0.1 Training QNLP models

In the previous section, we have showed how to evaluate a given DisCoCat model on a quantum computer. But where do we get this model from in the first place? This section will review previous approaches to solving this problem and how we can adapt them to noisy intermediate-scale quantum (NISQ) computers.

0.1.1 DisCoCat models via knowledge graph embedding

The first two DisCoCat papers [CCS08; CCS10] focused on the mathematical foundations for their models, assuming that the meaning for words was given they showed how to compute the meaning for grammatical sentences. Grefenstette and Sadrzadeh [GS11] gave a first implementation of a DisCoCat model for a simple pregroup grammar $G = (V, X, D, s)$ made of common nouns and transitive verbs. Concretely, their vocabulary is given $V = E + R$ for some finite sets E and R which we may call *entities* and (binary) *relations*. They take basic types $X = \{s, n\}$ and dictionary $D = \{(e, n)\}_{e \in E} \cup \{(r, n^r s n^l)\}_{r \in R}$. Thus, every grammatical sentence is of the form $f : xry \to s$ for what we may call a *triple* of subject-verb-object $(x, r, y) \in L(G) \simeq E \times R \times E$. They define a DisCoCat model $F : \mathbf{G} \to \mathbf{Mat}_{\mathbb{R}}$ with $F(s) = 1$ and a hyper-parameter $F(n) = d \in \mathbb{N}$ using the following recipe:

1. extract a co-occurrence matrix from a corpus of text and compute a d-dimensional word vector $F(e) \in \mathbb{R}^d$ for each noun $e \in E$,

2. for each transitive verb $r \in R$, find the set $K_r \subseteq E \times E$ of all pairs of nouns $(x, y) \in K_r$ such that the sentence $xry \in E \times R \times E$ occurs in the corpus, then define

$$F(r) = \sum_{(x,y) \in K_r} F(x) \otimes F(y)$$

The meaning of a sentence $f : xry \to s$ would then given by the inner product $F(f) = \langle F(x) \otimes F(y) | F(r) \rangle$ which we can rewrite as

$$F(f) = \sum_{(x',y') \in K_r} \langle F(x) | F(x') \rangle \langle F(y) | F(y') \rangle$$

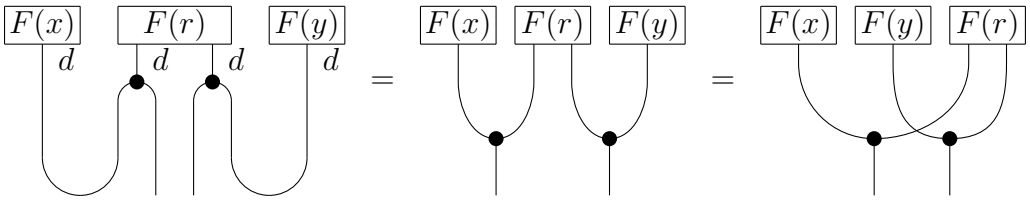
i.e. we take the sum of the similarities between our subject-object pair (x, y) and the pairs (x', y') that appeared in the corpus. However, the task they aim to solve is *word sense disambiguation* which they cast in terms of *sentence similarity*, but if the meaning of sentences are given by scalars there is no meaningful way compare

them. For example, the verb "draw" can be synonymous to "sketch" but also to "pull". Thus, they want their model to predict that "Bob draws diagrams" is more similar to "Bob sketches diagrams" than to "Bob pulls diagrams". Using a spider trick that has been later formalised by Kartsaklis et al. [KSP12], they decide to replace inner product by element-wise multiplication. This amounts to defining a new functor $F' : \mathbf{G} \to \mathbf{Mat}_{\mathbb{R}}$ by post-composing verb meanings with spiders, i.e.

$$F'(n) = F(n) = d, \quad F'(s) = d^2, \quad F'(e) = F(e)$$

$$\text{and} \quad F'(r) \;=\; F(r) \fatsemi \texttt{spider}_{1,2}(d) \otimes \texttt{spider}_{1,2}(d)$$

Indeed, the element-wise multiplication of two vectors $u, v \in \mathbb{R}^d$ can be defined as $u \odot v = u \otimes v \fatsemi \texttt{spider}_{2,1}(d)$, from which we get the desired meaning for the sentence:



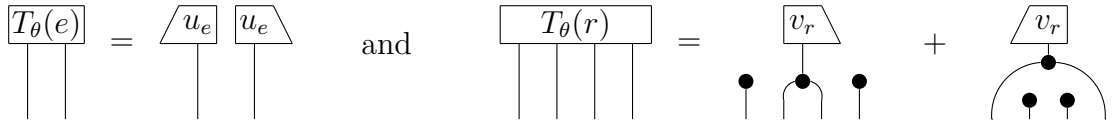
Now that sentence meanings are given by d^2-dimensional vectors rather than scalars, we can compute their similarity with inner products and use this to solve the disambiguation task.

The key observation which motivated our previous dissertation [Tou18] as well as the subsequent articles [Coe+18] and [FMT19] is that a corpus $K \subseteq E \times R \times E$ of such subject-verb-object sentences can be seen as the data for a *knowledge graph*. In this simplified setting, a DisCoCat model can be seen as an instance of *knowledge graph embedding* (KGE) where we want to find a low-dimensional representation of entities and relations. The interpretation of a sentence can be used for *link prediction*, where we want generalising the corpus to unseen sentences. By extending the grammar with the words "who" and "whom", we can use the same model to solve simple instances of question answering. As discussed in section **??**, extending the grammar with anaphora which we interpret as spiders then allows to answer any conjunctive query.

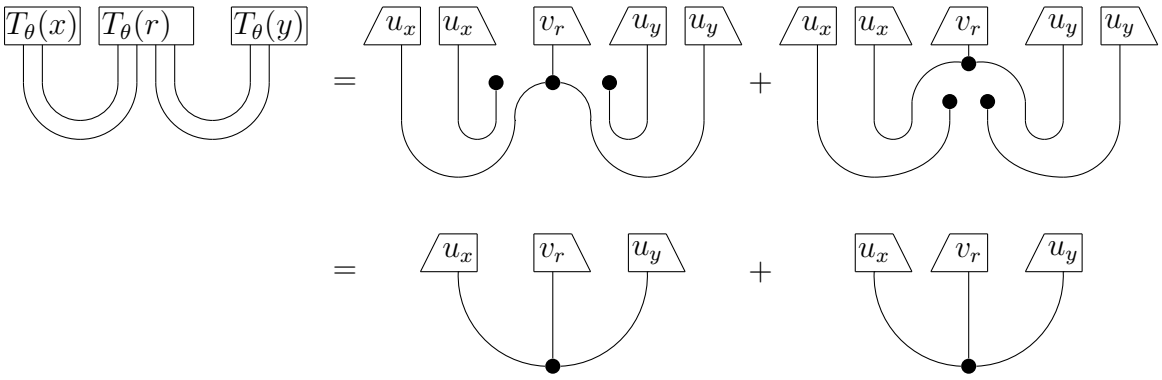
Taking this DisCoCat-KGE analogy in the other direction, we can interpret knowledge graph embeddings as DisCoCat models. Take for example[1] the ComplEx model of Trouillon et al. [Tro+16; Tro+17], it is defined by a dimension $d \in \mathbb{N}$, a (normalised) complex vector $u_e \in \mathbb{C}^d$ for each entity $e \in E$ and a complex vector

[1] We focus on ComplEx, other examples of KGE as functors are treated in [Fel22, Section 2.6].

$v_r \in \mathbb{C}^d$ for each relation $r \in R$. Let us pack this into a dependent pair $\theta \in \Theta = \coprod_{d \in \mathbb{N}} \mathbb{C}^{d(|E|+|R|)}$. The interpretation of a triple $f : xry \to s$ is given by the scoring function $T_\theta(f) = 2\mathrm{Re}(\langle u_x, v_r, u_y^\star \rangle)$, where $\langle u, v, w \rangle = \sum_{i \leq d} u_d v_d w_d$ is a multi-linear generalisation of inner product, $u_y^\star$ is the element-wise conjugate, and 2Re takes twice[1] the real part of a complex number. We can reformulate this as the complex-valued DisCoCat model $T_\theta : \mathbf{G} \to \mathbf{Mat}_\mathbb{C}$ given by $T_\theta(s) = 1$, $T_\theta(n) = d^2$ and:



where we draw the conjugate of a vector as the horizontal reflection of its box. We can use the same spider trick as above to rewrite $\langle u_x, v_r, u_y^\star \rangle$ in terms of post-composition with a three-legged spider. We can also rewrite the real part of a scalar as half the sum with its conjugate $2\mathrm{Re}(z) = z + \bar{z}$, from which we get the desired meaning for sentences:



The meaning of a sentence $f : xry \to s$ is given by a real scalar which we interpret as true when $T_\theta(f) \geq 0$. In fact, any knowledge graph $K : E \times R \times E \to \{\pm 1\}$ can be written as $K = T_\theta \fatsemi \mathtt{sign}$ for the function $\mathtt{sign} : \mathbb{R} \to \{\pm 1\}$ [Tro+17, Theorem 4]. Furthermore, the dimension $d \in \mathbb{N}$ of the model can be bounded by the *sign-rank* of the matrices for each relation [Fel22, Proposition 2.5.17], with theoretical guarantees that $d \ll |E|$ if the problem is learnable efficiently [AMY16].

Let us summarise what we have just done: we have defined a space of parameters Φ together with a function $T_- : \Phi \to [\mathbf{G}, \mathbf{Mat}_\mathbb{C}]$ which sends parameters $\theta \in \Theta$ to functors $T_\theta : \mathbf{G} \to \mathbf{Mat}_\mathbb{C}$. Now if we are given a training set $\Omega \subseteq E \times R \times E$ annotated by $Y : \Omega \to \{\pm 1\}$ for whether each triple belongs to the knowledge

[1] We scale the original definition by 2 in order to avoid cluttering the diagrams with $\frac{1}{2}$ scalars.

graph[1], we want to find the parameters that best approximate the data:

$$\theta^\star \;=\; \arg\min_{\theta\in\Theta}\; \lambda\|\theta\| + \sum_{f\in\Omega} \texttt{loss}(T_\theta(f), Y(f))$$

where $\|\theta\|$ is a choice of norm (usually L^2) scaled by some regularisation hyper-parameter $\lambda \geq 0$ and $\texttt{loss} : \mathbb{R}\times\mathbb{R} \to \mathbb{R}^+$ is a choice of loss function, usually the negative log-likelihood of the logistic model $\texttt{loss}(y, y') = \log(1 + \exp(-yy'))$. If we fix the dimension $d \in \mathbb{N}$ (i.e. we take it as a hyper-parameter) we can use stochastic gradient descent to find $\theta^\star$ and hence the optimal DisCoCat model $T_{\theta^\star} : \mathbf{G} \to \mathbf{Mat}_{\mathbb{C}}$. We can use $T_{\theta^\star}$ to predict the value of triples $f \in E\times R\times E - \Omega$ that we have not seen during training, we can answer any conjunctive query by extending the grammar with question words and anaphora.

0.1.2 Learning monoidal functors from data

While supervised machine learning has traditionally been formulated in terms of learning *functions*, in our case we are in fact *learning functors*. Before we attempt to categorify machine learning, let us recall the *probably approximately correct* (PAC) learning framework as introduced by Valiant [Val84] . We start from a set $\mathcal{X}$ called the *instance space*, usually taken to be $\mathbb{B}^n$ or $\mathbb{R}^n$ for some dimension $n \in \mathbb{N}$. A *concept* is a function $c : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{Y}$ is a *label space*, usually taken to be the Booleans $\mathcal{Y} = \mathbb{B}$ so that a concept can be identified with a subset $c \subseteq \mathcal{X}$. Given a *hypothesis* $h \subseteq \mathcal{X}$, a *target concept* $c \subseteq \mathcal{X}$ and a distribution $\mathcal{D}$ over $\mathcal{X}$, we define $\texttt{error}_{c,\mathcal{D}}(h) = \mathbb{P}_{x\sim\mathcal{D}}[h(x) \neq c(x)]$.

A *concept class* is a collection of concepts $\mathcal{C} \subseteq \mathcal{X} \to \mathcal{Y}$, together with a function $\texttt{size} : \mathcal{C} \to \mathbb{N}$ where $\texttt{size}(c)$ is usually the number of bits reuired to encode a concept $c \in \mathcal{C}$. A *learning algorithm* A for a concept class $\mathcal{C}$ is one that takes as input a confidence $\delta > 0$, an error rate $\epsilon > 0$ and a training set $\Omega \subseteq \mathcal{X}\times\mathcal{Y}$ returns a hypothesis $A(\Omega, \epsilon, \delta) \in \mathcal{C}$. A concept class $\mathcal{C}$ is *PAC learnable* whenever there exists a learning algorithm A and a polynomial p such that:

- for all target concepts $c \in \mathcal{C}$,
- for all distributions $\mathcal{D}$ over $\mathcal{X}$,

[1]When the dataset contains only positive triples, it is common use the *local closed world assumption* where we randomly change either the subject or object to generate negative triples. This can be improved by *adversarial sampling* methods [CW18], akin to *generative adversarial networks* where we train a model to generate hard negative examples. In [Fel+20] we investigate how this can be formalised in terms of *functorial language games*.

- for all confidence $\delta > 0$ and error rate $\epsilon > 0$

$$\mathbb{P}_{X \sim \mathcal{D}^N}\left[\texttt{error}_{c,\mathcal{D}}(A(X, \epsilon, \delta)) \geq \epsilon\right] \leq \delta$$

where we draw $N = p(n, \frac{1}{\delta}, \frac{1}{\epsilon})$ samples $X \in \mathcal{X}^N$ independently and identically distributed according to $\mathcal{D}$. $\mathcal{C}$ is *efficiently PAC learnable* if furthermore A runs in time $O(N\texttt{size}(c))$.

References

[AMY16] Noga Alon, Shay Moran, and Amir Yehudayoff. "Sign Rank versus VC Dimension". In: *Conference on Learning Theory*. 2016, pp. 47–80. arXiv: 1503.07648.

[CW18] Liwei Cai and William Yang Wang. "KBGAN: Adversarial Learning for Knowledge Graph Embeddings". Apr. 16, 2018.

[CCS08] Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. "A Compositional Distributional Model of Meaning". In: *Proceedings of the Second Symposium on Quantum Interaction (QI-2008)*. 2008, pp. 133–140.

[CCS10] Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. "Mathematical Foundations for a Compositional Distributional Model of Meaning". In: *A Festschrift for Jim Lambek*. Ed. by J. van Benthem, M. Moortgat, and W. Buszkowski. Vol. 36. Linguistic Analysis. 2010, pp. 345–384. arXiv: 1003.4394.

[Coe+18] Bob Coecke, Giovanni de Felice, Dan Marsden, and Alexis Toumi. "Towards Compositional Distributional Discourse Analysis". In: *Proceedings CAPNS 2018*. EPTCS, Nov. 8, 2018. DOI: 10.4204/EPTCS.283.1.

[Fel22] Giovanni de Felice. "Categorical Tools for Natural Language Processing". University of Oxford, 2022.

[Fel+20] Giovanni de Felice, Elena Di Lavore, Mario Román, and Alexis Toumi. "Functorial Language Games for Question Answering". In: *Proceedings of the 3rd Annual International Applied Category Theory Conference 2020, ACT 2020, Cambridge, USA, 6-10th July 2020*. Ed. by David I. Spivak and Jamie Vicary. Vol. 333. EPTCS. 2020, pp. 311–321. DOI: 10.4204/EPTCS.333.21.

[FMT19] Giovanni de Felice, Konstantinos Meichanetzidis, and Alexis Toumi. "Functorial Question Answering". In: *Proceedings Applied Category Theory 2019, ACT 2019, University of Oxford, UK*. Vol. 323. EPTCS. 2019. DOI: 10.4204/EPTCS.323.6.

[GS11] Edward Grefenstette and Mehrnoosh Sadrzadeh. "Experimental Support for a Categorical Compositional Distributional Model of Meaning". 2011.

[KSP12] Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Stephen G. Pulman. “A Unified Sentence Space for Categorical Distributional-Compositional Semantics: Theory and Experiments”. In: *COLING*. 2012.

[Tou18] Alexis Toumi. “Categorical Compositional Distributional Questions, Answers & Discourse Analysis”. PhD thesis. Master’s thesis, University of Oxford, 2018.

[Tro+17] Théo Trouillon, Christopher R. Dance, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. “Knowledge Graph Completion via Complex Tensor Factorization”. In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 4735–4772. eprint: arXiv:1702.06879.

[Tro+16] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. “Complex Embeddings for Simple Link Prediction”. June 20, 2016.

[Val84] L. G. Valiant. “A Theory of the Learnable”. In: *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*. STOC ’84. New York, NY, USA: ACM, 1984, pp. 436–445. DOI: 10.1145/800057.808710.