QUANTUM COMPUTATIONAL LINGUISTICS

Konstantinos Meichanetzidis

Cambridge Quantum

QUANTINUUM

Solstice of Foundations

ETH Zurich, June 2022

Featuring

Alexis Toumi
Giovanni de Felice

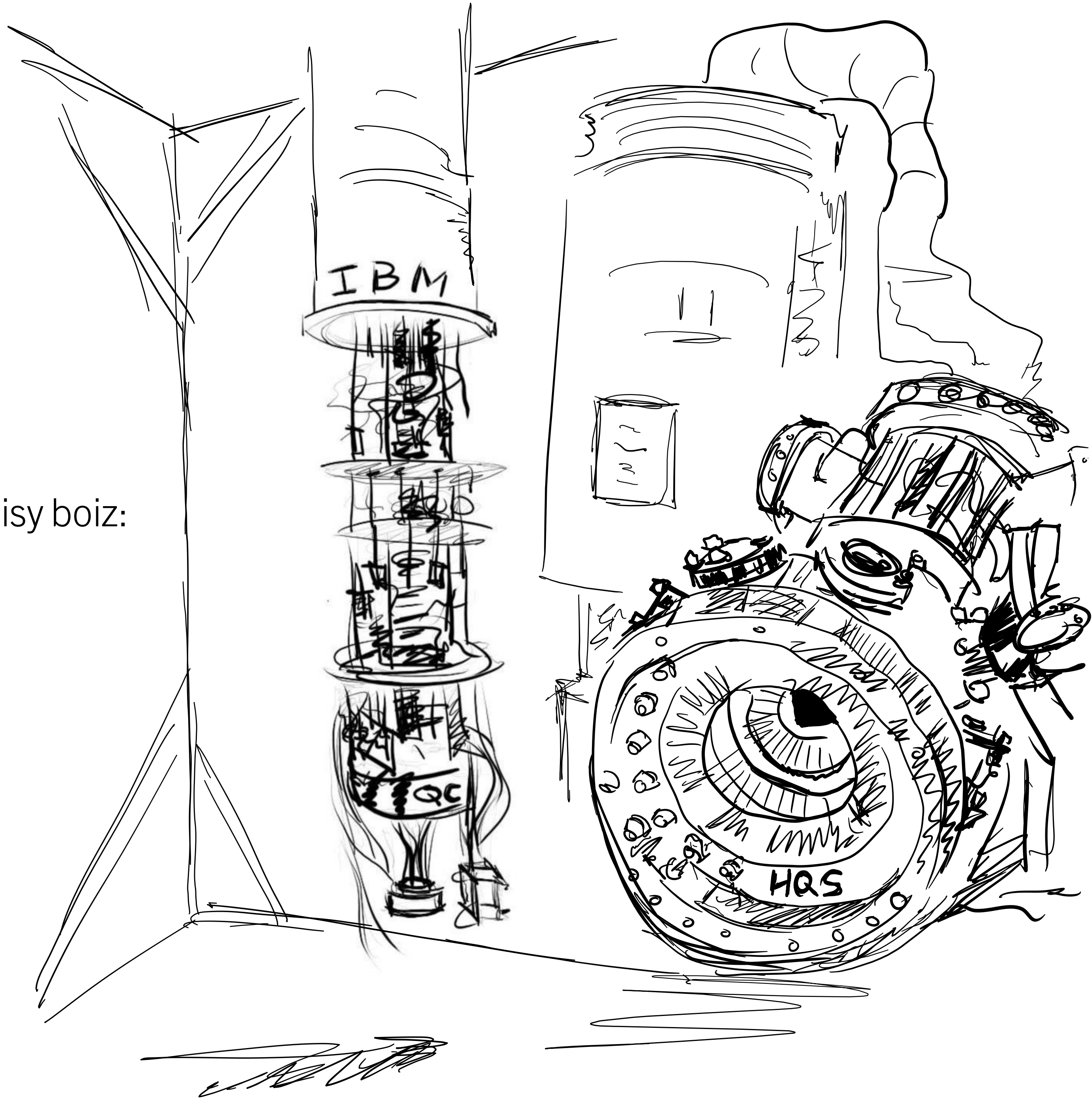
Carys Harvey
Douglas Brown
Anna Pearson
Robin Lorenz

Richie Yeung
Ian Fan
Razin Shaikh
Jonathon Liu
Benjamin Rodatz
Thomas Hoffman

Vid Kocijan
Wenduan Xu

Matty Hoban
Dimitri Kartsaklis
Stephen Clark
Bob Coecke

And these noisy boiz:

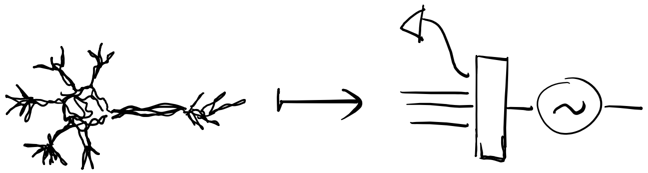


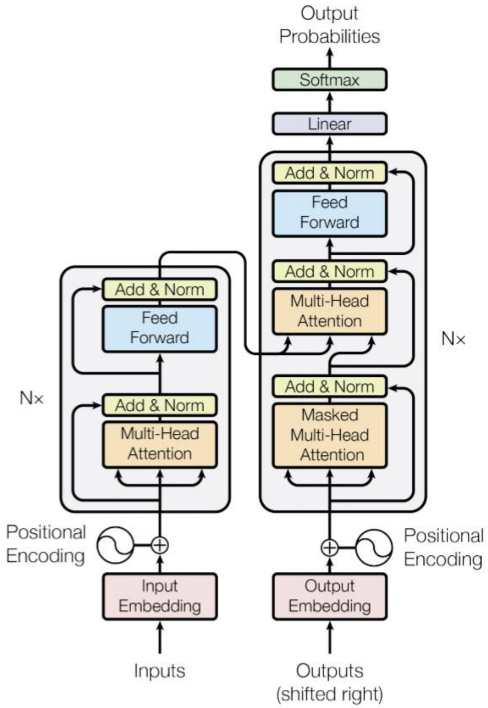
Computational linguistics and natural language processing

- CL: syntax, grammar, formal language theory, top-down rule-based

- NLP: engineering, big text data, machine learning, bottom-up statistical methods

SotA





GPT-3 : 175B params

Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*
Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry
Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan
Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter
Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray
Benjamin Chess Jack Clark Christopher Berner
Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI

LaMDA : 137B params
+ consulting external knowledge

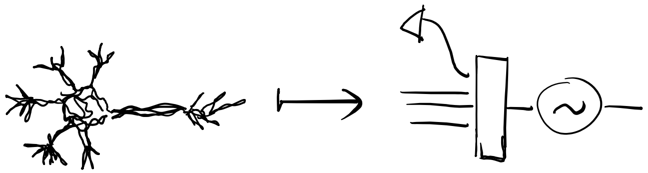
LaMDA: Language Models for Dialog Applications

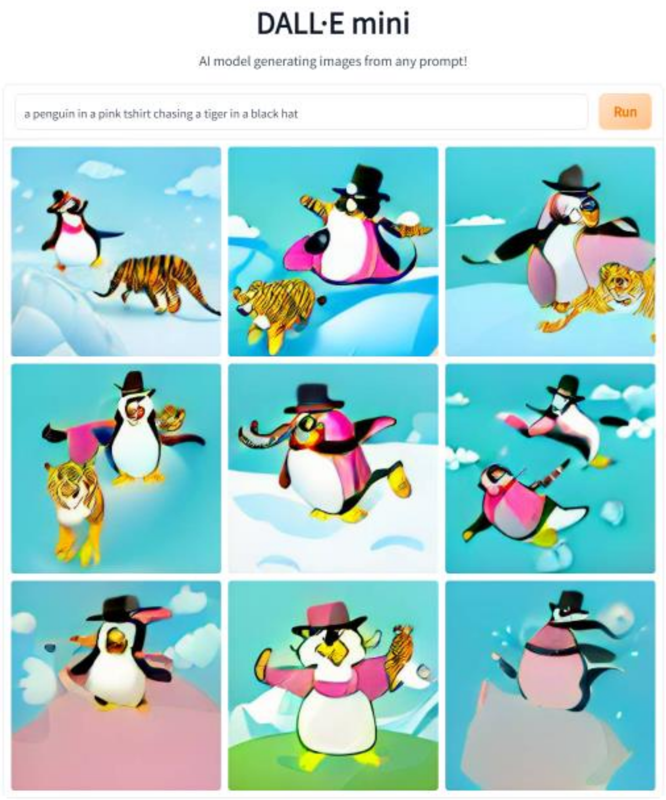
Romal Thoppilan Daniel De Freitas * Jamie Hall Noam Shazeer * Apoorv Kulshreshtha
Heng-Tze Cheng Alicia Jin Taylor Bos Leslie Baker Yu Du YaGuang Li Hongrae Lee
Huaixiu Steven Zheng Amin Ghafouri Marcelo Menegali Yanping Huang Maxim Krikun
Dmitry Lepikhin James Qin Dehao Chen Yuanzhong Xu Zhifeng Chen Adam Roberts
Maarten Bosma Vincent Zhao Yanqi Zhou Chung-Ching Chang Igor Krivokon Will Rusch
Marc Pickett Pranesh Srinivasan Laichee Man Kathleen Meier-Hellstern
Meredith Ringel Morris Tulsee Doshi Renelito Delos Santos Toju Duke Johnny Soraker
Ben Zevenbergen Vinodkumar Prabhakaran Mark Diaz Ben Hutchinson Kristen Olson
Alejandra Molina Erin Hoffman-John Josh Lee Lora Aroyo Ravi Rajakumar
Alena Butryna Matthew Lamm Viktoriya Kuzmina Joe Fenton Aaron Cohen
Rachel Bernstein Ray Kurzweil Blaise Aguera-Arcas Claire Cui Marian Croak Ed Chi
Quoc Le

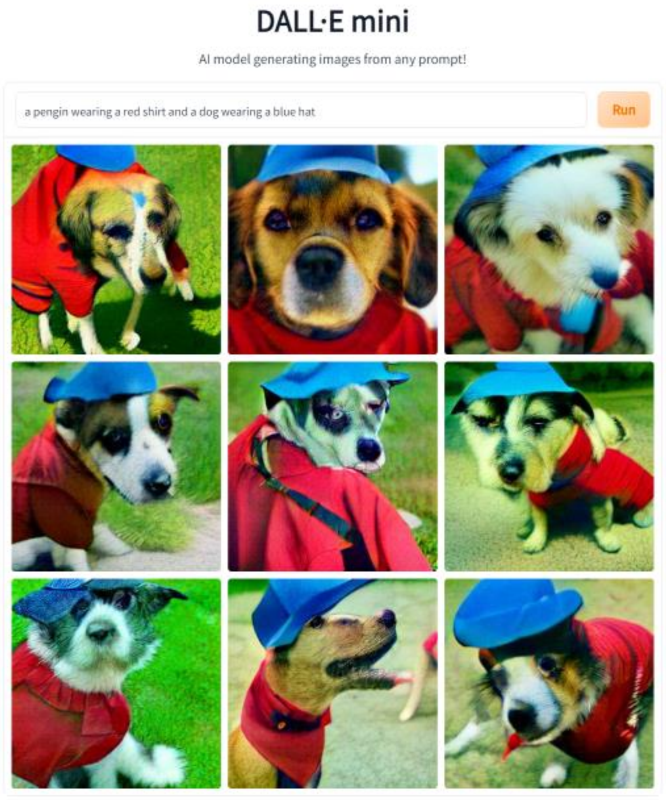
Google

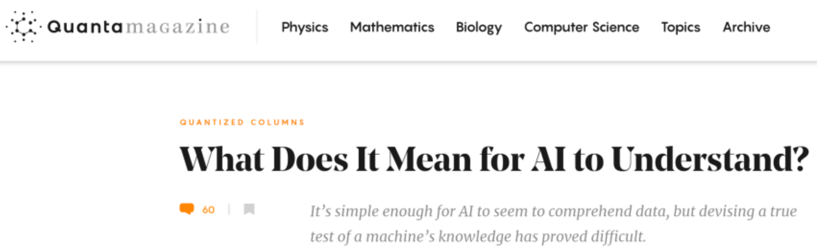
Human Brain: 200 billion neurons, 125 trillion synapses (just in the cerebral cortex).
New larger language models with 100 trillion parameters.

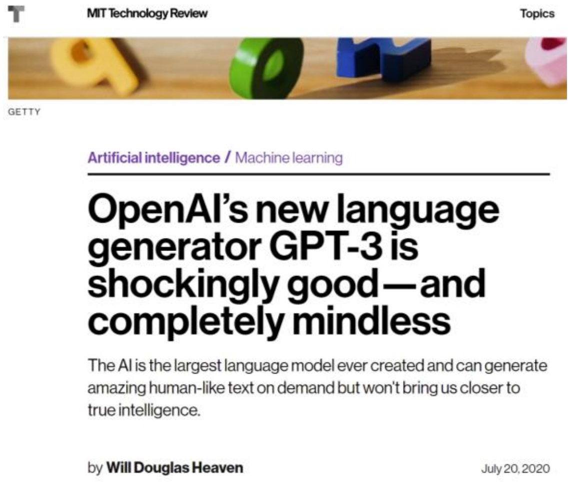
SotA

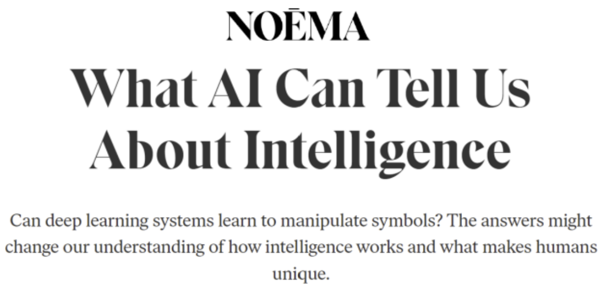












`large language models are expert blabbers' – J Bach
`Deep learning models need symbolic structures hardcoded into them' – G Marcus
`Deep learning models can eventually learn to do anything only from data' – G Hinton

Language is `AI-hard'

"The limits of my language mean the limits of my world"
- Ludwig Wittgenstein

"Language faculty is what separates us from other species"
- Noam Chomsky

Turing test : verify intelligent behaviour via language!

QUANTUM DISCO THEORY

Enter the DisCo

A hybrid model

Simple, less black-boxy model
hopefully provides understandable solution
rather than outsource our hope in scaling up.

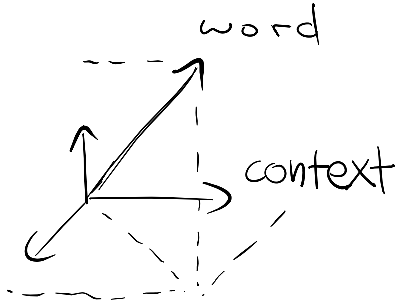
"What is the **mechanics of meaning**?"

DisCoCat [1003.4394] Coecke, Sadrzadeh, Clark

Distributional meaning + Compositional structure

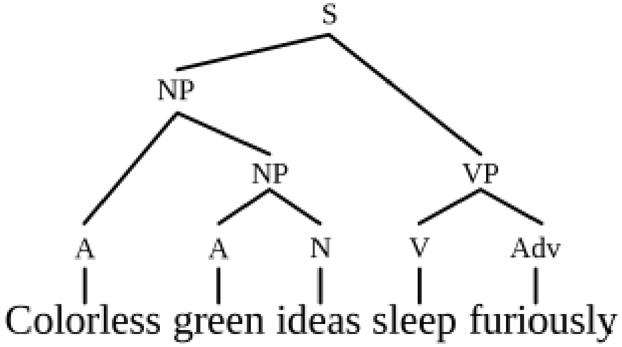
'meaning of a word is the words around it'

John Firth



'grammar can be described by formal rules'

valid nonsense



N. Chomsky

THE BABY DEFINITION OF A CATEGORY

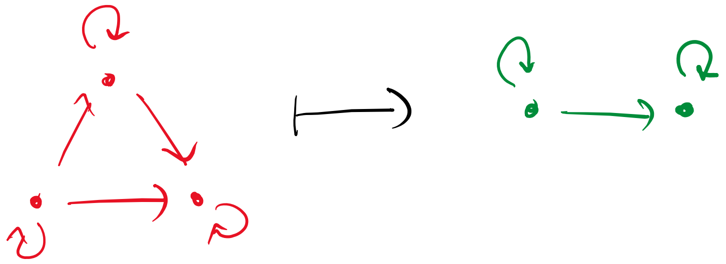
[0905.3010]

A category is:

- a collection of **Objects**.
- with **Arrows** (or ‘transformations’, or ‘morphisms’) between them.

Functor: structure preserving map between two cats:

- sends objects to objects and arrows to arrows
- respects composition



Heraclitus-style framework.
It tries to say as little as possible about objects
and talks about how they transform.
Allows one to think of meaning as analogy.

PROCESS THEORIES AND STRING DIAGRAMS

String diagrams are composed of boxes and wires.

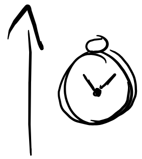
objects : types
arrows : boxes

A box represents a process.

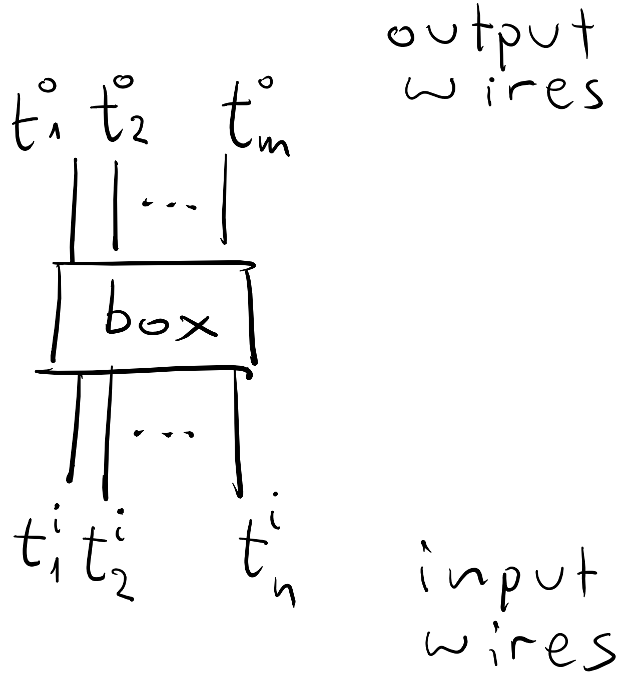
Wires are typed.

[Picturing Quantum Processes (dodo book)]

You have to choose a reading direction ‘foliation’

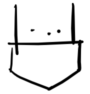


process:

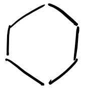


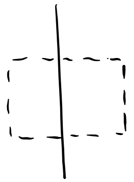
PROCESS THEORIES AND STRING DIAGRAMS

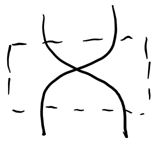
Special cases of boxes:

state :  only output

effect :  only input

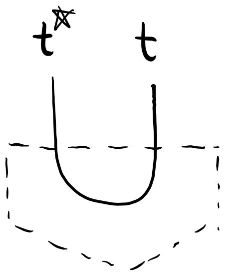
scaler :  no input
no output

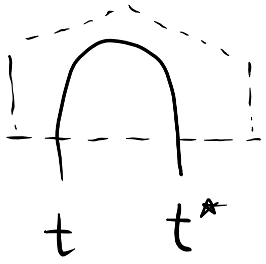
identity :  wire = nothing-box

SWAP('symmetry') : 

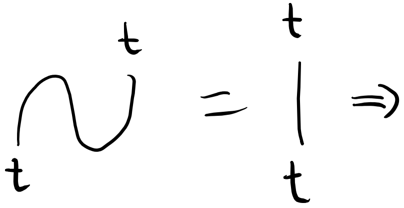
PROCESS THEORIES AND STRING DIAGRAMS

‘Cups’ and ‘caps’: special cases of states and effects:





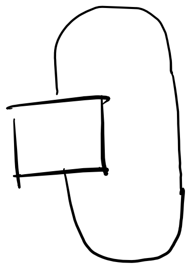
Snake equation: only topology (connectivity) matters:



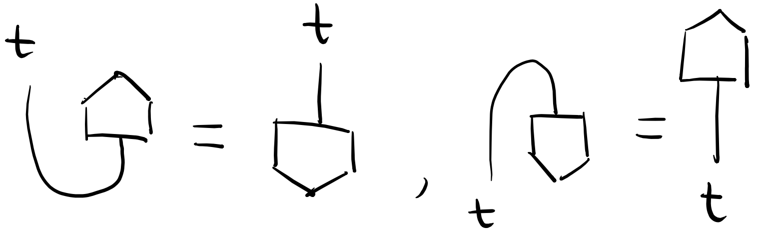
wires can wiggle

PROCESS THEORIES AND STRING DIAGRAMS

Trace:



Transposition:

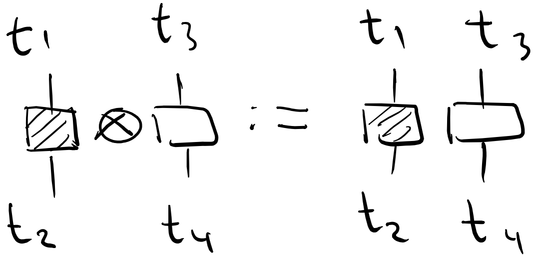


$\Rightarrow$ states $\leftrightarrow$ effects

PROCESS THEORIES AND STRING DIAGRAMS

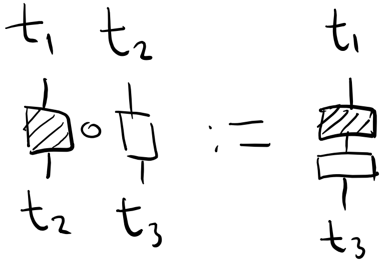
Composition: process theories are symmetric monoidal categories

parallel :



"monoidal" product

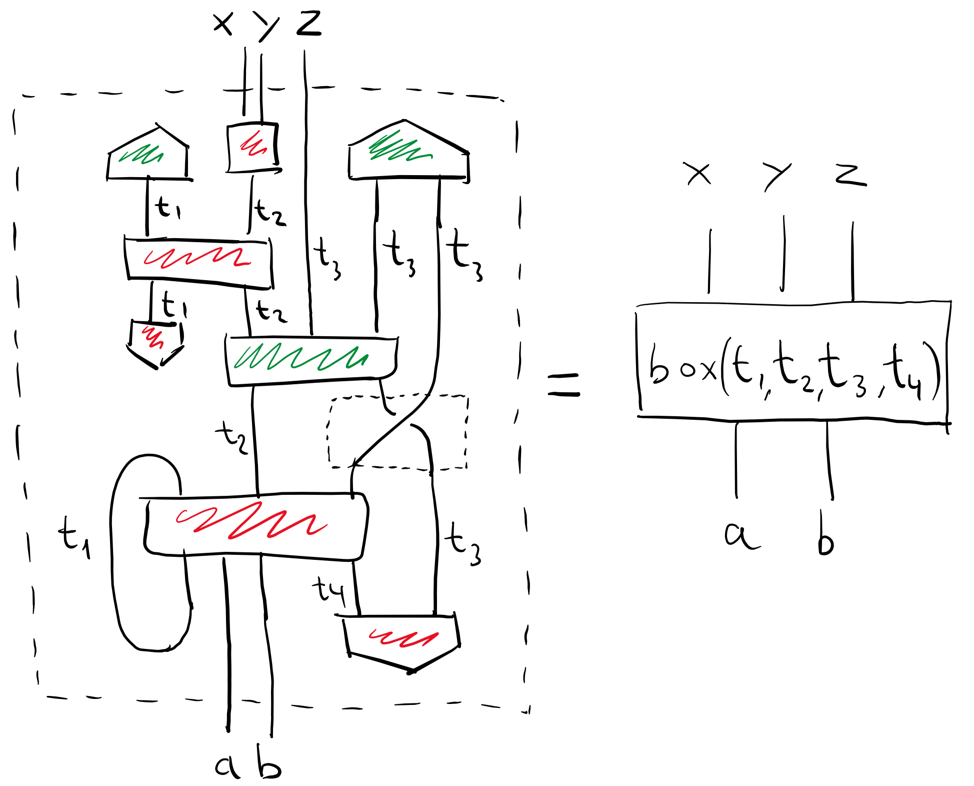
sequential :



types must match!

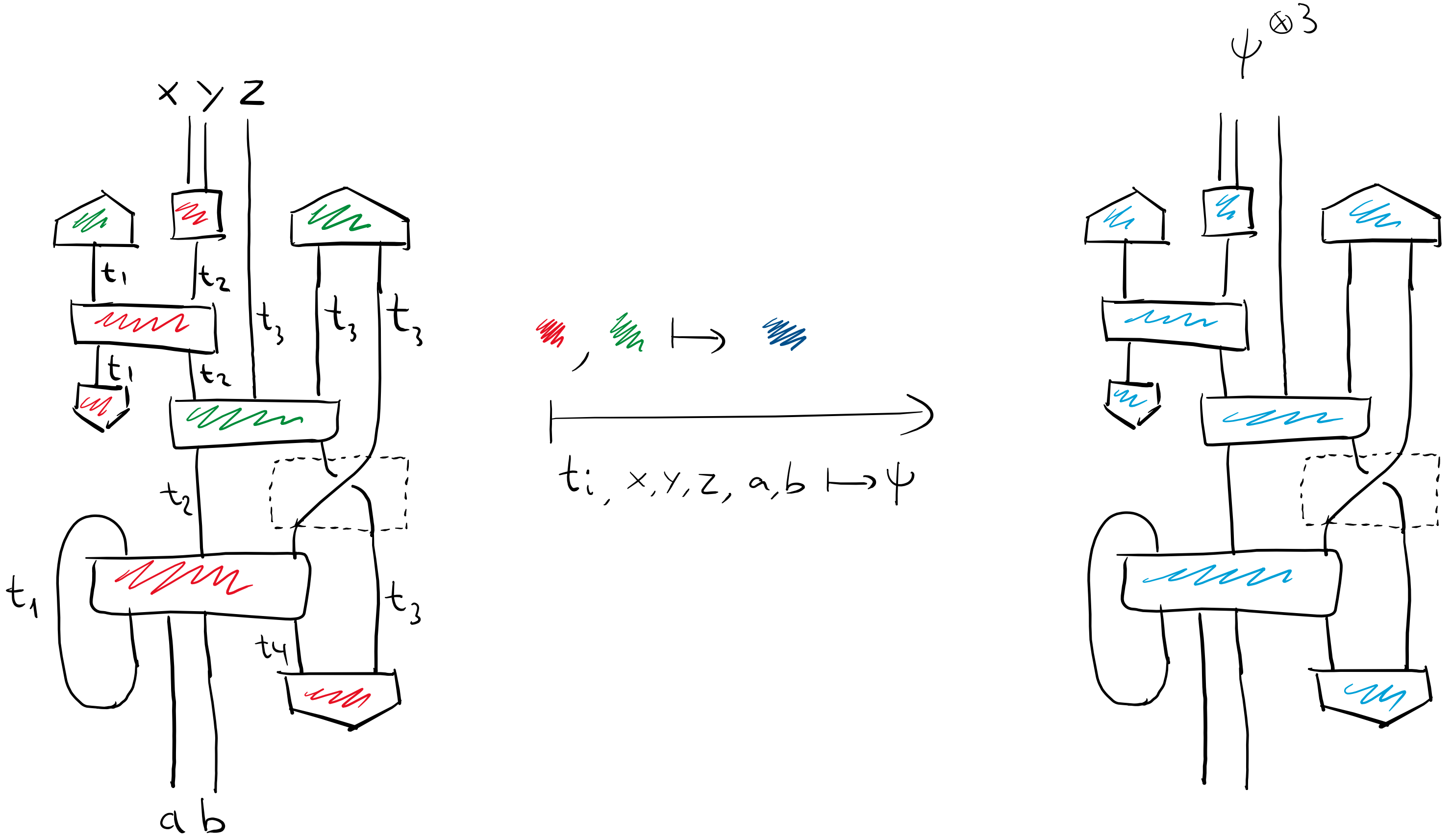
We can **freely compose**
as long as we **respect the types**

example:



FUNCTOR = TYPE-WISE AND BOX-WISE SUBSTITUTIONS

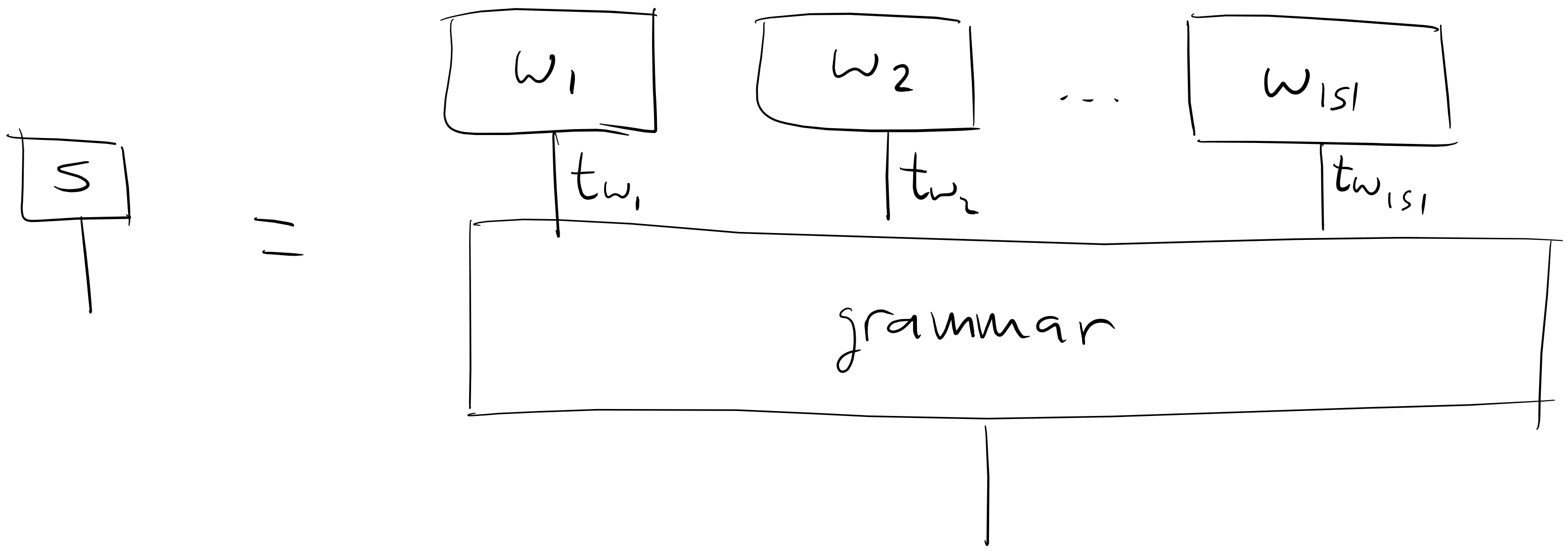
example:



LINGUISTIC PROCESSES

Syntactic structures as string diagrams: the parsing of a sentence under a type-based grammar can be viewed as a process and can be drawn as a string diagram.

types : grammar types
boxes : type-reduction rules



States of parts

Composition

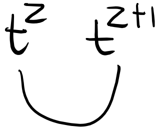
State of whole

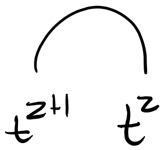
PREGROUP GRAMMAR

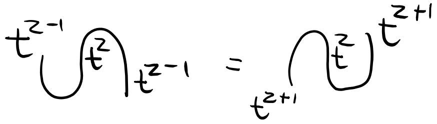
Parser assigns <u>pregroup</u> types to words:

$$w \longrightarrow t_w = t_{i_1}^{z_{i_1}} \cdot t_{i_2}^{z_{i_2}} \cdot \ldots \cdot t_{i_{|w|}}^{z_{i_{|w|}}} \quad , \quad z_i \in \mathbb{Z} \simeq \{\ldots, lll, ll, l, -, r, rr, rrr, \ldots\} \quad , \quad t_i \in B$$

Types reduce as: $t^z \cdot t^{z+1} \to 1$, $1 \to t^{z+1} \cdot t^z$







PREGROUP REDUCTIONS AS STRING DIAGRAMS

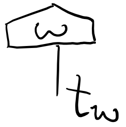
pc runs complex code

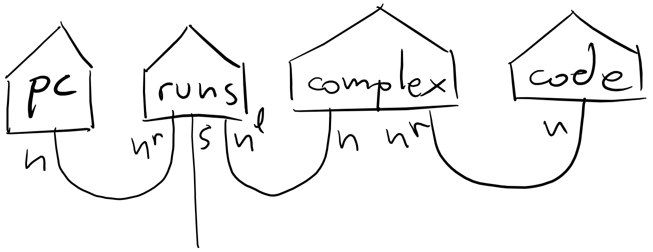
$$\downarrow \qquad \downarrow \qquad \downarrow \qquad \downarrow$$

$$n \cdot (n^r \cdot s \cdot n^l) \cdot (n \cdot n^r) \cdot n$$

$\cdot \mapsto \otimes$

$\longmapsto$

$w \mapsto$ 



COMBINATORY CATEGORIAL GRAMMAR

parser assigns CCG types to words

$w \longrightarrow t_w$

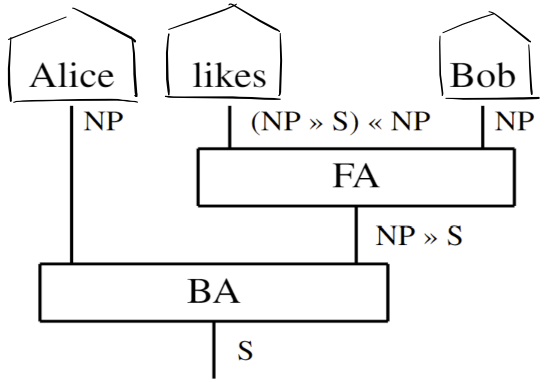
Type reductions:

$$\text{FA } (>)\ \frac{\alpha : X \leftarrowtail Y \qquad \beta : Y}{\alpha\beta : X} \qquad \text{FC } (B_>)\ \frac{\alpha : X \leftarrowtail Y \qquad \beta : Y \leftarrowtail Z}{\alpha\beta : X \leftarrowtail Z} \qquad \text{FTR } (T_>)\ \frac{\alpha : X}{\alpha : T \leftarrowtail (X \rightarrowtail T)}$$

$$\text{BA } (<)\ \frac{\alpha : Y \qquad \beta : Y \rightarrowtail X}{\alpha\beta : X} \qquad \text{BC } (B_<)\ \frac{\alpha : Z \rightarrowtail Y \qquad \beta : Y \rightarrowtail X}{\alpha\beta : Z \rightarrowtail X} \qquad \text{BTR } (T_<)\ \frac{\alpha : X}{\alpha : (T \leftarrowtail X) \rightarrowtail T}$$

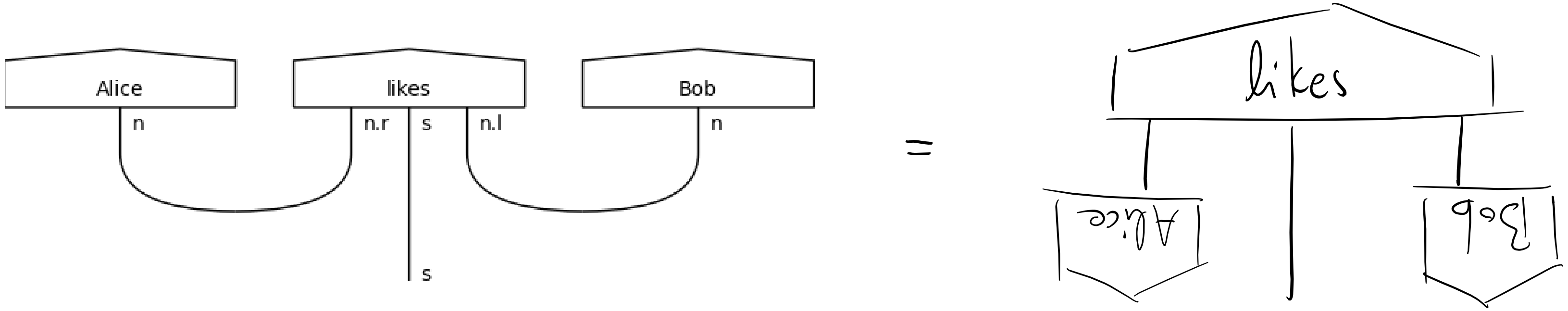
CCG REDUCTIONS AS STRING DIAGRAMS

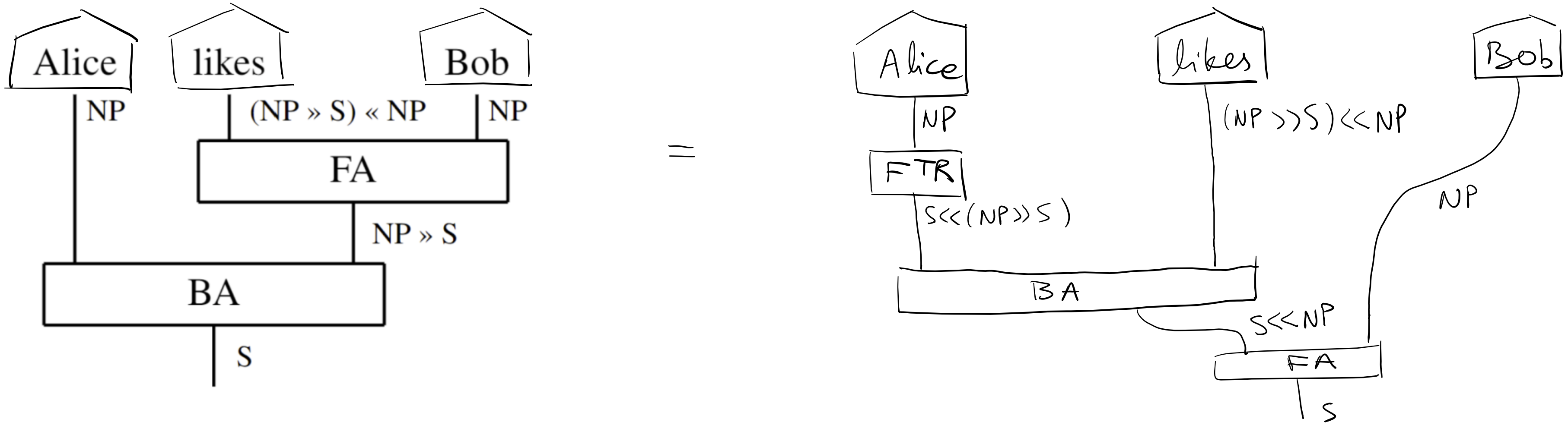
$$\dfrac{\overset{\text{Alice}}{\overline{NP}} \quad \dfrac{\overset{\text{likes}}{\overline{(NP \rightarrowtail S) \leftarrowtail NP}} \quad \overset{\text{Bob}}{\overline{NP}}}{NP \rightarrowtail S}>}{S}< \qquad \longmapsto$$



LINGUISTIC PROCESSES

Semantics-respecting diagram **rewrites**





SEMANTIC FUNCTOR

Grammar

String Diagram

Vect

Hilb

Tensor Network:
runs on CPU

one simulates the other

Quantum Circuit:
runs on QPU

states : vectors

processes : linear maps

effects : co-vectors

states : quantum states

processes : quantum maps

effects : quantum effects

QUANTUM PROCESSES

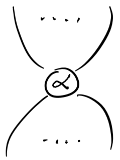
Book: Picturing Quantum Processes
Coecke and Kissinger

Quantum theory is a process theory too!

[0906.4725] : foundational work by Coecke and Duncan

Types : qubit Hilbert spaces
(complex vector spaces of dim $2^n, n \in \mathbb{N}$)
Boxes : Spiders (linear maps)

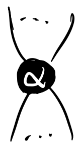
Z-spider :



$$= |0\ldots0\rangle\langle 0\ldots0| + e^{i\alpha}|1\ldots1\rangle\langle 1\ldots1| \quad , \quad \alpha \in [0, 2\pi)$$

X-spider :

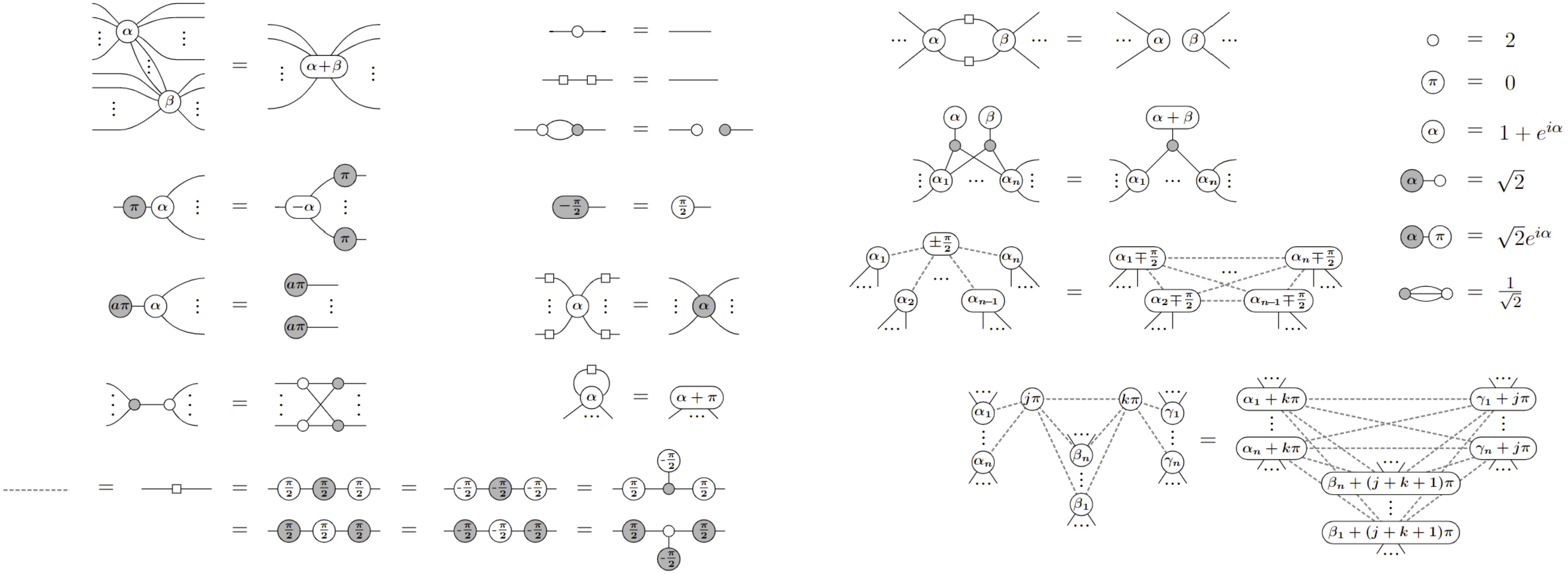
$$|\pm\rangle = |0\rangle \pm |1\rangle$$



$$= |+\ldots+\rangle\langle +\ldots+| + e^{i\alpha}|-\ldots-\rangle\langle -\ldots-| \quad , \quad \alpha \in [0, 2\pi)$$

A.5 ZX-calculus full cheatsheet

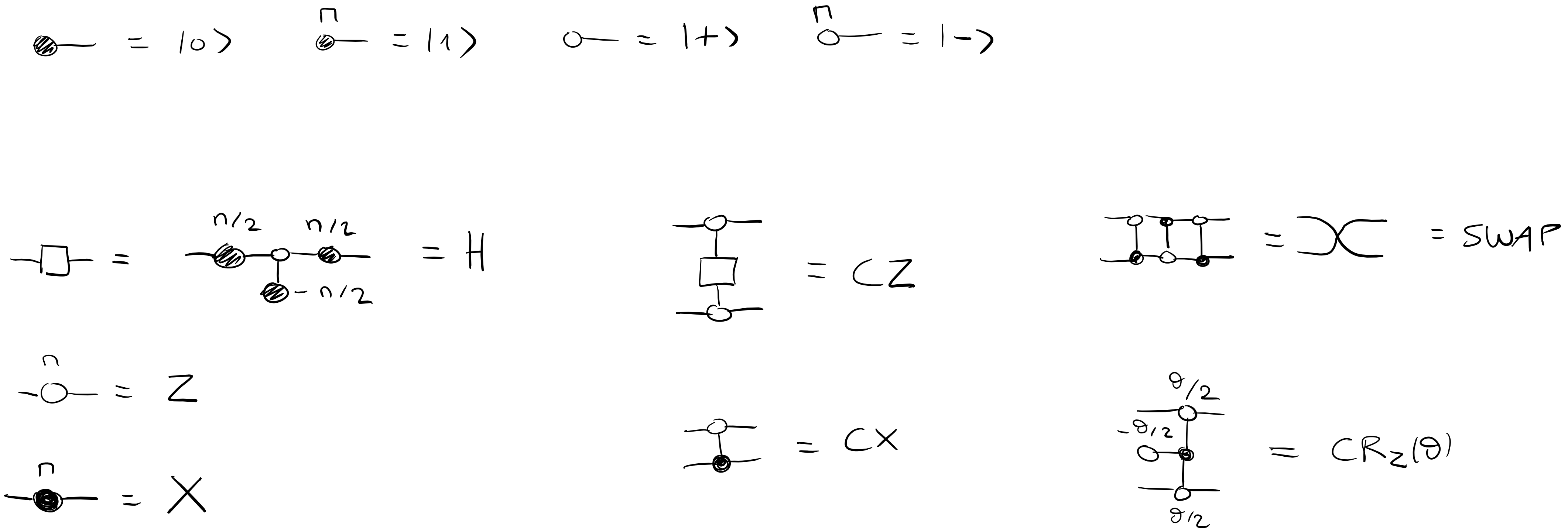
The following rewrite rules hold for all $\alpha, \beta, \alpha_i, \beta_j, \gamma_k \in \mathbb{R}$ and $a \in \{0, 1\}$ (up to global non-zero scalar).



QUANTUM PROCESSES

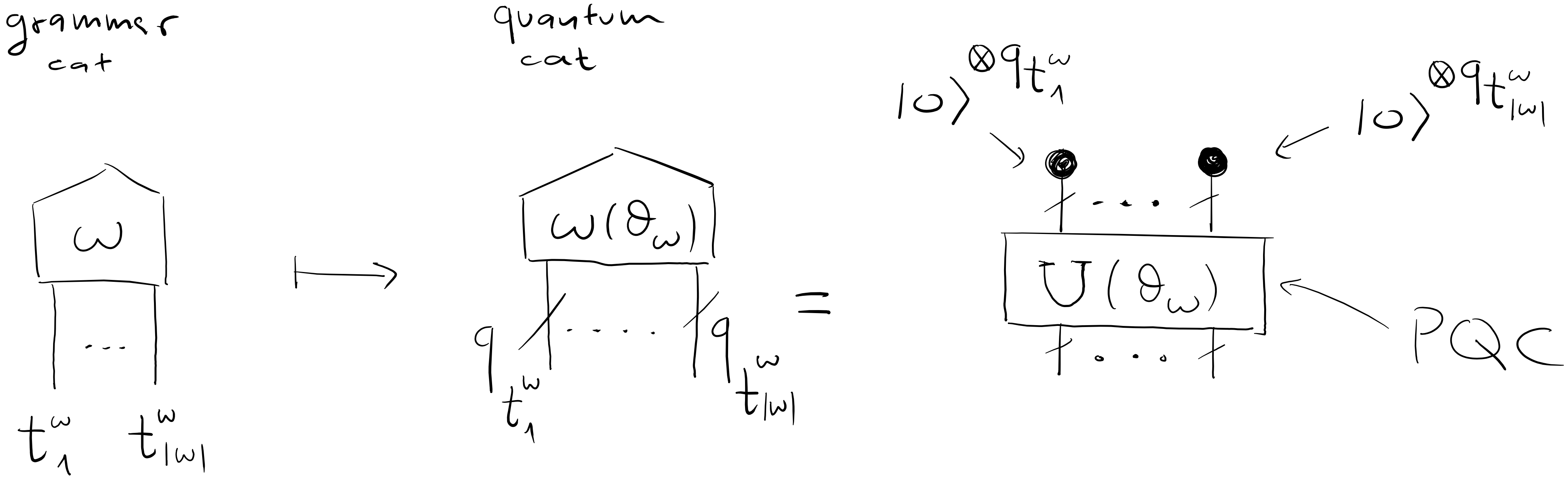
ex

The standard Clifford+T states and gates are built out of spiders.



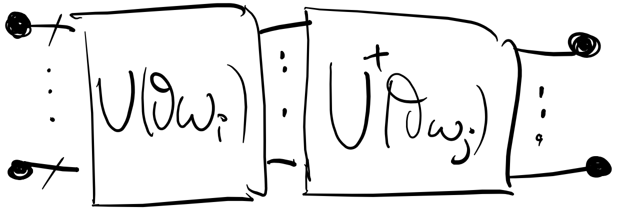
QUANTUM WORD EMBEDDINGS

Quantum states act as word embeddings.



θ_ω: trained from data

QUANTUM WORD EMBEDDINGS *



$= \left|\langle 0 | U(\vartheta_{w_i})\, U^{\dagger}(\vartheta_{w_j}) | 0 \rangle\right|^2 \sim$ word similarity

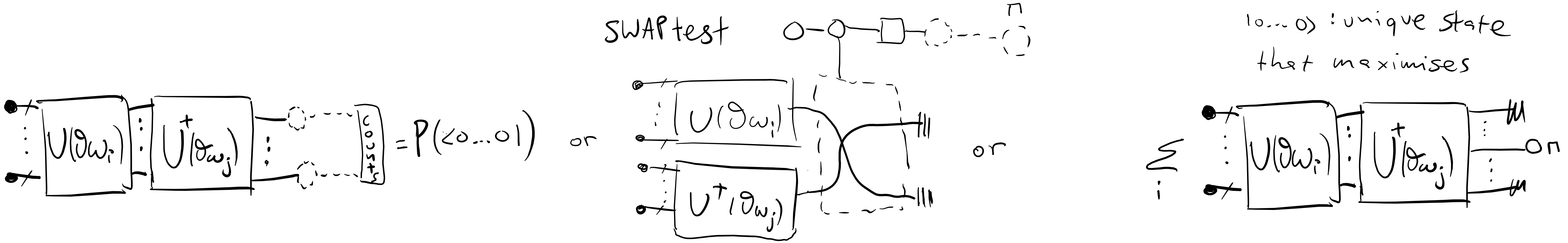
These word-similarities can go into a glove cost func and θ_{w_i} are trained on big text-data from which a cooc matrix X_{ij} is obtained.

$$\min_{\theta} \; \left|\langle 0 | U(\theta_i) U^{\dagger}(\theta_j) | 0 \rangle\right|^2 + b_i + b_j' - \log X_{ij} \qquad \theta = \cup_i \, \theta_i$$

* caveats when training embeddings for the pregroup version

WORD-WORD OVERLAP

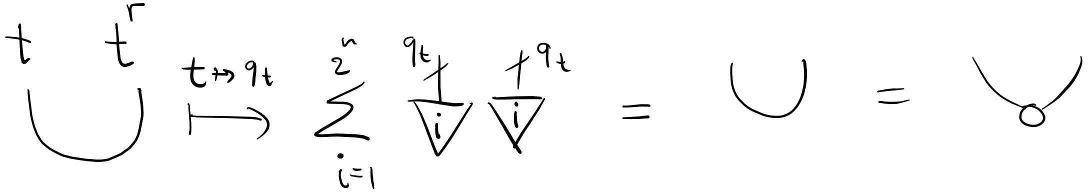
We can estimate $\left|\langle 0|U(\theta_{w_i})U^\dagger\left(\theta_{w_j}\right)|0\rangle\right|^2$ up to additive error with $O\left(|V|^2\right)$ QC calls.



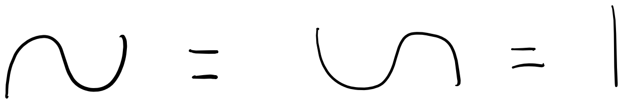
QUANTUM LINGUISTIC PROCESSES

Type reductions are mapped to quantum processes.

pregroups



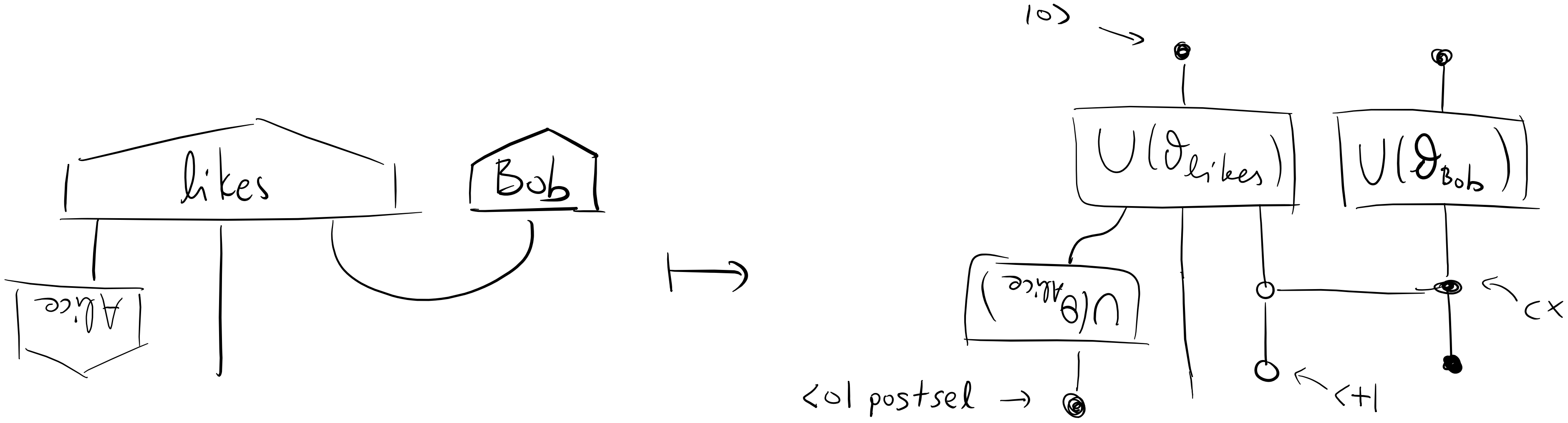
Bell effect



QUANTUM LINGUISTIC PROCESSES

Lambeq
[2110.04236]

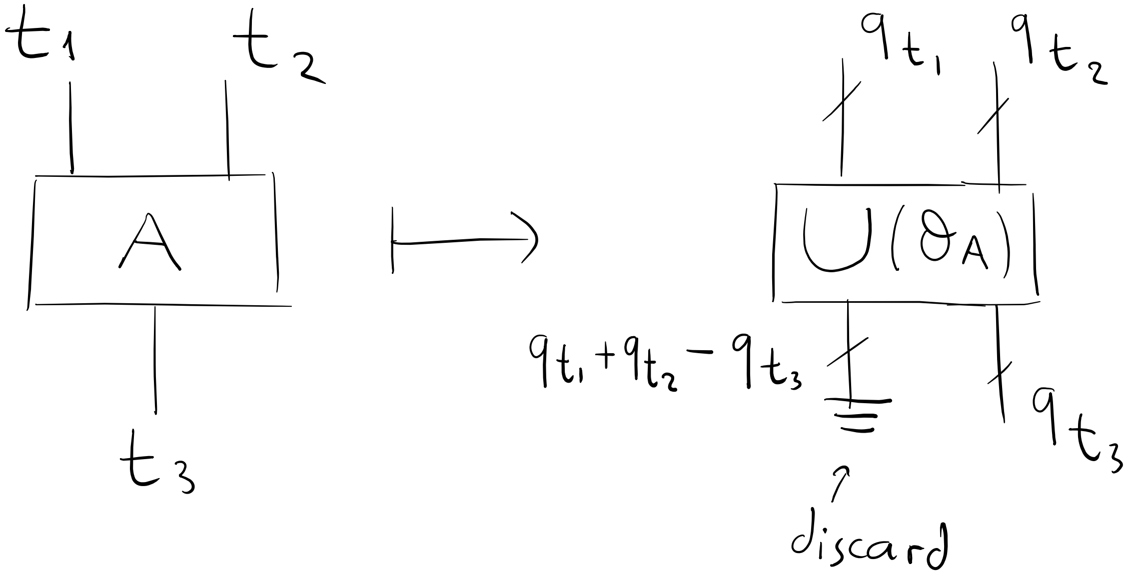
ex: $q_t = 1, \forall t$



QUANTUM LINGUISTIC PROCESSES

Type reductions are mapped to quantum processes.

CCG

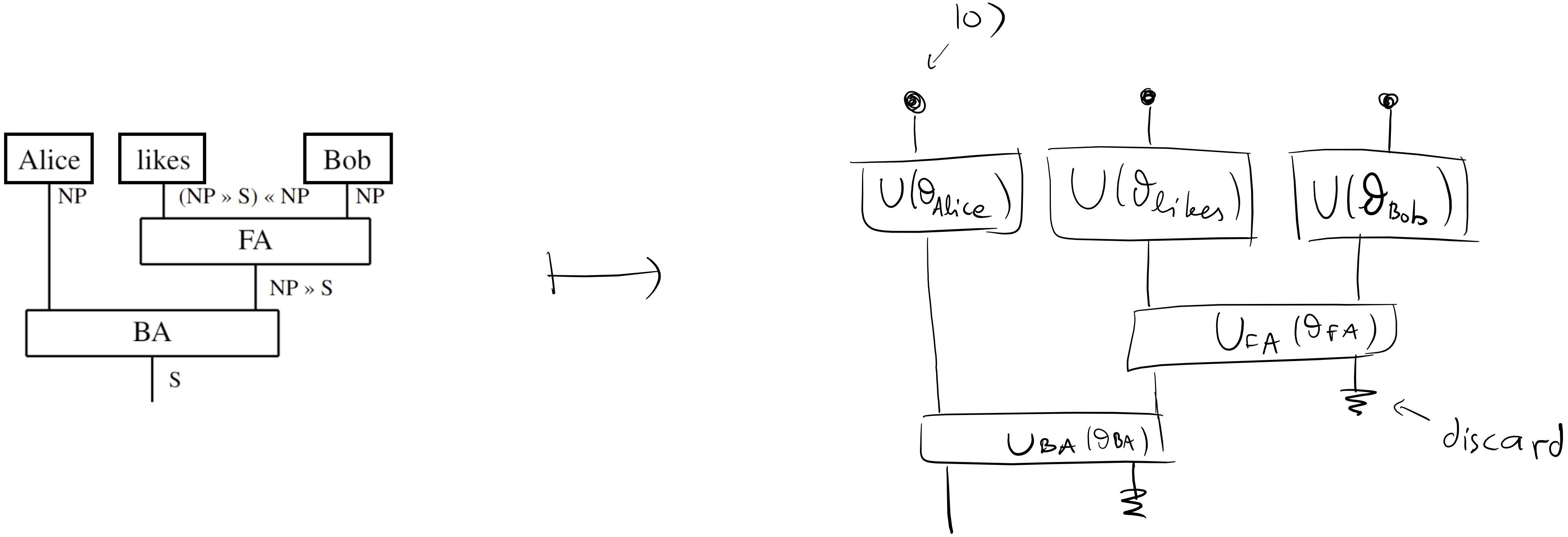


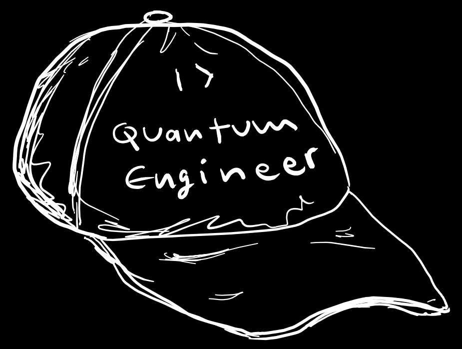
θ_A: trained in an NLP task

QUANTUM LINGUISTIC PROCESSES

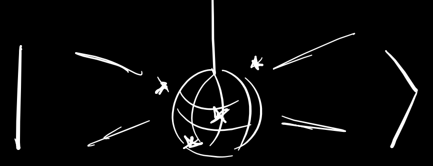
Lambeq
[2110.04236]

ex: $q_t = 1, \forall t$





EXPERIMENTS IN THE QUANTUM DISCO



[2005.02975]

DisCoPy: Monoidal Categories in Python

Giovanni de Felice, Alexis Toumi, Bob Coecke
Department of Computer Science, University of Oxford. Cambridge Quantum Computing Ltd.
{firstname.lastname}@cs.ox.ac.uk

We introduce DisCoPy, an open source toolbox for computing with monoidal categories. The library provides an intuitive syntax for defining string diagrams and monoidal functors. Its modularity allows the efficient implementation of computational experiments in the various applications of category theory where diagrams have become a lingua franca. As an example, we used DisCoPy to perform natural language processing on quantum hardware for the first time.

[2205.05190]

DisCoPy for the quantum computer scientist

Alexis Toumi[⋆†], Giovanni de Felice[†], and Richie Yeung[†]

† *Cambridge Quantum Computing Ltd.*
⋆ *Department of Computer Science, University of Oxford*

May 12, 2022

Abstract

DisCoPy (Distributional Compositional Python) is an open source toolbox for computing with string diagrams and functors. In particular, the diagram data structure allows to encode various kinds of quantum processes, with functors for classical simulation and optimisation, as well as compilation and evaluation on quantum hardware. This includes the ZX calculus and its many variants, the parameterised circuits used in quantum machine learning, but also linear optical quantum computing. We review the recent developments of the library in this direction, making DisCoPy a toolbox for the quantum computer scientist.

https://github.com/oxford-quantum-group/discopy
pip install discopy

[2110.04236]

https://github.com/CQCL/lambeq
pip install lambeq

lambeq: An Efficient High-Level Python Library for Quantum NLP

Dimitri Kartsaklis Ian Fan Richie Yeung Anna Pearson
Robin Lorenz Alexis Toumi Giovanni de Felice
Konstantinos Meichanetzidis Stephen Clark Bob Coecke

Cambridge Quantum Computing
17 Beaumont Street, Oxford, OX1 2NA, UK

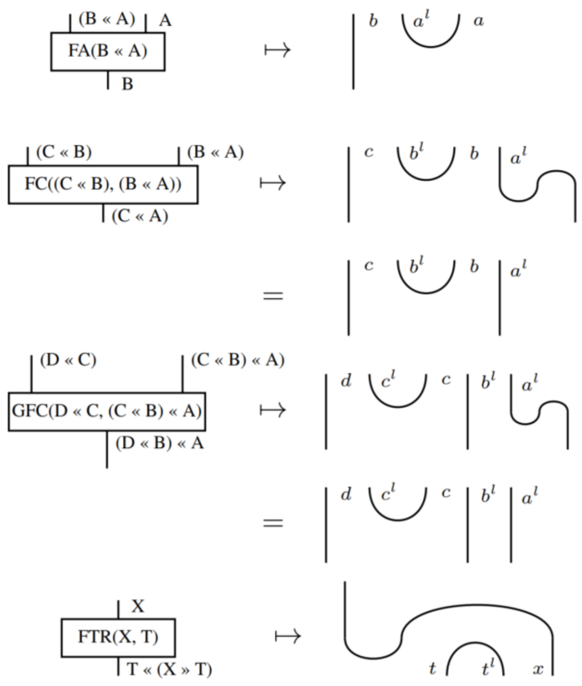
{dimitri.kartsaklis;ian.fan;richie.yeung;anna.pearson;
robin.lorenz;alexis.toumi;giovanni.defelice;
kmei;steve.clark;bob.coecke}@cambridgequantum.com

Abstract

We present lambeq, the first high-level Python library for Quantum Natural Language Processing (QNLP). The toolkit offers a detailed hierarchy of modules and classes implementing all stages of a pipeline for converting sentences to string diagrams, tensor networks, and quantum circuits ready to be used on a quantum computer. lambeq supports syntactical parsing, rewriting and simplification of string diagrams, ansatz creation and manipulation, as well as a number of compositional models for preparing quantum-friendly representations of sentences, employing various degrees of syntax sensitivity. We present the generic architecture and we describe the most important modules in detail, demonstrating the usage with illustrative examples. Further, we test the toolkit in practice by using it in order to perform a number of experiments on simple NLP tasks, implementing both classical and quantum pipelines, with success.

Online demo:
https://qnlp.cambridgequantum.com/generate.html

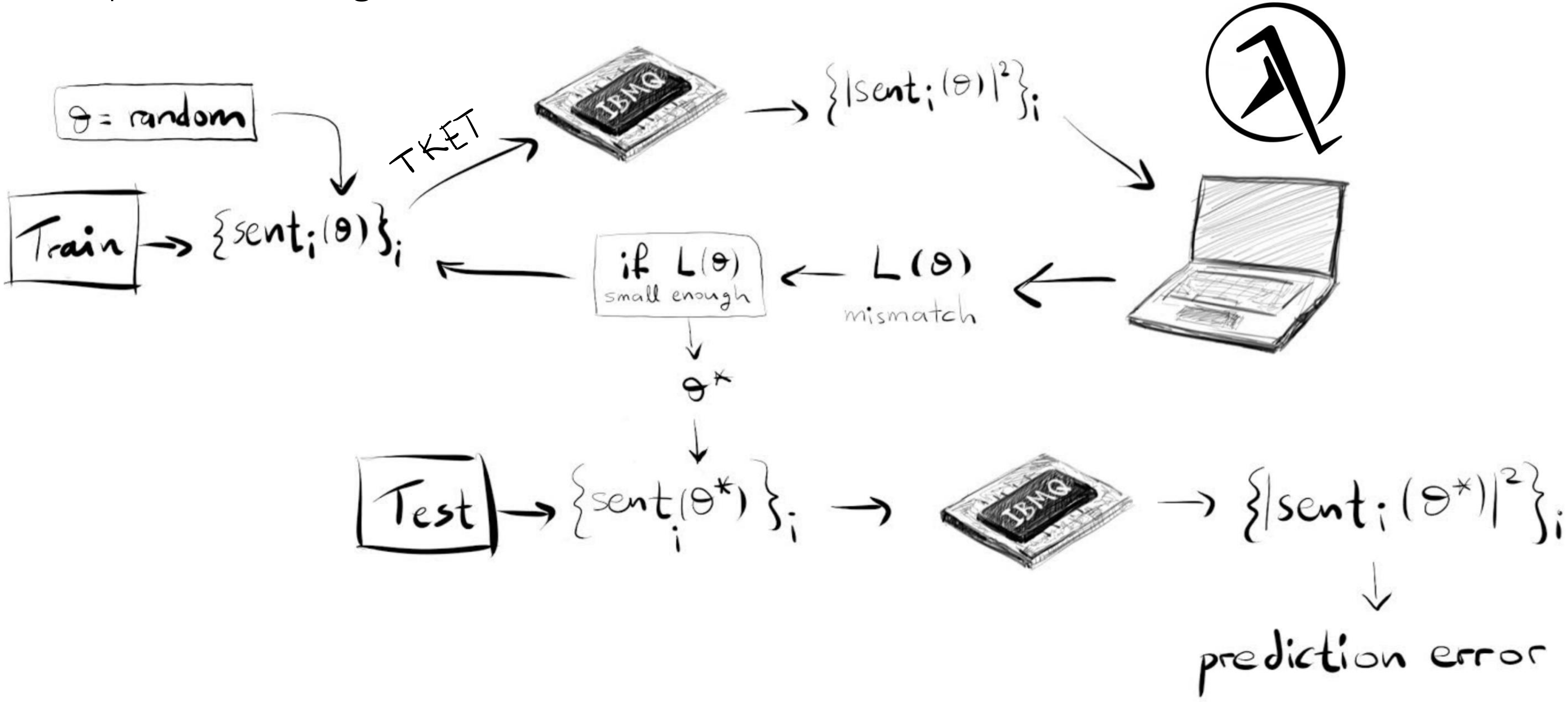
λambeq enables the automatic deployment of large-scale compositional language models



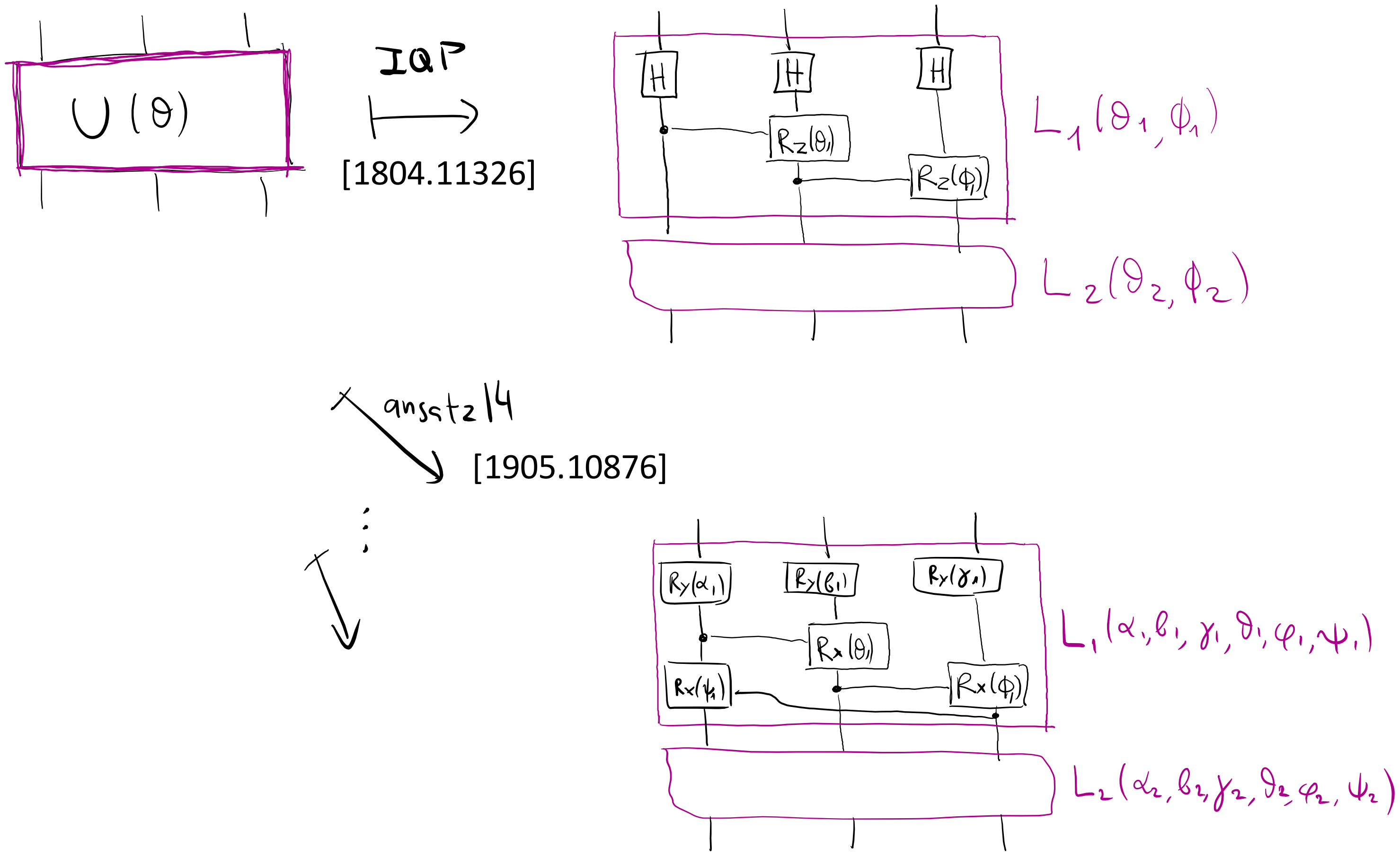
[2105.07720]

QUANTUM NATURAL LANGUAGE PROCESSING

Supervised learning



ANASAETZE



U(θ)
IQP
[1804.11326]
H
H
H
Rz(θ1)
Rz(ϕ1)
L1(θ1, ϕ1)
L2(θ2, ϕ2)
ansatz 14
[1905.10876]
Ry(α1)
Ry(β1)
Ry(γ1)
Rx(θ1)
Rx(ψ1)
Rx(ϕ1)
L1(α1, β1, γ1, θ1, φ1, ψ1)
L2(α2, β2, γ2, θ2, φ2, ψ2)

TOY DATA: CLASSICAL SIM

[2012.03756]

Pregroup model

|V| = 7

15 train sents
15 test sents

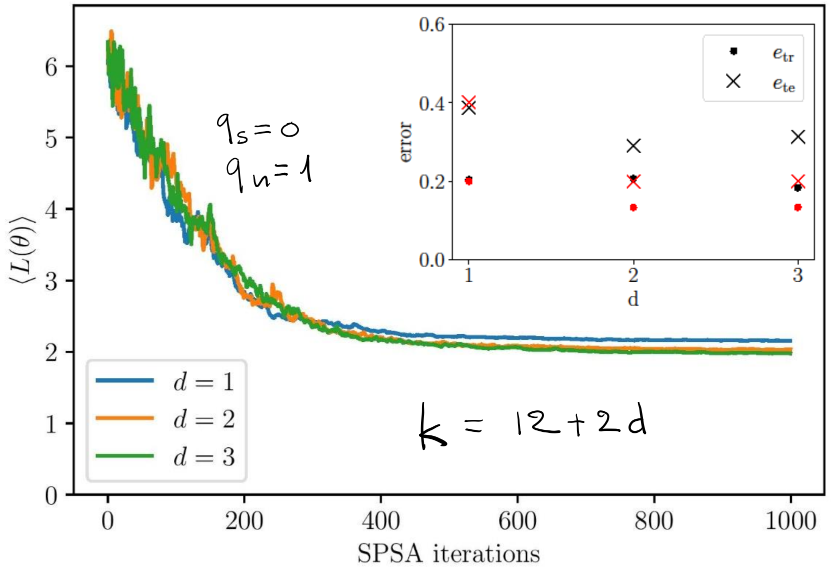
('Dude who loves Walter bowls', 1),
('Dude bowls', 1),
('Dude annoys Walter', 0),
('Walter who abides bowls', 0),
('Walter loves Walter', 1),
('Walter annoys Dude', 1),
('Walter bowls', 1),
('Walter abides', 0),
('Dude loves Walter', 1),
('Dude who bowls abides', 1),
('Walter who bowls annoys Dude', 1),
('Dude who bowls bowls', 1),
('Dude who abides abides', 1),
('Dude annoys Dude who bowls', 0),
…

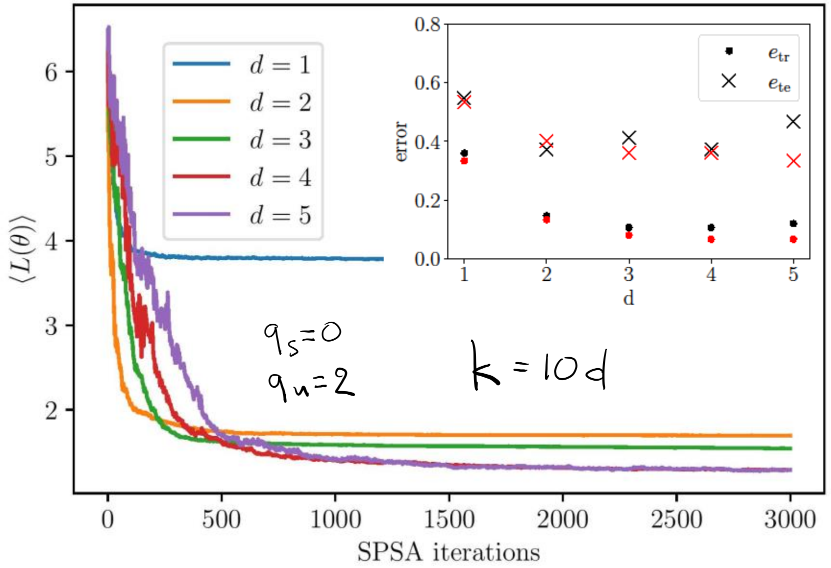
$$L(\theta) = \sum_s \left(l_s^{pr}(\theta) - l_s^{data}\right)^2$$

IQP ansatz

$\langle L(\theta) \rangle$ over 20 realisations

basinhopping over Nelder-Mead





Note:

Intra-sentence correlations are 'quantum': due to grammar
Inter-sentence correlations are 'classical': due to shared words

TOY DATA: QUANTUM RUN

[2012.03756]

Pregroup model

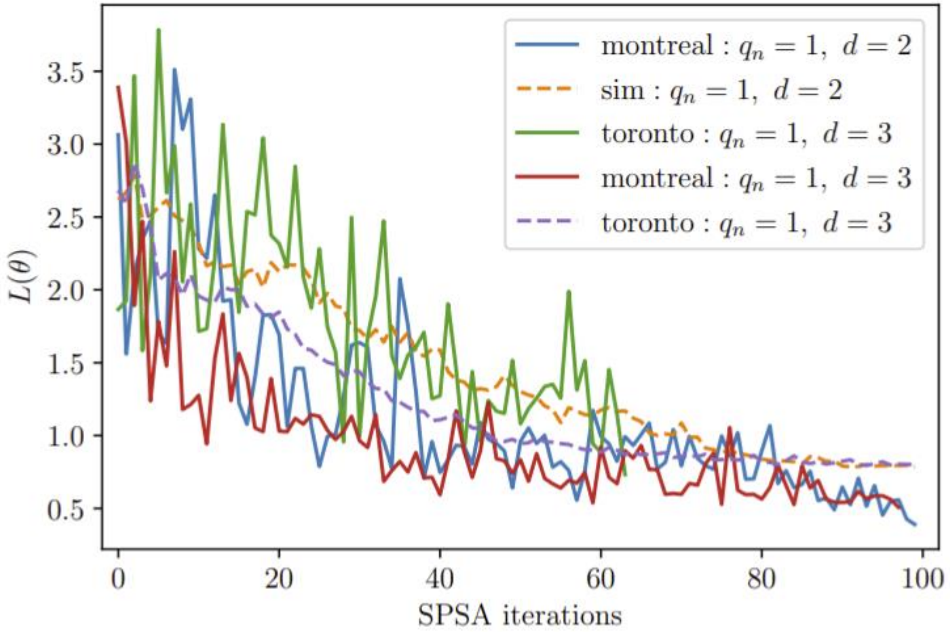
$q_n = 1, q_s = 0$, IQP ansatz

|V|= 5

8 train sents
8 test sents

False:
(Juliet kills Romeo who dies, 0)
(Romeo kills Juliet, 0)
(Romeo who kills Juliet dies, 0)
...

True:
(Juliet dies, 1)
(Romeo who dies loves Juliet, 1)
(Romeo who kills Romeo dies, 1)
...



Ibmq_montreal

$$L(\theta) = \sum_s \left(l_s^{pr}(\theta) - l_s^{data} \right)^2$$

TOY DATA: CLASSICAL SIM

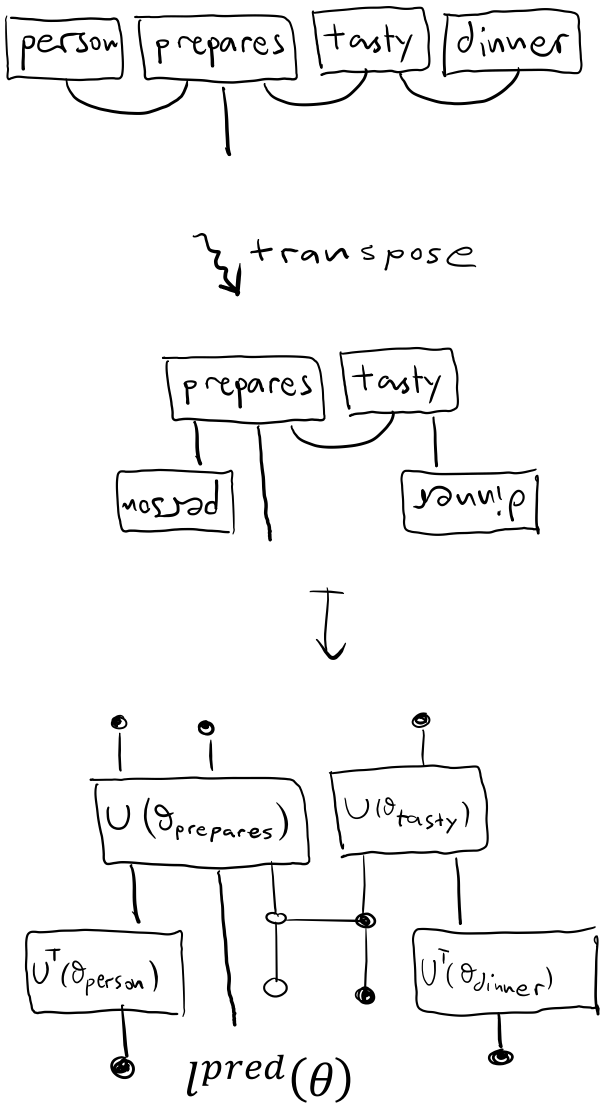
Pregroup model

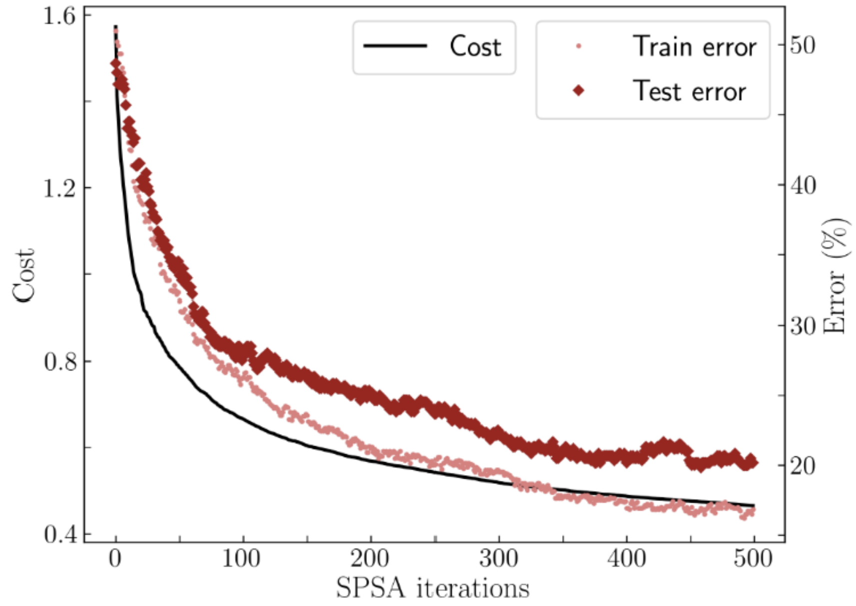
$q_n = 1, q_s = 1$, IQP Ansatz, $d = 1$

Train sents: 70
Dev sents: 30
Test sents: 30

Cooking:
(Skilful man prepares sauce, 0)
(Woman cooks tasty meal , 0)
(Skilful person prepares meal, 0)
…

Technology:
(Skilful woman debugs program , 1)
(Man prepares useful application, 1)
(Person debugs useful software, 1)
…

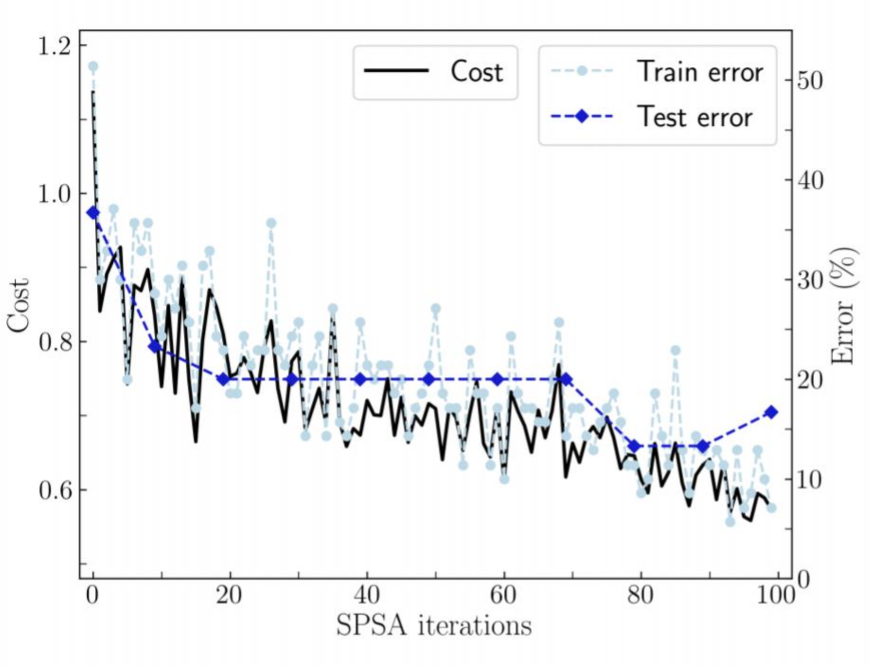




$$Cost(\theta) = \sum_s l_s^{data} \log l_s^{pred}(\theta) + (1 - l_s^{data}) \log (1 - l_s^{pred}(\theta))$$

TOY DATA: QUANTUM RUN

[2102.12846]



Ibmq_bogota

Same topic classification task.

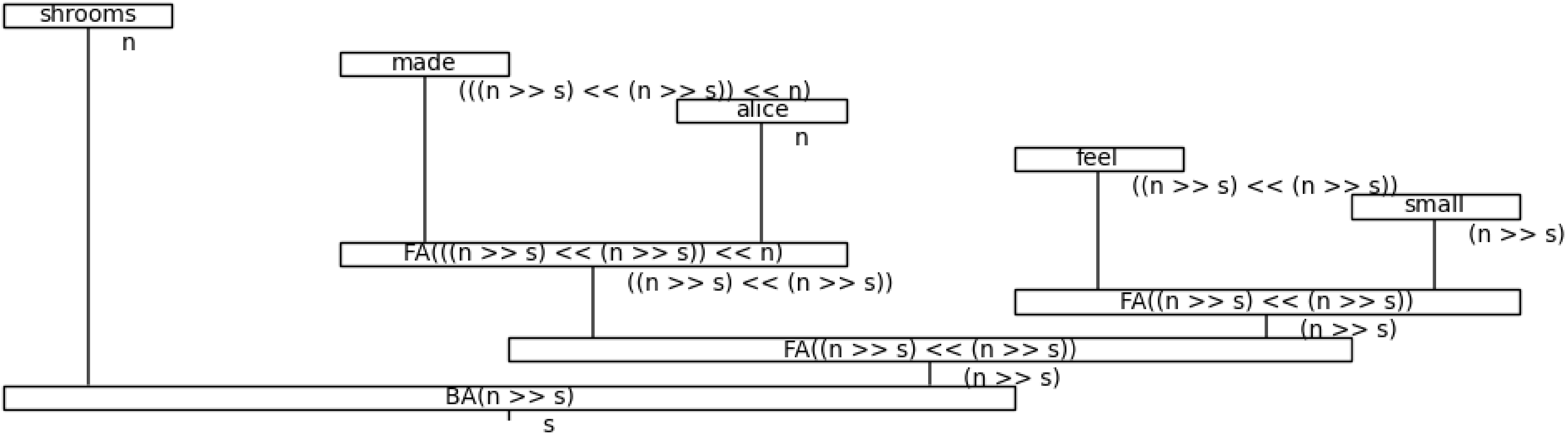
Train classically,
not even considering noise.

Then Test on H1 (12 qbts).

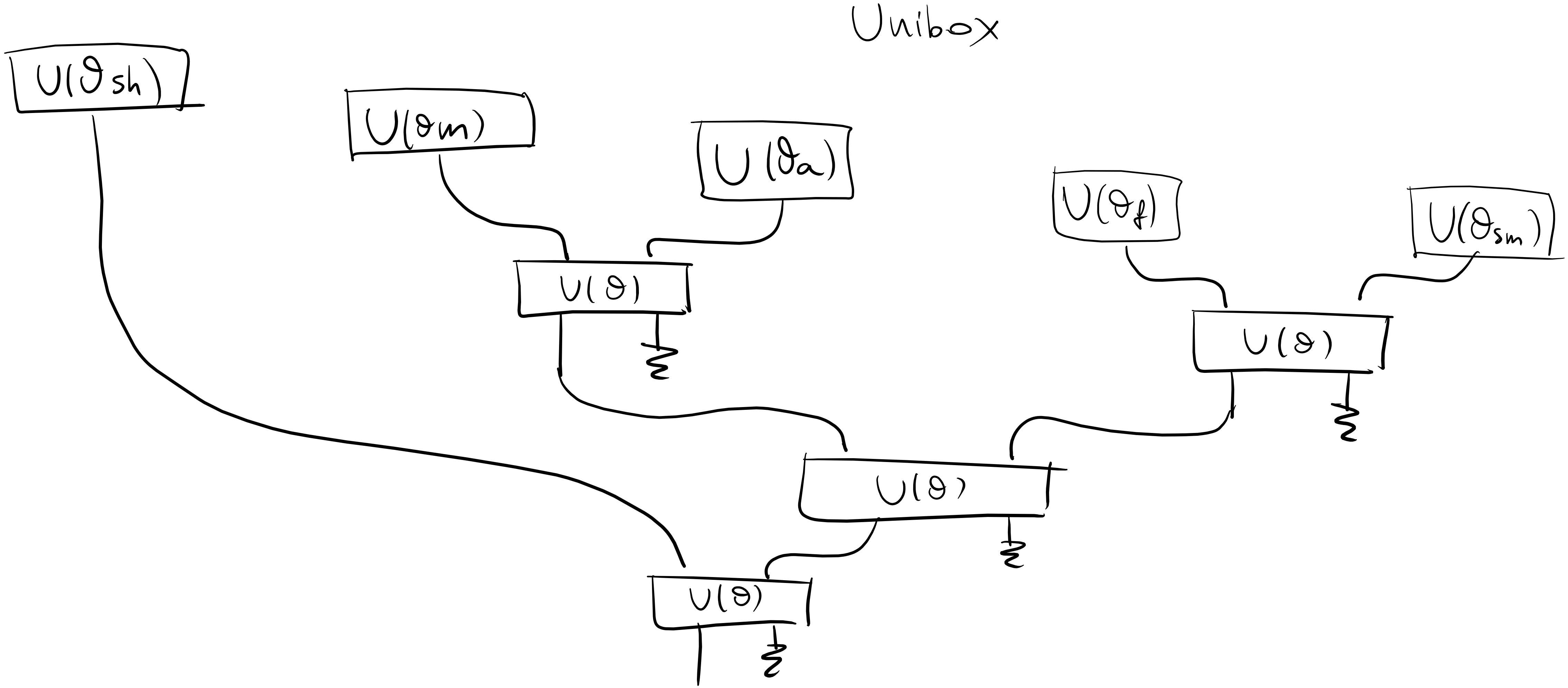
Acc 97%.

QUANTUM CCG TREE SPECIES

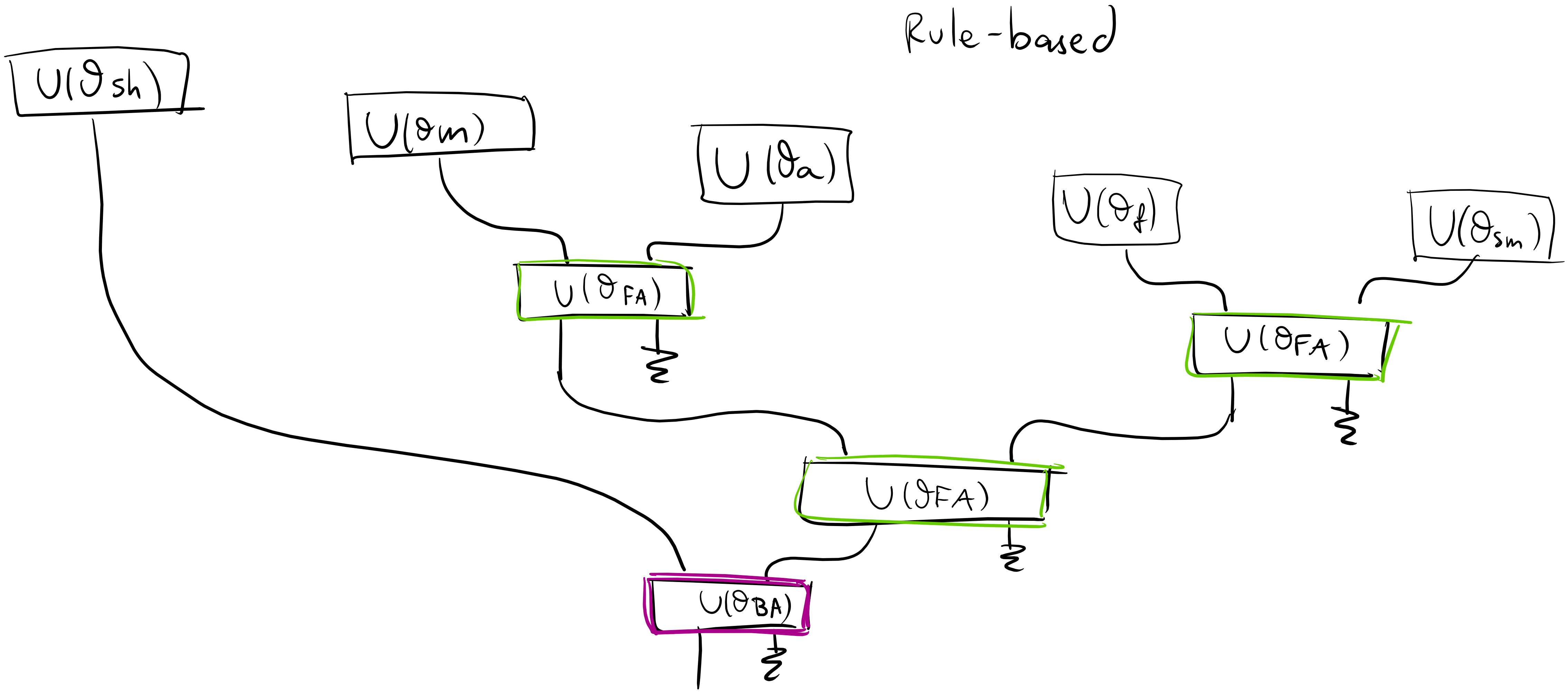
Lambeq's tree-reader
[2110.04236]



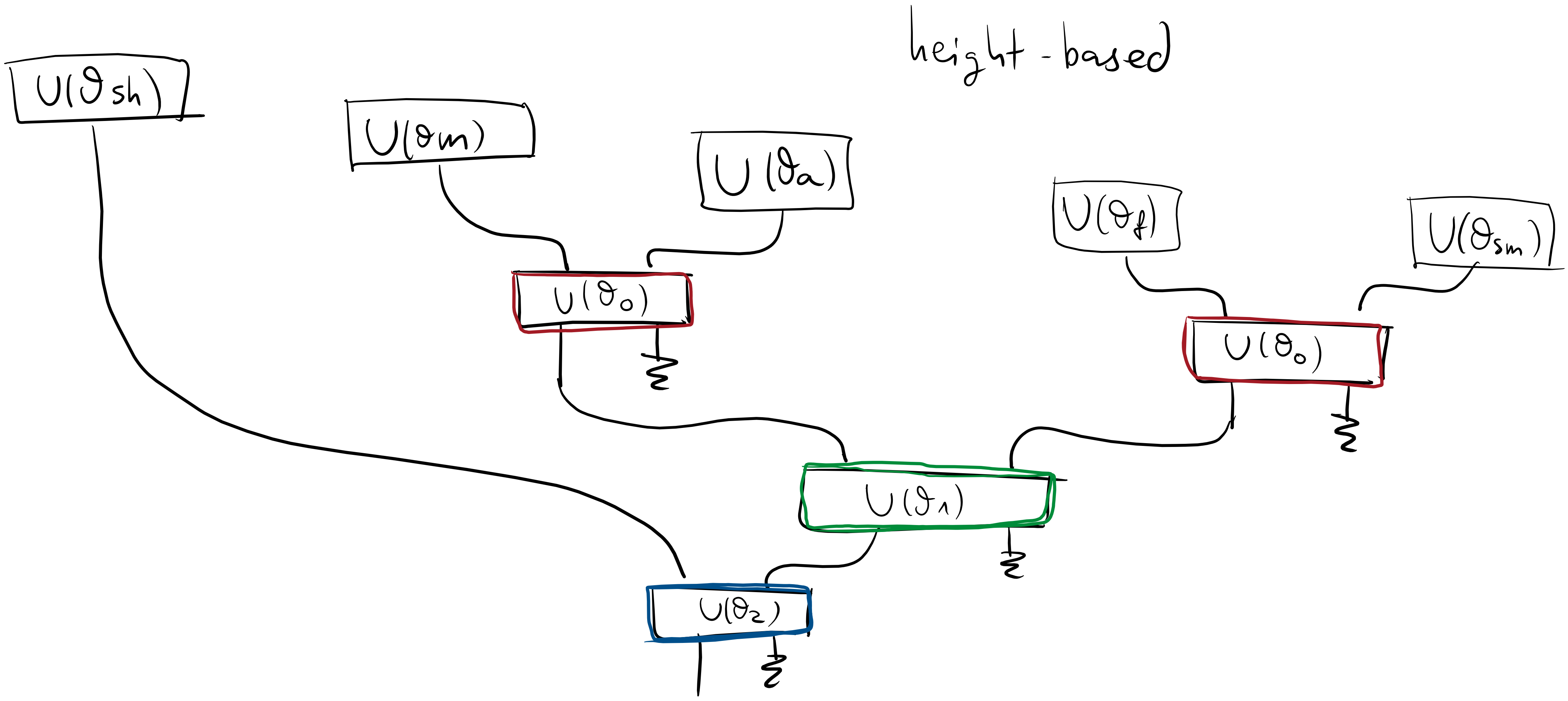
QUANTUM TREE SPECIES



QUANTUM TREE SPECIES



QUANTUM TREE SPECIES



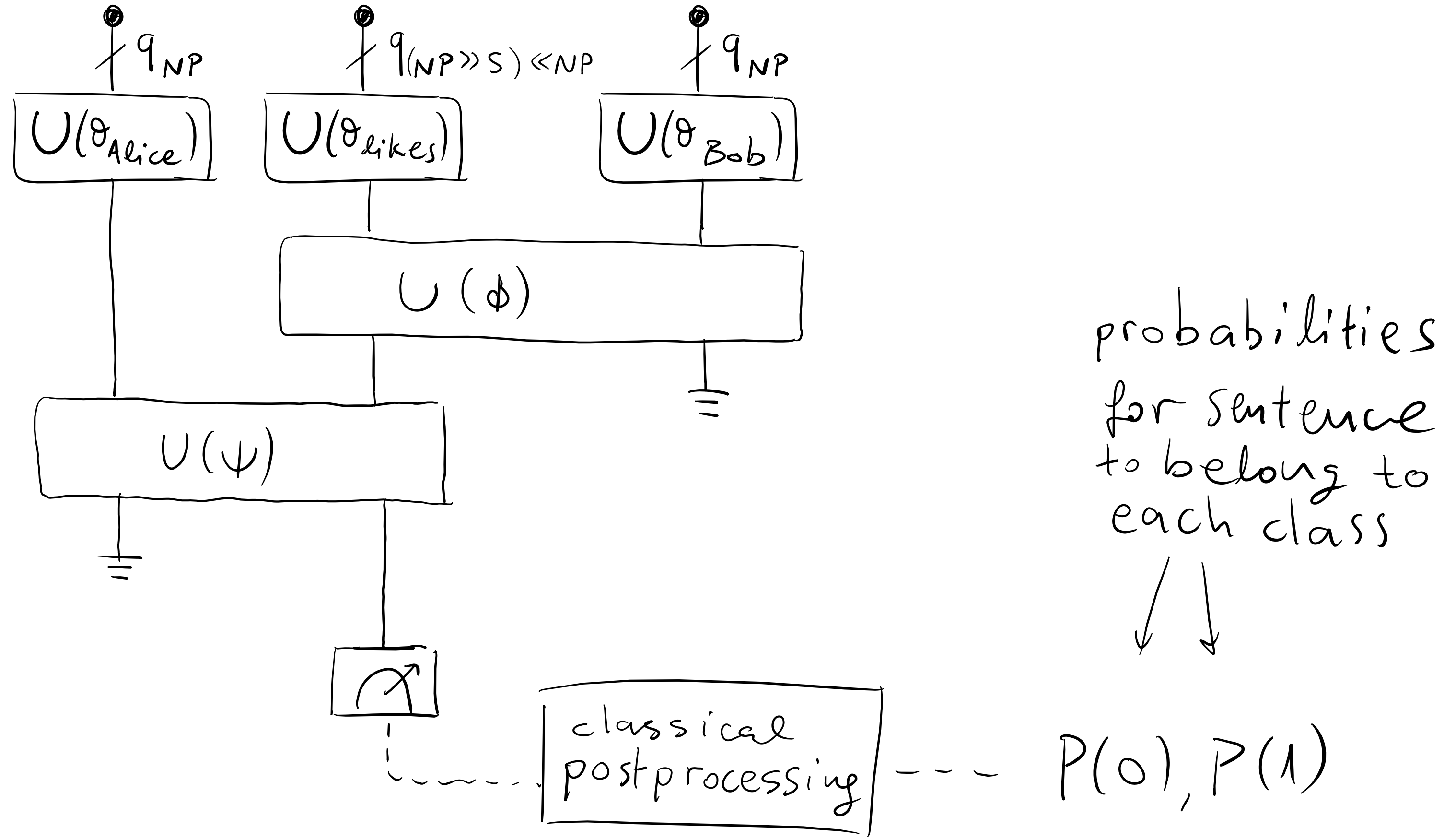
GETTING A BINARY LABEL OUT

" Syntactic QCNN "

[1810.03787]

no barren plateaus for tree qcircs

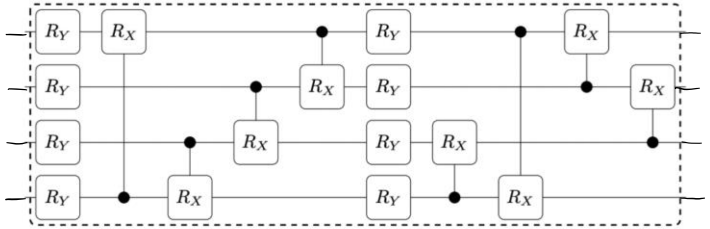
[2011.06258]



TOY DATA: CLASSICAL SIM

ansatz14

$U(\theta) =$



word states trained in-task

Dataset:

Food:

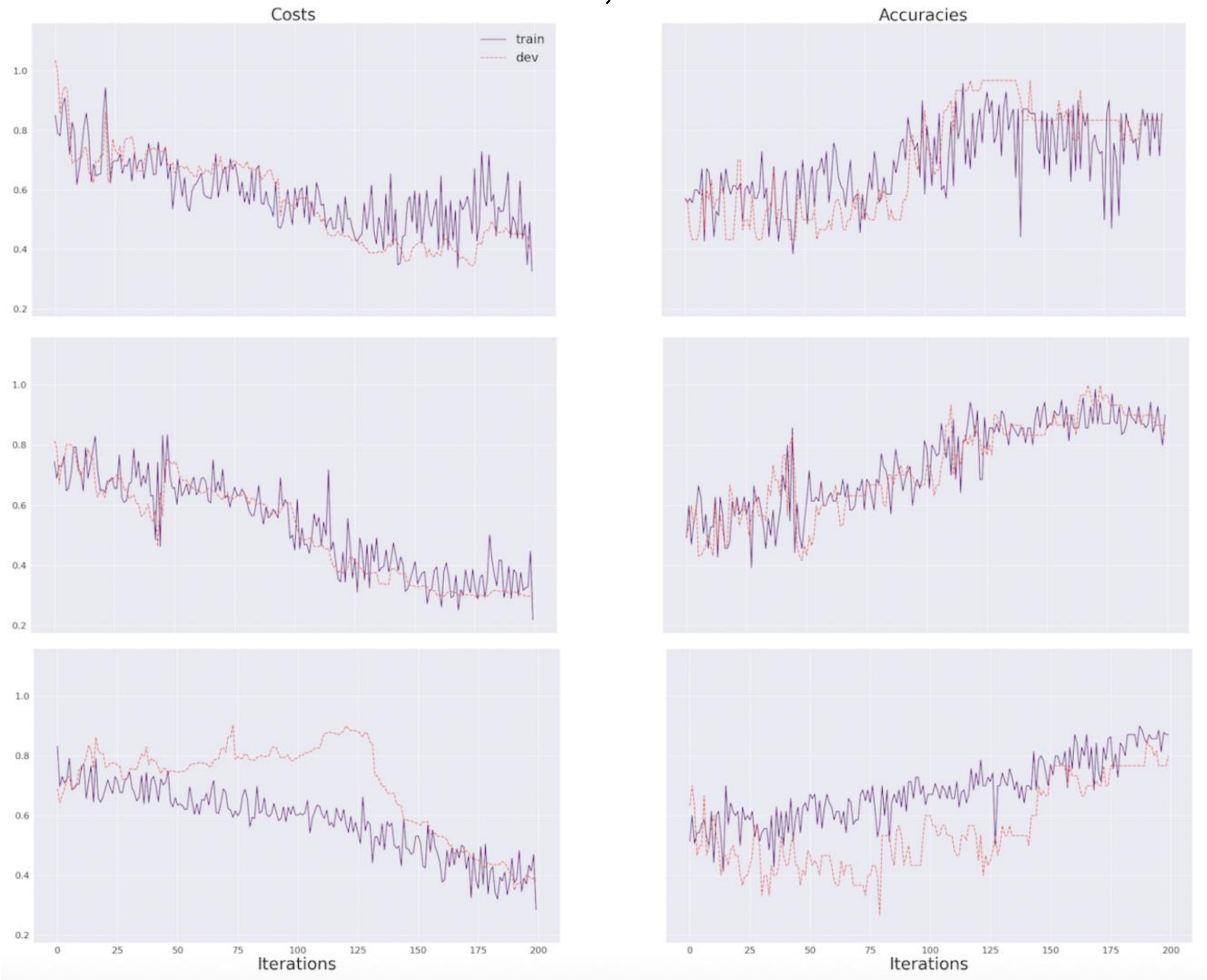
(Skilful man prepares sauce, 0)
(Woman cooks tasty meal , 0)
(Skilful person prepares meal, 0)
. . .

IT:

(Skilful woman debugs program , 1)
(Man prepares useful application, 1)
(Person debugs useful software, 1)
. . .

[2102.12846]

$q_t = 1, \forall t$



REAL-WORLD DATA: CLASSICAL SIM

Tree species: unibox
t : 3qb ∀t
ansatz: 2-layer IQP
dataset: rt-polarity
Train: 216 sents
Test: 35 sents

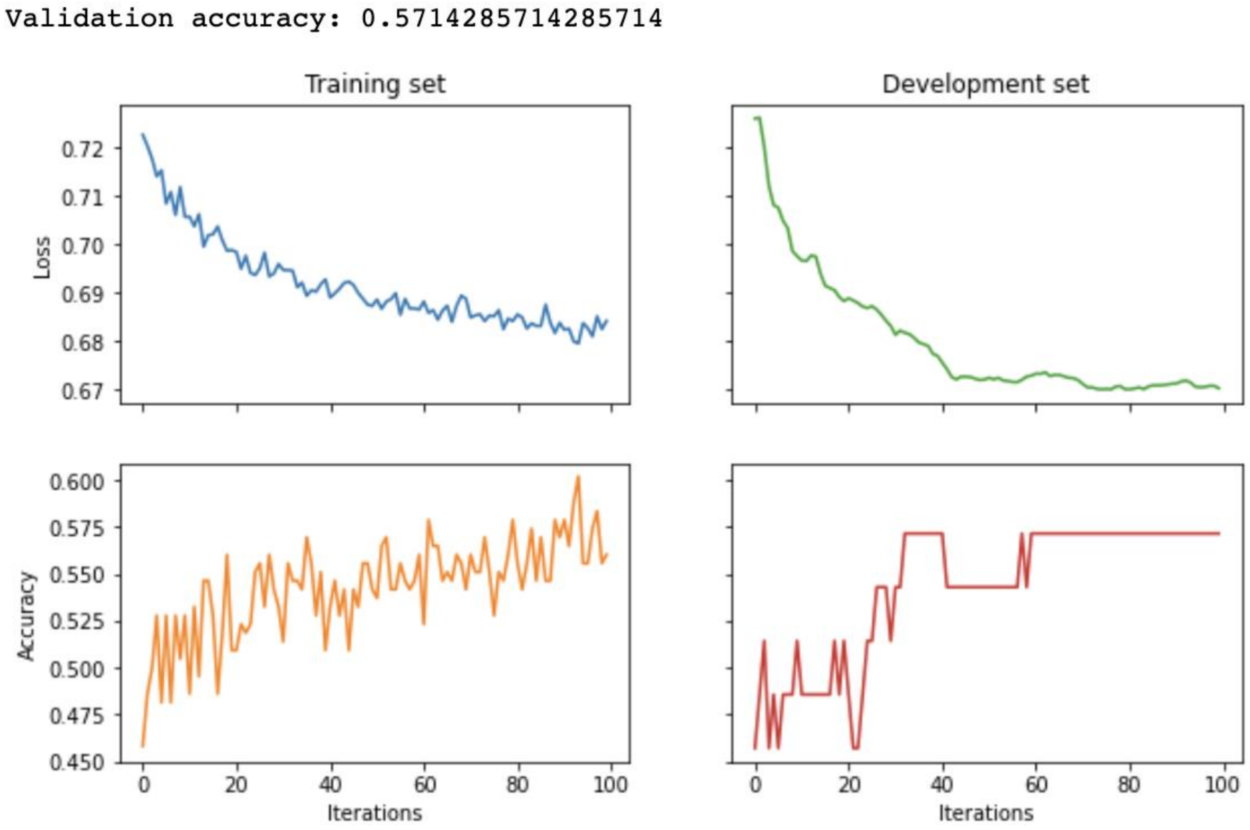
word states
trained in-task

“A masterful film from a master filmmaker, unique in its deceptive grimness, compelling in its fatalist worldview.”

“if you love reading and/or poetry , then by all means check it out . you'll probably love it .”

“… stumbles over every cheap trick in the book trying to make the outrage come even easier “

“the only way to tolerate this insipid , brutally clueless film might be with a large dose of painkillers “

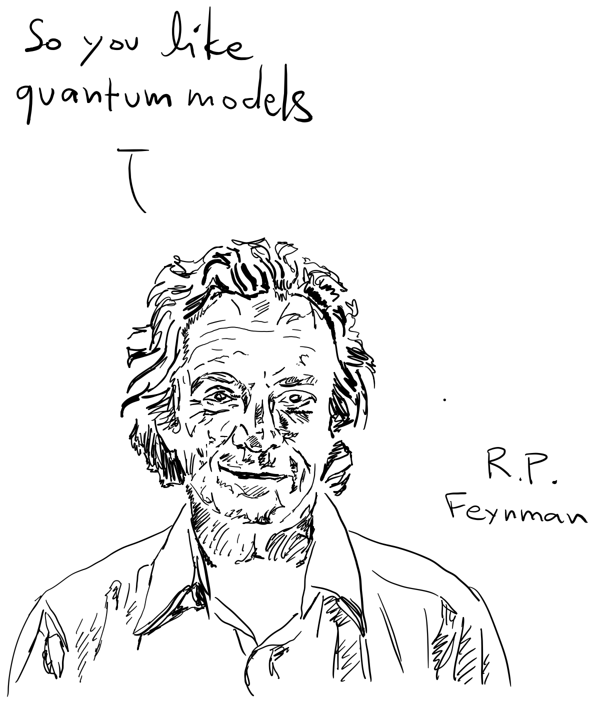


Sentence composition

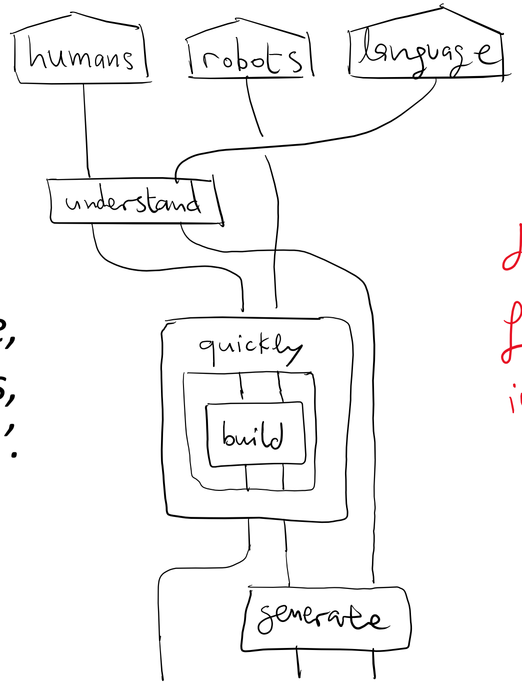
[1904.03478]

Quantum DisCoCirc: sentence composition; satisfies **specific desiderata** of compositionality.

Bob Coecke: “*get quadvantage a la quantum simulation*”.
So **IF** this model works well for some tasks, one would need a QC to run it.



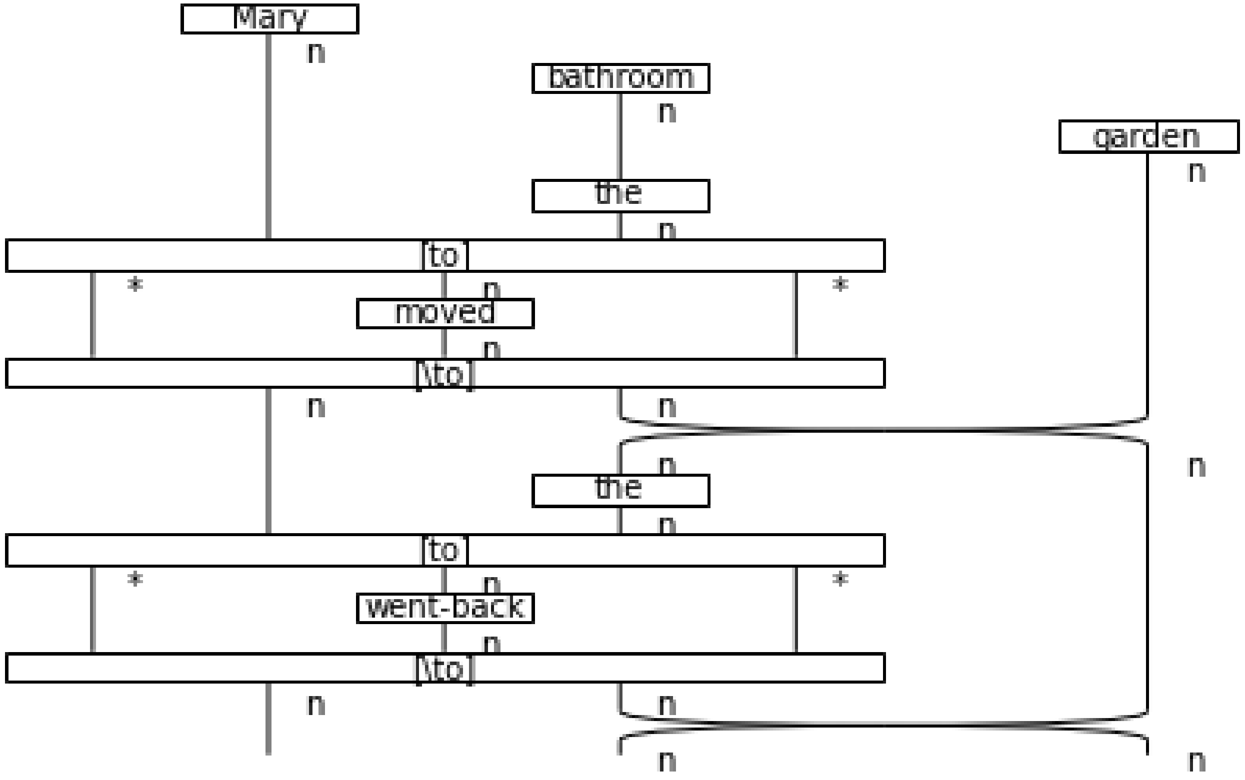
`Humans understand language, humans quickly build robots, robots generate language’.



define functor into qcircs

text2circ

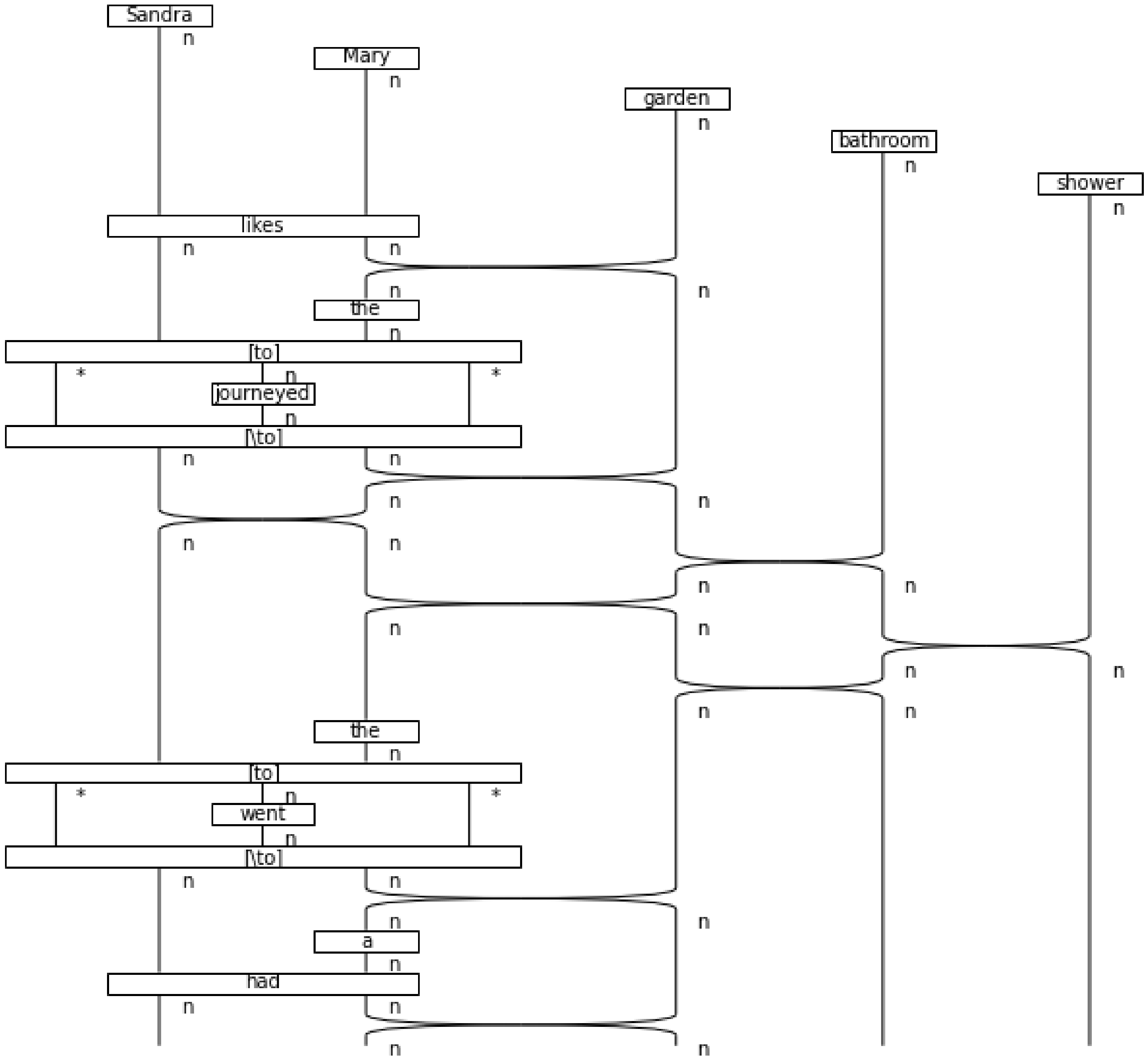
Mary moved to the bathroom.
Mary went back to the garden.



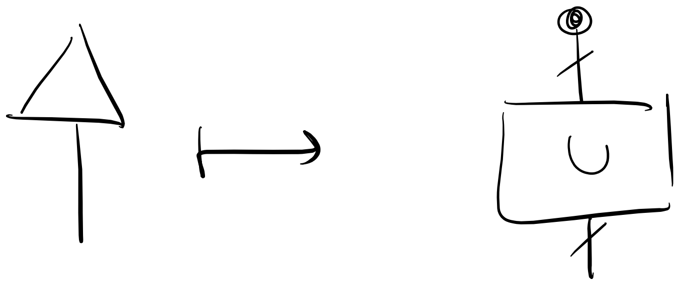
text2circ

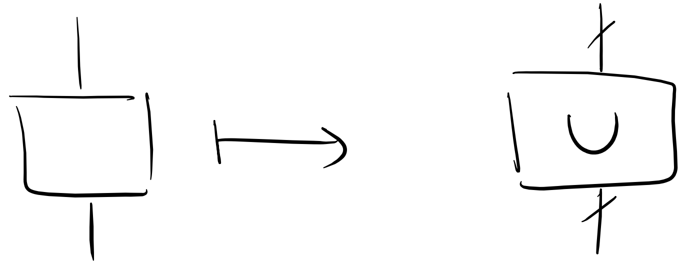
Sandra likes Mary.
Sandra journeyed to the garden.
Mary went to the bathroom.
Mary had a shower

$\longmapsto$

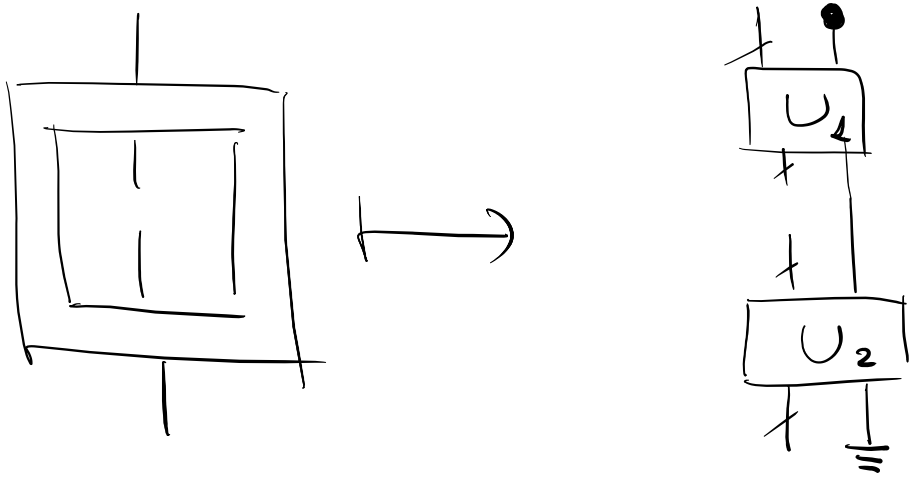


text2qcirc

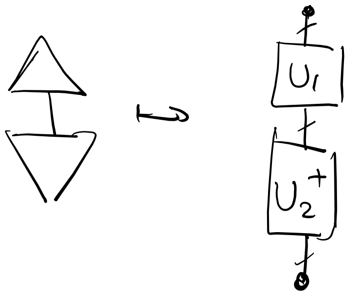


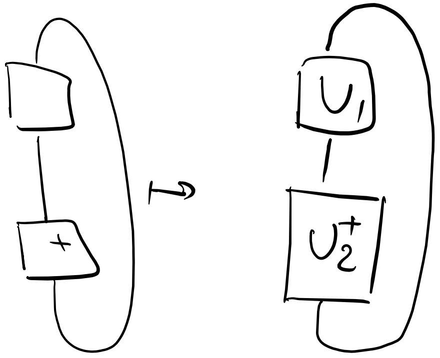


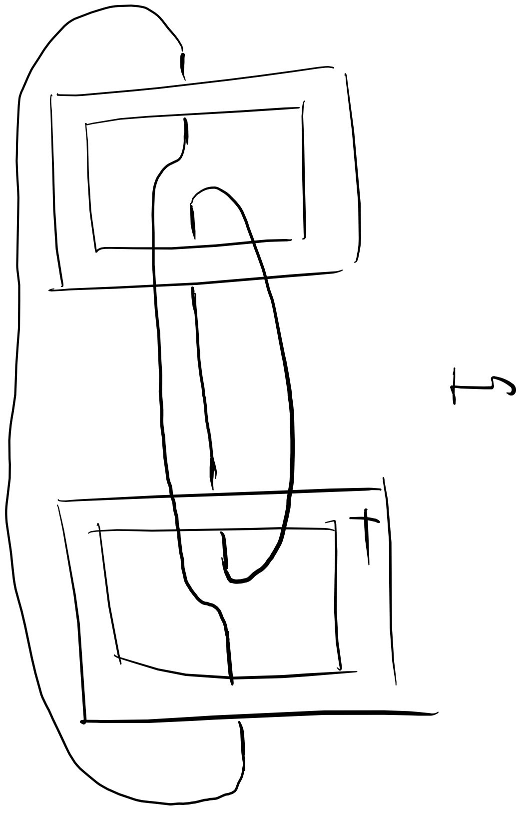
"combs" :

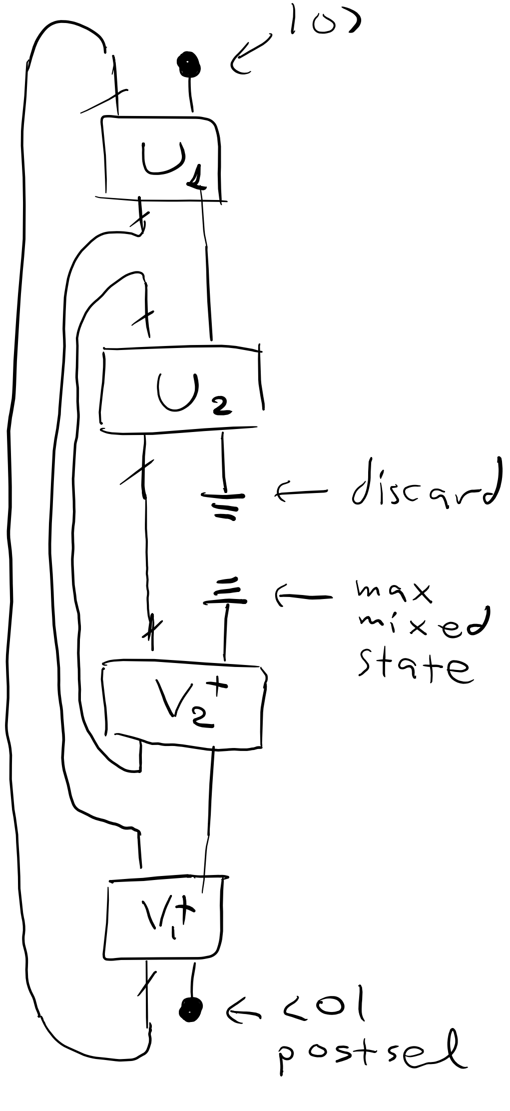


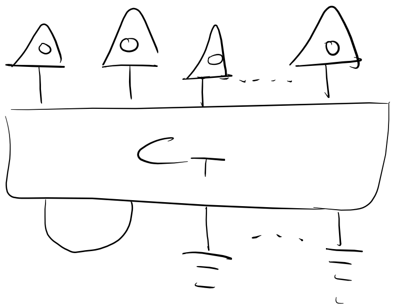
Overlaps of higher-oder processes





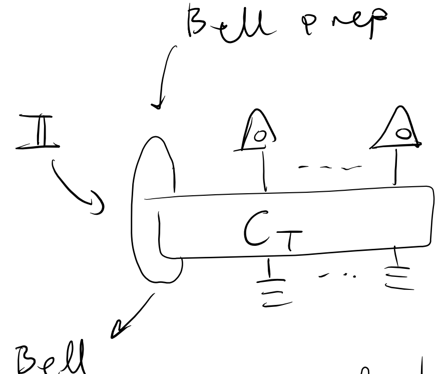




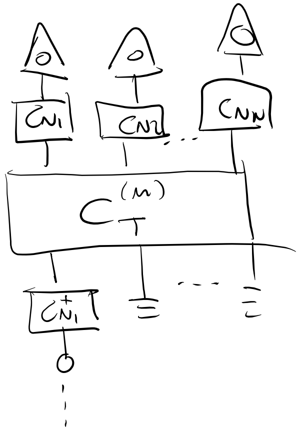


one Bell effect

similarity between nouns N_1, N_2
after the modifiers acted on them

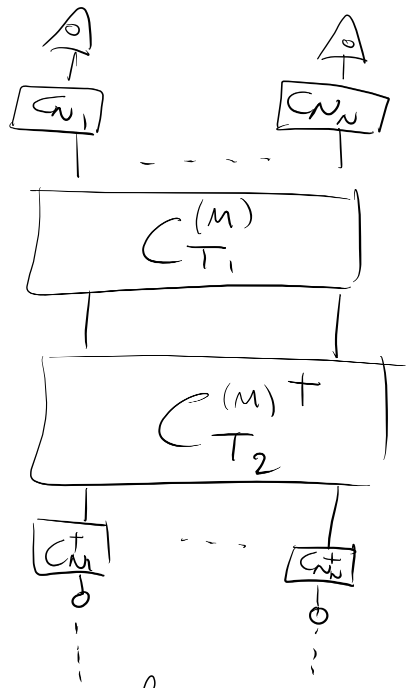


degree of change
between any input and output
in that noun position in the text



freq $\langle 0|$ ~ degree of change of noun N_1 (sampling)
"character arc"

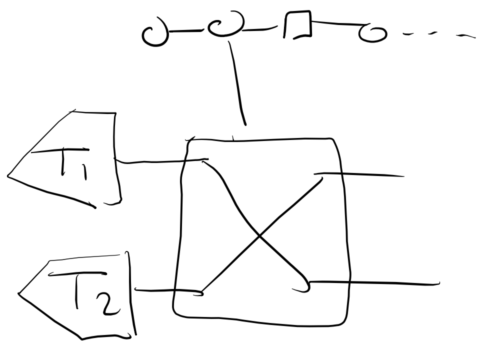
text similarity



another text involving same nouns

freq $\langle 0|$ ~ degree of similarity of how T_1, T_2 affect the same nouns

or do swap test

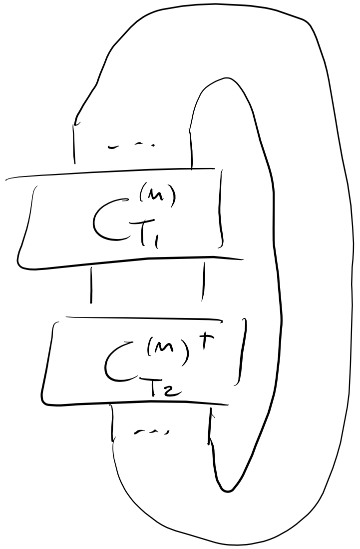


$P(0) \sim |\langle T_1|T_2\rangle|^2$

additive error $\varepsilon \sim \frac{1}{\sqrt{\text{samples}}}$

$$\left\| P(0) - |\langle T_1|T_2\rangle|^2 \right\| \leqslant \varepsilon$$

if some nouns are different between T_1, T_2, you can't compare the texts.
But maybe if they share some nouns, we can discard the ones that are not common



if $C_{T_1}^{(M)} C_{T_2}^{(M)\dagger} \sim \mathbb{I}$

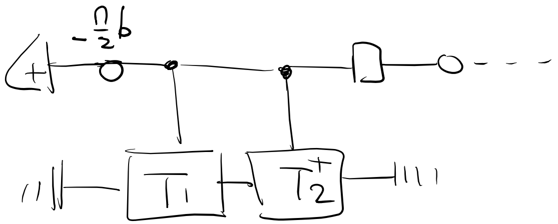
then texts are the same process on <u>any</u> input means

QMA-hard (non-$\mathbb{I}$ check)

(exp-many samples worst case)

DQC$_1$

for trace estimation



$P(0) \sim \begin{cases} \mathrm{Re}\,\mathrm{tr}\, T_1 T_2^\dagger, & b=0 \\ \mathrm{Im}\,\mathrm{tr}\, T_1 T_2^\dagger, & b=1 \end{cases}$

additive error $\varepsilon \sim \frac{1}{\sqrt{\text{samples}}}$

Join the QNLP Discord server!

https://discord.gg/zW9zHNpdnN

Check out opportunities at Quantinuum!

quantinuum.com/careers