Distributional Compositional Models of Discourse (Thesis Abstract)

Lachlan McPheat
Department of Computer Science
University College London, United Kingdom
l.mcpheat at ucl.ac.uk

I. PROBLEM STATEMENT

How should a computer understand written language? For the last 10 or so years, the answer has been *language models*, and in particular the *word embeddings* learned by them. Embeddings are vectors representing word meaning and they have seen vast successes in commercial and academic applications. Though good for modelling word meaning it is unclear how to form embeddings for larger parts of language such as phrases, sentences or discourses (i.e. written language consisting of two or more sentences). We ask:

How do we *meaningfully* represent discourse?

The goal of my thesis is to create a distributional compositional model of written discourse. By *distributional* I mean that the input is distributional[1] data, such as language models, which *composes* to form the meaning of the whole discourse.

II. BACKGROUND

Distributional Compositional Categorical (DisCoCat) models arose from the lack of grammatical structure in conventional language models. Consider how to represent the meaning of the sentence "*dogs eat snacks*". We have a few choices:

1) Extract a sentence embedding from a hidden layer of a large language model. This works but how the meaning is computed is opaque.
2) Let the sentence embedding be the sum of the embeddings for the words in the sentence.

Using arrows to denote embeddings (i.e. $\overrightarrow{dogs}$ is an embedding for the word *dogs*), the second method means that sentences are represented as:

$$\overrightarrow{dogs\ eat\ snacks} = \overrightarrow{dogs} + \overrightarrow{eat} + \overrightarrow{snacks} \tag{1}$$

but addition is commutative, so (1) is the same as both

$$\overrightarrow{snacks} + \overrightarrow{eat} + \overrightarrow{dogs} \tag{2}$$

$$\overrightarrow{eat} + \overrightarrow{dogs} + \overrightarrow{snacks} \tag{3}$$

where in (2) we have the meaning of a different sentence "*snacks eat dogs*" and in (3) we have no sentence at all "*eat dogs snacks*", yet mathematically (1), (2) and (3) are equal.

This lack of structure and composition is resolved in 2010 by Coecke, Sadrzadeh and Clark [1] by making grammar a necessary datum for learning word embeddings. They used a symbolic model of grammar as its base, the *Lambek calculus*, **L**. Given a set of atomic types $\{n, s\}$ where n represents nouns (or noun phrases) and s sentences, we can generate **L** types for adjectives and verbs and so on using the three connectives $\backslash, \bullet$ and $/$. The slashes $\backslash$ and $/$ let us form functional word types, and $\bullet$ represents concatenation of words.

By functional word types we mean word types that modify others. For example, we think of adjectives as functions from nouns to noun phrases, and verbs as taking subjects (and objects) and return sentences. Not only are there many functional word types, but the position of their input is necessary to understand them. For example, in English we can parse "*blue car*", but not "*car blue*", so adjectives take input nouns on the *right*, and verbs take subjects on the *left* (and objects on the right) and return a sentence. This order sensitivity is modelled via the two slashes, by typing adjectives as n/n and verbs as $n\backslash s$[2]. Note that this is how **L** is interpreted for English, but a similar analysis is possible for many languages of varied grammars (so far Japanese, Arabic, most Indo-European languages and Chinese languages).

Once we know the **L**-types of words, we can parse strings of English using deductions of **L**. For example "*Mary sleeps.*" is typed *Mary* : n and *sleeps* : $n\backslash s$ and then, concatenation gives us *Mary sleeps* : $n \bullet n\backslash s$. The formula $n \bullet n\backslash s$ is proven equivalent to sentence type s as shown in (4) below, where horizontal lines denote deduction and horizontal juxtaposition represents $\bullet$.

$$\dfrac{\dfrac{Mary}{n} \quad \dfrac{sleeps}{n\backslash s}}{s} \tag{4}$$

Although a popular model of grammar, the problem of creating sentence embeddings was not answered with **L** alone. One needed a way to see the grammatical structure of **L** in the vector structure of embeddings. This was done using *category theory* to translate the logical structure of **L** into linear algebra, giving us the 'cat' in DisCoCat.

DisCoCat has outperformed standard neural models in a variety of word disambiguation and sentence similarity tasks

[1]This name comes from the *distributional hypothesis* of Firth [2] which states that words with similar contexts have similar meanings.

[2]It helps to think of these slashes as directed fractions, where a/b has b as a righthand denominator, and $c\backslash d$ has c as a lefthand denominator.

[5], [8], [9] albeit on limited academic datasets. Scaling DisCoCat to industrial applications is actively studied using quantum computers [6].

However, DisCoCat cannot model beyond sentence level fundamentally because $\mathbf{L}$ cannot parse beyond sentence level, i.e. discourse level. This is because at discourse level, language contains referential words and phrases. For example "*Sam sleeps. He snores.*" is a 2 sentence discourse where *He* means *Sam*. In $\mathbf{L}$ there is no way to analyse this meaningfully, which led Gerhard Jäger to develop a way to symbolically parse reference [3]. Jäger, inspired by linguistic theory, argued that reference is made up of **copying** and **movement**: the word being referred to should be copied, then one copy is moved to and identified with the referring word, as demonstrated in figure 1. Once this copying and movement

(Starting point)	*Sam sleeps. He snores.*
(Copying)	*Sam [Sam] sleeps. He snores.*
(Movement)	*Sam sleeps. [Sam] He snores.*
(Identification)	*Sam sleeps. Sam$_{=He}$ snores.*

Fig. 1. Jäger parse of reference.

is done, one parses in $\mathbf{L}$ as before. However, copying and moving in $\mathbf{L}$ is not possible since $\mathbf{L}$ formulas represent words and their concatenation represents word order. Hence we cannot copy and move formulas, since we cannot freely repeat or move words when writing. The resulting question, answered in my thesis, is

> How do we extend DisCoCat to logics that can parse discourse?

III. RESEARCH

We identified a logic that handles reference: $\mathbf{SLLM}$, the *Lambek calculus with soft subexponentials* [4]. $\mathbf{SLLM}$ has two modalities, one for copying ! and one for moving ∇, letting us type referential words as functional words that take copies as input and return those copies (just like how *He* returned *Sam* in figure 1). Hence we type referential words like *He* as $\nabla n \backslash n$, to say that *He* looks for a copy of an ∇n-type word on its left, and returns the copy. Whereas referable words like *Sam* are typed $!\nabla n$, because *Sam* is copyable (!) and movable (∇). This lets us parse *Sam sleeps. He snores.* as a discourse where *He* means *Sam*, as seen in (5).

$$\dfrac{\dfrac{\dfrac{\dfrac{\textit{Sam}}{!\nabla n}}{\nabla n \bullet \nabla n}}{\dfrac{\nabla n}{n}} \quad \dfrac{\textit{sleeps}}{n\backslash s} \quad \dfrac{\dfrac{\textit{[Sam]}}{\nabla n} \quad \dfrac{\textit{He}}{\nabla n \backslash n}}{n} \quad \dfrac{\textit{snores}}{n\backslash s}}{s} \Big/ s \bullet s \tag{5}$$

We defined a model of $\mathbf{SLLM}$-formulas as finite dimensional vector spaces, and words as vectors in those spaces [7]. The model has atomic spaces N and S for noun and sentence types, containing vectors like $\overrightarrow{Sam} \in N$ and $\overrightarrow{Sam\ sleeps} \in S$. Functional word types like adjectives n/n are interpreted as the tensor space $N \otimes N$ and intransitive verb type is $N \otimes S$ and so on. We interpret !-formulas as bounded Fock-spaces,

$$!V = \mathbb{R} \oplus V \oplus (V \otimes V) \oplus \cdots \oplus V^{\otimes k} \tag{6}$$

for some constant k. This model gives the original DisCoCat embeddings of [1] reference structure, allowing us to form embeddings of whole discourses in a structured way.

Surprisingly, the category theory used in our model makes it compatible with quantum software leading us to run it on a quantum computer, testing it on a classification task [10]. Specifically, the task was to classify if the object or subject pronoun as being referred to in discourses of the form

> The girls ate the cookies. They looked hungry. (7)
>
> The men enjoyed the pancakes. They were tasty. (8)

where in (7) "*They*" refers to the subject "*The girls*" but in (8) refers to the object "*the pancakes*". We saw that models with grammatical and discourse structure were more accurate than simply neural models.

IV. OUTLOOK

I have two main questions for future research:

1) How can we use the vector semantics of $\mathbf{SLLM}$ to improve *transparency* in pronoun resolution?
2) How do we incorporate other forms of reference, like event-anaphora?

More broadly, it remains to be understood how to leverage the use of structure, like grammar and logic, in artificial intelligence. How do we create architectures that allow structure to be introduced in applications other than language?

REFERENCES

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

[2] John R Firth. A synopsis of linguistic theory, 1930-1955, 1957.

[3] Gerhard Jäger. A multi-modal analysis of anaphora and ellipsis. *University of Pennsylvania Working Papers in Linguistics*, 5(2):2, 1998.

[4] Max Kanovich, Stepan Kuznetsov, Vivek Nigam, and Andre Scedrov. Soft Subexponentials and Multiplexing. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 500–517. 2020.

[5] Dimitri Kartsaklis and Mehrnoosh Sadrzadeh. Prior disambiguation of word tensors for constructing sentence vectors. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1590–1601, Seattle, Washington, USA, oct 2013. Association for Computational Linguistics.

[6] Robin Lorenz, Anna Pearson, Konstantinos Meichanetzidis, Dimitri Kartsaklis, and Bob Coecke. Qnlp in practice: Running compositional models of meaning on a quantum computer, 2021.

[7] Lachlan McPheat, Gijs Wijnholds, Mehrnoosh Sadrzadeh, Adriana Correia, and Alexis Toumi. Anaphora and ellipsis in lambek calculus with a relevant modality: Syntax and semantics. *Journal of Cognitive Science*, 22(2):1–34, 2021.

[8] Mehrnoosh Sadrzadeh, Stephen Clark, and Bob Coecke. The Frobenius anatomy of word meanings I: Subject and object relative pronouns. *Journal of Logic and Computation*, 2013.

[9] Mehrnoosh Sadrzadeh, Dimitri Kartsaklis, and Esma Balkır. Sentence entailment in compositional distributional semantics. *Annals of Mathematics and Artificial Intelligence*, 2018.

[10] Hadi Wazni, Kin Ian Lo, Lachlan McPheat, and Mehrnoosh Sadrzadeh. A quantum natural language processing approach to pronoun resolution, 2022.