meaning are intimately related, like two sides of the same sheet of paper, was dominant in the field of linguistics.

While the introduction of Chomsky's novel generative linguistics in the late 1950s, brought a dramatic slowdown to the structuralist program, empirical approaches returned to linguistics toward the end of the 20th century. Connectionism, corpus linguistics, Latent Semantic Analysis, and other formal approaches to language learnability have provided a myriad of conceptual and technical means to intertwine semantics and syntax (see [CCGP15], and references within).

Although it may come as a surprise that semantics are at stake in language models with access to linguistic form only, the point of this brief review is that a theory of the emergence of meaning from form is part of an extensive and well-established tradition of linguistic thought. And what such a tradition tells us, in particular in its structuralist version, is that, if meaning is at stake in the analysis of syntactic objects, it is entirely due to structural features reflected in linguistic form.

It is at this precise point, however, where current neural language models fall short since they do not *reveal* the structural features that are necessarily at work as they perform their tasks. The mathematical discussion in this article, suggests that this is not an insurmountable issue but rather is a fascinating research subject, squarely contained in a mathematical

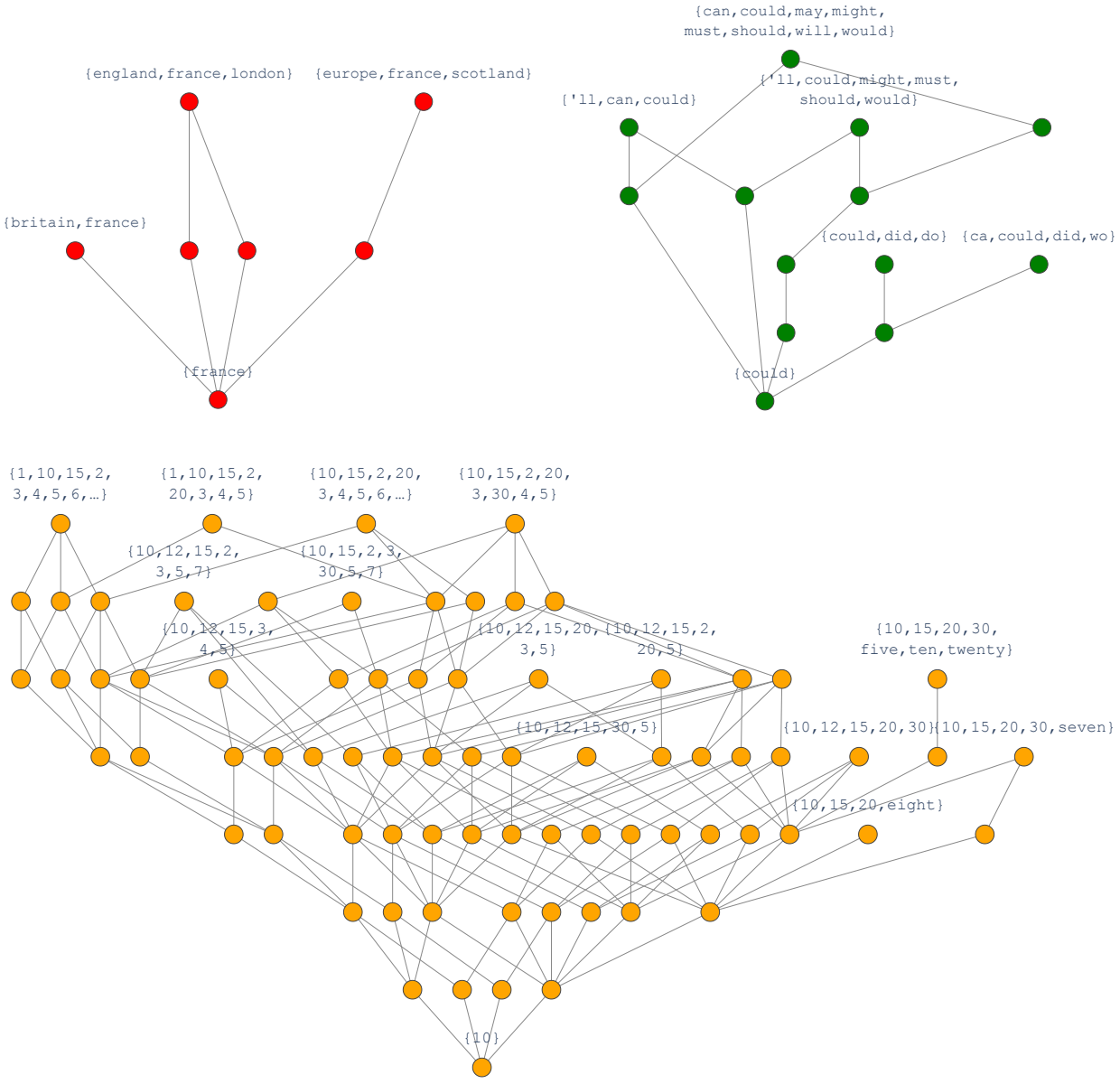


Figure 6: Sublattices of formal concepts for words in the British National Corpus for the words `France`, `could`, and `10` for which there are significant contexts (at least ten for the words `France` and `10`, and at least five for the word `could`). Only the minimal and maximal nodes are labeled. Contexts are not shown.

domain, and independent of the architectures of the language models.

Conclusion: Looking Forward

By understanding word embeddings through well-known tools in linear algebra and by framing formal concept analysis in categorical terms, one finds parallel narratives to unearth structural features of language from purely syntactical input. More specifically, using a real-valued matrix that encodes syntactical relationships found in real world data, one can use linear algebra to pass to a space of meanings that displays some semantic information and structure. By introducing cutoffs to obtain a $\{0,1\}$-valued matrix, one can use formal concept analysis to reveal semantic structures arising from syntax. While there is nothing new about the well-known tools from linear algebra and enriched category theory used in this article (principal component analysis and formal concept analysis), the parallel narratives surrounding both sets of tools is less well known. More important than communicating the narrative, however, is the possibility that the framework of enriched category theory can provide new tools, inspired by linear algebra, to improve our understanding of how semantics emerges from syntax and to study the structure of semantics.

One approach that immediately comes to mind is a way to bring the linear algebra and formal concept analysis closer together. The extended real line $[-\infty,\infty]$ or the unit interval $[0,1]$ can be given the structure of a closed symmetric monoidal category making it an appropriate base category over which other categories can be enriched. So, $[-\infty,\infty]^X$, the functions on a set X valued in the extended reals, can be viewed as a category of presheaves enriched over $[-\infty,\infty]$, much in the same way that 2^X can be viewed as a category of presheaves enriched over $\{0 \to 1\}$. Then, the matrix M or some variation of it can be regarded as a profunctor enriched over the category of extended reals. The structure of the nucleus could be studied directly in a way comparable to the formal concept analysis without introducing cutoffs to obtain a $\{0,1\}$-matrix and also would involve the order on the reals that arranges singular vectors in order of importance. This idea has been around in certain mathematical circles for about a decade. See [Pav12, Wil13, Ell17, Bra20, BTV22] and the references within. One might think of this approach as a way to unify the lattices of formal concepts for all cutoff values into one mathematical object, formal concepts intricately modulated by real numbers.

Another important point is that the linear algebra discussion began with sets X and Y having no more structure than an ordering on the elements. To keep the categorical discussion as parallel as possible, X and Y were considered discrete categories with no non-identity morphisms. However, one can introduce morphisms or enriched morphisms into X and Y and the presence of those morphisms will be carried throughout the constructions described and ultimately reflected in the nucleus of any X-Y profunctor. There is no obvious way to account for such information with existing tools in linear algebra. The recent PhD thesis [dF22] recasts a number of linguistic models of grammar—regular grammars, context-free grammars, pregroup grammars, and more—in the language of category theory, which then fits in the wider context of Coecke et. al's compositional distributional models of language [CCS10]. In these models, which date back to the 2010s, the meanings of sentences are proposed to arise from the meanings of their constituent words together with how those words are composed according to the rules of grammar. This relationship is modeled by a functor from a chosen grammar category to a category that captures distributional information, such as finite dimensional vector spaces. Such models may also be thought of as a passage from syntax to semantics, though they rely heavily on a choice of grammar. The point here, however, is that if one would like to begin with X having more structure than a set, then enriched category theory provides a way to do so without disrupting the mathematical narrative described in this article.

One place where the linear algebra tools have developed further than their analogues in enriched category theory is in multilinear algebra. For example, factorizing a tensor in the tensor product of vector spaces $V_1 \otimes V_2 \otimes \cdots \otimes V_n$ into what is called a *tensor train*, or *matrix product state*, can be interpreted as

a sequence of $n-1$ compatible truncated SVDs. We are not aware of any similar theory of "sequences of compatible nuclei" for a functor on product of categories $\mathsf{C}_1 \times \mathsf{C}_2 \times \cdots \times \mathsf{C}_n$. Given that text data is more naturally regarded as a long sequence of characters than mere term-context pairs, it is reasonable to think an enriched categorical version of such an object could be the dominant way to understand how semantic structures emerge from syntactical ones in language.

References

[Bae23] John C. Baez, *Isbell duality*, Notices of the American Mathematical Society: Short Stories (2023January). `https://www.ams.org/journals/notices/202301/rnoti-p140.pdf`.

[BK20] Emily M. Bender and Alexander Koller, *Climbing towards NLU: On meaning, form, and understanding in the age of data*, Proceedings of the 58th annual meeting of the association for computational linguistics, July 2020, pp. 5185–5198.

[BNC07] BNC Consortium, *British National Corpus, XML Edition* (Oxford Text Archive, ed.), 2007.

[Bra20] Tai-Danae Bradley, *At the interface of algebra and statistics*, 2020. PhD thesis, CUNY Graduate Center `https://academicworks.cuny.edu/gc_etds/3719/`.

[BTV22] Tai-Danae Bradley, John Terilla, and Yiannis Vlassopoulos, *An enriched category theory of language: From syntax to semantics*, La Matematica **1** (2022), 551–580. `https://doi.org/10.1007/s44007-022-00021-2`.

[CCGP15] Nick Chater, Alexander Clark, John A. Goldsmith, and Amy Perfors, *Empiricism and language learnability*, First edition, Oxford University Press, Oxford, United Kingdom, 2015. OCLC: ocn907131354.

[CCS10] Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh, *Mathematical foundations for a compositional distributional model of meaning*, 2010.

[dF22] Giovanni de Felice, *Categorical tools for natural language processing*, 2022. PhD thesis, University of Oxford.

[ea] D. Smilkov et. al., *Embedding projector: Interactive visualization and interpretation of embeddings*. `https://projector.tensorflow.org`. Accessed: 2023-08-02. Model: "Word2Vec All".

[Ell17] Jonathan A. Elliott, *On the fuzzy concept complex*, 2017. PhD thesis, University of Sheffield `https://etheses.whiterose.ac.uk/18342/`.

[Fir57] John Rupert Firth, *A synopsis of linguistic theory 1930–1955*, Studies in linguistic analysis, 1957, pp. 1–32.

[GW99] Bernhard Ganter and Rudolf Wille, *Formal concept analysis*, Mathematical Foundations, Springer, 1999.

[Har70] Zellig Harris, *Distributional structure*, Papers in structural and transformational linguistics, 1970, pp. 775–794.

[LG14] Omer Levy and Yoav Goldberg, *Neural word embedding as implicit matrix factorization*, Proceedings of the 27th international conference on neural information processing systems - volume 2, 2014, pp. 2177–2185.

[LGD15] Omer Levy, Yoav Goldberg, and Ido Dagan, *Improving distributional similarity with lessons learned from word embeddings*, Transactions of the Association for Computational Linguistics **3** (2015), 211–225.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, *Distributed representations of words and phrases and their compositionality*, Advances in neural information processing systems, 2013.

[Pav12] Dusko Pavlovic, *Quantitative concept analysis*, Formal concept analysis, 2012, pp. 260–277.

[Sau59] Ferdinand de Saussure, *Course in general linguistics*, McGraw-Hill, New York, 1959. Translated by Wade Baskin.

[Wik] Wikimedia Foundation, *Wikimedia downloads*. `https://dumps.wikimedia.org`, 20220301.en dump.

[Wil13] Simon Willerton, *Tight spans, isbell completions, and semi-tropical modules*, Theory and Applications of Categories **28** (2013), 696–732.