An Introduction
to
Quantum Natural Language Processing
(QNLP)

Part 4 :

Quantum Natural Language Processing

Outline of the lecture

Outline of the lecture

❖ Introducing Quantum Natural Language Processing

Outline of the lecture

- ❖ Introducing Quantum Natural Language Processing
- ❖ Distributional Word Representation

Outline of the lecture

- Introducing Quantum Natural Language Processing
- Distributional Word Representation
- Compositionality of Grammar

Outline of the lecture

- ❖ Introducing Quantum Natural Language Processing
- ❖ Distributional Word Representation
- ❖ Compositionality of Grammar
- ❖ QNLP Basics – Adjective & Noun

Outline of the lecture

- ❖ Introducing Quantum Natural Language Processing
- ❖ Distributional Word Representation
- ❖ Compositionality of Grammar
- ❖ QNLP Basics – Adjective & Noun
- ❖ Subject-Verb-Object Sentence

Outline of the lecture

- ❖ Introducing Quantum Natural Language Processing
- ❖ Distributional Word Representation
- ❖ Compositionality of Grammar
- ❖ QNLP Basics – Adjective & Noun
- ❖ Subject-Verb-Object Sentence
- ❖ DisCoCat Algorithm

Outline of the lecture

- ❖ Introducing Quantum Natural Language Processing
- ❖ Distributional Word Representation
- ❖ Compositionality of Grammar
- ❖ QNLP Basics – Adjective & Noun
- ❖ Subject-Verb-Object Sentence
- ❖ DisCoCat Algorithm
- ❖ String Diagram to ZX Quantum Circuit

Outline of the lecture

- ❖ Introducing Quantum Natural Language Processing
- ❖ Distributional Word Representation
- ❖ Compositionality of Grammar
- ❖ QNLP Basics – Adjective & Noun
- ❖ Subject-Verb-Object Sentence
- ❖ DisCoCat Algorithm
- ❖ String Diagram to ZX Quantum Circuit
- ❖ Introducing λambeq & its Features

Outline of the lecture

- ❖ Introducing Quantum Natural Language Processing
- ❖ Distributional Word Representation
- ❖ Compositionality of Grammar
- ❖ QNLP Basics – Adjective & Noun
- ❖ Subject-Verb-Object Sentence
- ❖ DisCoCat Algorithm
- ❖ String Diagram to ZX Quantum Circuit
- ❖ Introducing λambeq & its Features
- ❖ QNLP Training Process

Outline of the lecture

- ❖ Introducing Quantum Natural Language Processing
- ❖ Distributional Word Representation
- ❖ Compositionality of Grammar
- ❖ QNLP Basics – Adjective & Noun
- ❖ Subject-Verb-Object Sentence
- ❖ DisCoCat Algorithm
- ❖ String Diagram to ZX Quantum Circuit
- ❖ Introducing λambeq & its Features
- ❖ QNLP Training Process
- ❖ Sentence Classification with λambeq – Classical & Quantum

Outline of the lecture

- ❖ Introducing Quantum Natural Language Processing
- ❖ Distributional Word Representation
- ❖ Compositionality of Grammar
- ❖ QNLP Basics – Adjective & Noun
- ❖ Subject-Verb-Object Sentence
- ❖ DisCoCat Algorithm
- ❖ String Diagram to ZX Quantum Circuit
- ❖ Introducing λambeq & its Features
- ❖ QNLP Training Process
- ❖ Sentence Classification with λambeq – Classical & Quantum
- ❖ Potential Applications of QNLP

Outline of the lecture

- Introducing Quantum Natural Language Processing
- Distributional Word Representation
- Compositionality of Grammar
- QNLP Basics – Adjective & Noun
- Subject-Verb-Object Sentence
- DisCoCat Algorithm
- String Diagram to ZX Quantum Circuit
- Introducing λambeq & its Features
- QNLP Training Process
- Sentence Classification with λambeq – Classical & Quantum
- Potential Applications of QNLP
- Future Directions for Research

Quantum Natural Language Processing (QNLP)

Quantum Natural Language Processing (QNLP)

- ❖ DisCoCat (Distributional Compositional Categorical) brand of NLP

Quantum Natural Language Processing (QNLP)

- DisCoCat (Distributional Compositional Categorical) brand of NLP

- Combines linguistic meaning and linguistic structure

Quantum Natural Language Processing (QNLP)

- ❖ DisCoCat (Distributional Compositional Categorical) brand of NLP
- ❖ Combines linguistic meaning and linguistic structure
- ❖ Originated from the Categorical Quantum Mechanics (CQM)

Quantum Natural Language Processing (QNLP)

- ❖ DisCoCat (Distributional Compositional Categorical) brand of NLP
- ❖ Combines linguistic meaning and linguistic structure
- ❖ Originated from the Categorical Quantum Mechanics (CQM)

An important feature of QNLP is its completely pictorial!

Quantum Natural Language Processing (QNLP)

- ❖ DisCoCat (Distributional Compositional Categorical) brand of NLP
- ❖ Combines linguistic meaning and linguistic structure
- ❖ Originated from the Categorical Quantum Mechanics (CQM)

An important feature of QNLP is its completely pictorial!

Uses three theories

Quantum Natural Language Processing (QNLP)

- ❖ DisCoCat (Distributional Compositional Categorical) brand of NLP

- ❖ Combines linguistic meaning and linguistic structure

- ❖ Originated from the Categorical Quantum Mechanics (CQM)

An important feature of QNLP is its completely pictorial!

Uses three theories

- ➢ CQM
- ➢ DisCoCat

Quantum Natural Language Processing (QNLP)

- DisCoCat (Distributional Compositional Categorical) brand of NLP
- Combines linguistic meaning and linguistic structure
- Originated from the Categorical Quantum Mechanics (CQM)

An important feature of QNLP is its completely pictorial!

Uses three theories

- CQM
- DisCoCat

Interpreting language as quantum processes

Quantum Natural Language Processing (QNLP)

- DisCoCat (Distributional Compositional Categorical) brand of NLP

- Combines linguistic meaning and linguistic structure

- Originated from the Categorical Quantum Mechanics (CQM)

An important feature of QNLP is its completely pictorial!

Uses three theories

- CQM
- DisCoCat
- ZX Calculus

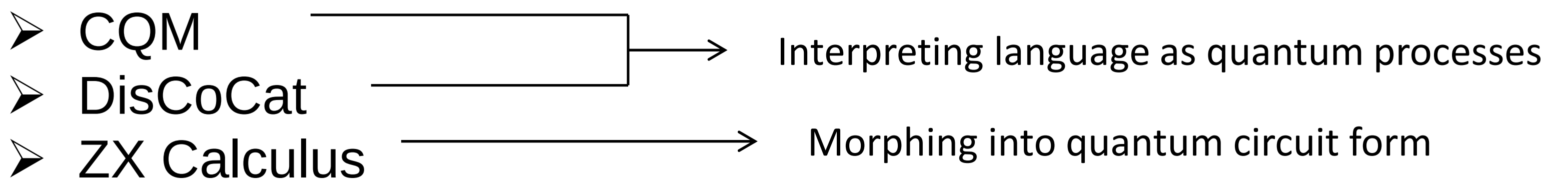
Interpreting language as quantum processes

Quantum Natural Language Processing (QNLP)

- ❖ DisCoCat (Distributional Compositional Categorical) brand of NLP
- ❖ Combines linguistic meaning and linguistic structure
- ❖ Originated from the Categorical Quantum Mechanics (CQM)

An important feature of QNLP is its completely pictorial!

Uses three theories



- ➢ CQM
- ➢ DisCoCat
- ➢ ZX Calculus

Interpreting language as quantum processes

Morphing into quantum circuit form

Quantum Natural Language Processing (QNLP)

Introduction to QNLP Concludes

Distributional Word Representation

Distributional Word Representation

- Consists of finding the meaning of a word according to the context in which it appears

Distributional Word Representation

❖ Consists of finding the meaning of a word according to the context in which it appears

❖ Consider that three context words – ‘fabric’, ‘washable’ & ‘clean’ appears 10 times, 4 times & 7 times in a corpus respectively

Distributional Word Representation

- Consists of finding the meaning of a word according to the context in which it appears

- Consider that three context words – ‘fabric’, ‘washable’ & ‘clean’ appears 10 times, 4 times & 7 times in a corpus respectively

- Therefore meaning of word ‘Shirt’ can be represented using the vector – [10,4,7]

Distributional Word Representation

- Consists of finding the meaning of a word according to the context in which it appears

- Consider that three context words – ‘fabric’, ‘washable’ & ‘clean’ appears 10 times, 4 times & 7 times in a corpus respectively

- Therefore meaning of word ‘Shirt’ can be represented using the vector – [10,4,7]

- Meanings of words can be depicted by high-dimensional “meaning-space” where context words act as orthogonal basis vectors

Distributional Word Representation

- Consists of finding the meaning of a word according to the context in which it appears

- Consider that three context words – ‘fabric’, ‘washable’ & ‘clean’ appears 10 times, 4 times & 7 times in a corpus respectively

- Therefore meaning of word ‘Shirt’ can be represented using the vector – [10,4,7]

- Meanings of words can be depicted by high-dimensional “meaning-space” where context words act as orthogonal basis vectors

- Inner product can be used to identify distances between different words which represents the closeness of meaning of the words

Distributional Word Representation

Distributional Word Representation Concludes

Compositionality of Grammar

Compositionality of Grammar

- In Lambek's pregroup grammar, noun is denoted by atomic type 'n' and sentence is denoted by atomic type 's'

Compositionality of Grammar

- In Lambek’s pregroup grammar, noun is denoted by atomic type ‘n’ and sentence is denoted by atomic type ‘s’

- Each word in a sentence are assigned these types – either ‘n’ or ‘s’ depending on the structure of the sentence

Compositionality of Grammar

- In Lambek’s pregroup grammar, noun is denoted by atomic type ‘n’ and sentence is denoted by atomic type ‘s’

- Each word in a sentence are assigned these types – either ‘n’ or ‘s’ depending on the structure of the sentence

- According to pregroup grammar rules, the words can be combined with each other

Compositionality of Grammar

- ❖ In Lambek’s pregroup grammar, noun is denoted by atomic type ‘n’ and sentence is denoted by atomic type ‘s’

- ❖ Each word in a sentence are assigned these types – either ‘n’ or ‘s’ depending on the structure of the sentence

- ❖ According to pregroup grammar rules, the words can be combined with each other

- ❖ Therefore, pregroup grammar is compositional grammar which makes it possible to give meaning to large syntactic units present in a sentence

Compositionality of Grammar

- ❖ In Lambek’s pregroup grammar, noun is denoted by atomic type ‘n’ and sentence is denoted by atomic type ‘s’

- ❖ Each word in a sentence are assigned these types – either ‘n’ or ‘s’ depending on the structure of the sentence

- ❖ According to pregroup grammar rules, the words can be combined with each other

- ❖ Therefore, pregroup grammar is compositional grammar which makes it possible to give meaning to large syntactic units present in a sentence

- ❖ If the juxtaposition of the types of the words within a sentence reduces to the basic type s, then the sentence is grammatical

Compositionality of Grammar

- In Lambek's pregroup grammar, noun is denoted by atomic type 'n' and sentence is denoted by atomic type 's'

- Each word in a sentence are assigned these types – either 'n' or 's' depending on the structure of the sentence

- According to pregroup grammar rules, the words can be combined with each other

- Therefore, pregroup grammar is compositional grammar which makes it possible to give meaning to large syntactic units present in a sentence

- If the juxtaposition of the types of the words within a sentence reduces to the basic type s, then the sentence is grammatical

- Diagrammatically, the composition is denoted by Cups & Caps, meaning that grammar compositionality is entanglement of words together!

Compositionality of Grammar

Compositionality of Grammar Concludes

QNLP Basics

Adjective & Noun

QNLP Basics

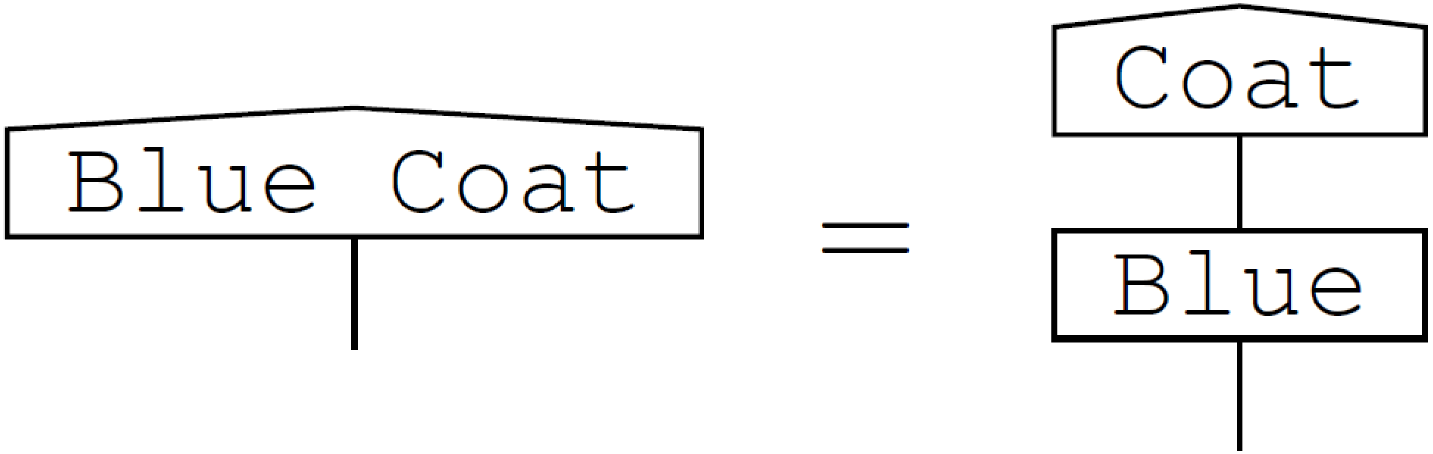
Adjective & Noun

Lets take the example Blue Coat, where ‘Coat’ is the noun and ‘Blue’ is an adjective

QNLP Basics

Adjective & Noun

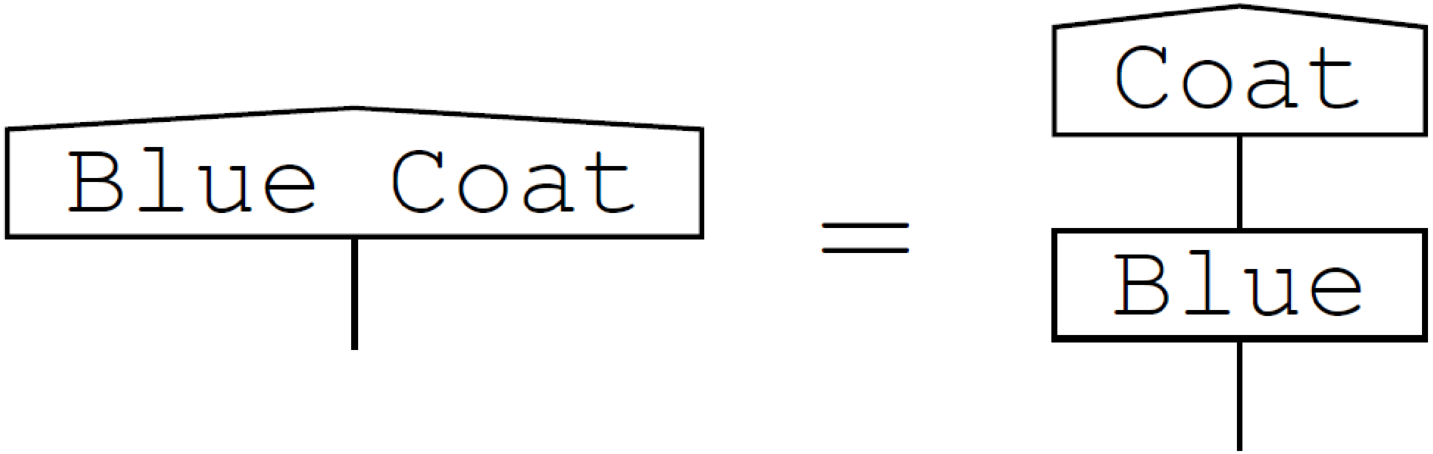
Lets take the example Blue Coat, where ‘Coat’ is the noun and ‘Blue’ is an adjective



QNLP Basics

Adjective & Noun

Lets take the example Blue Coat, where ‘Coat’ is the noun and ‘Blue’ is an adjective



Applying an adjective ‘Blue’ to a noun ‘Coat’ is like applying a process to ‘Coat’

QNLP Basics

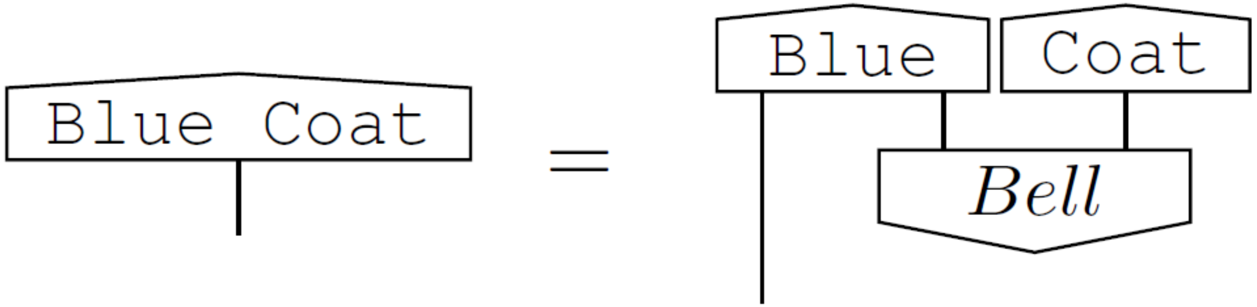
Adjective & Noun

Representing ‘Blue’ as a 2-qubit state, we get another way of getting the meaning of Blue Coat from the meaning of ‘Blue’ and ‘Coat’

QNLP Basics

Adjective & Noun

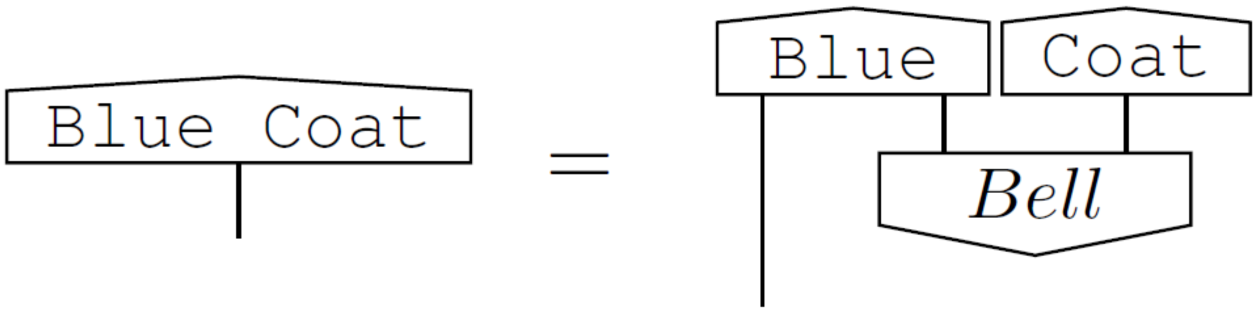
Representing ‘Blue’ as a 2-qubit state, we get another way of getting the meaning of Blue Coat from the meaning of ‘Blue’ and ‘Coat’

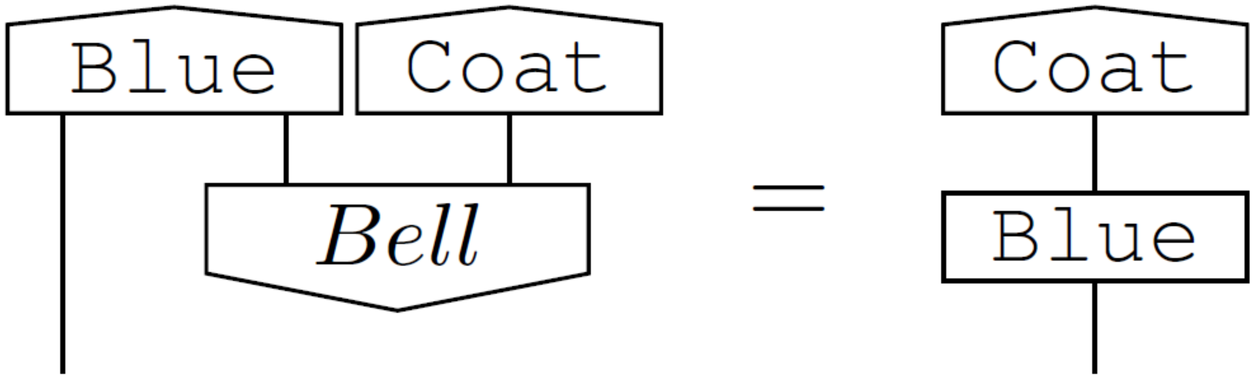


QNLP Basics

Adjective & Noun

Representing ‘Blue’ as a 2-qubit state, we get another way of getting the meaning of Blue Coat from the meaning of ‘Blue’ and ‘Coat’





QNLP Basics

Adjective & Noun

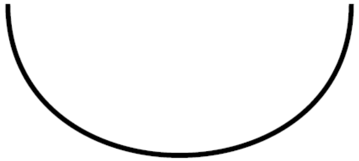
Representing the Bell-state/effect by cap-/cup-shaped bend wires

QNLP Basics

Adjective & Noun

Representing the Bell-state/effect by cap-/cup-shaped bend wires

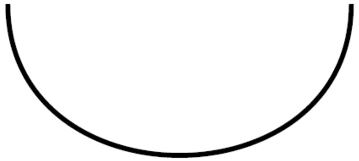
$$\langle Bell| =$$

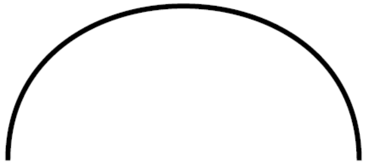


QNLP Basics

Adjective & Noun

Representing the Bell-state/effect by cap-/cup-shaped bend wires

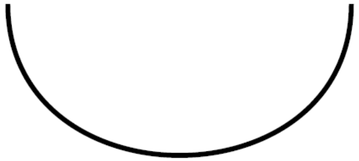
$\langle Bell| =$ 

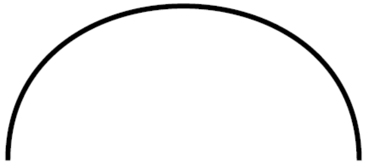
$|Bell\rangle =$ 

QNLP Basics

Adjective & Noun

Representing the Bell-state/effect by cap-/cup-shaped bend wires

$\langle Bell| =$ 

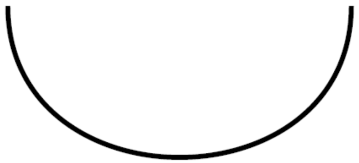
$|Bell\rangle =$ 

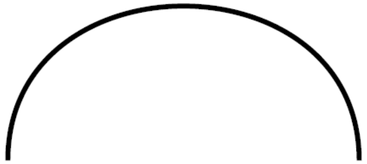
We obtain the following diagram of Blue Coat after yanking

QNLP Basics

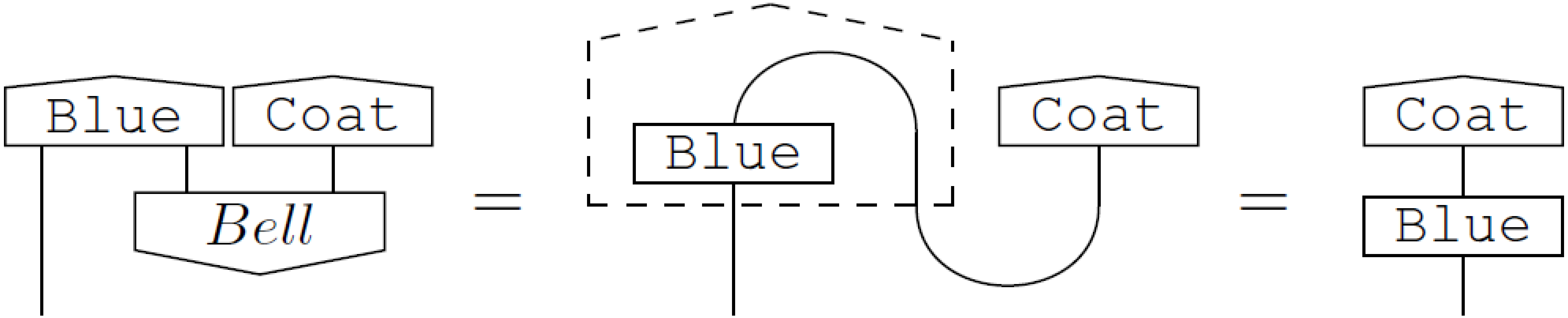
Adjective & Noun

Representing the Bell-state/effect by cap-/cup-shaped bend wires

$\langle Bell| =$ 

$|Bell\rangle =$ 

We obtain the following diagram of Blue Coat after yanking



QNLP Basics

Adjective & Noun

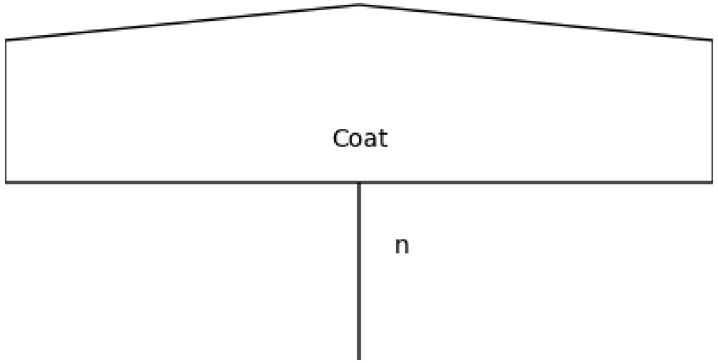


Diagram Generated using https://qnlp.cambridgequantum.com/generate.html

QNLP Basics

Adjective & Noun

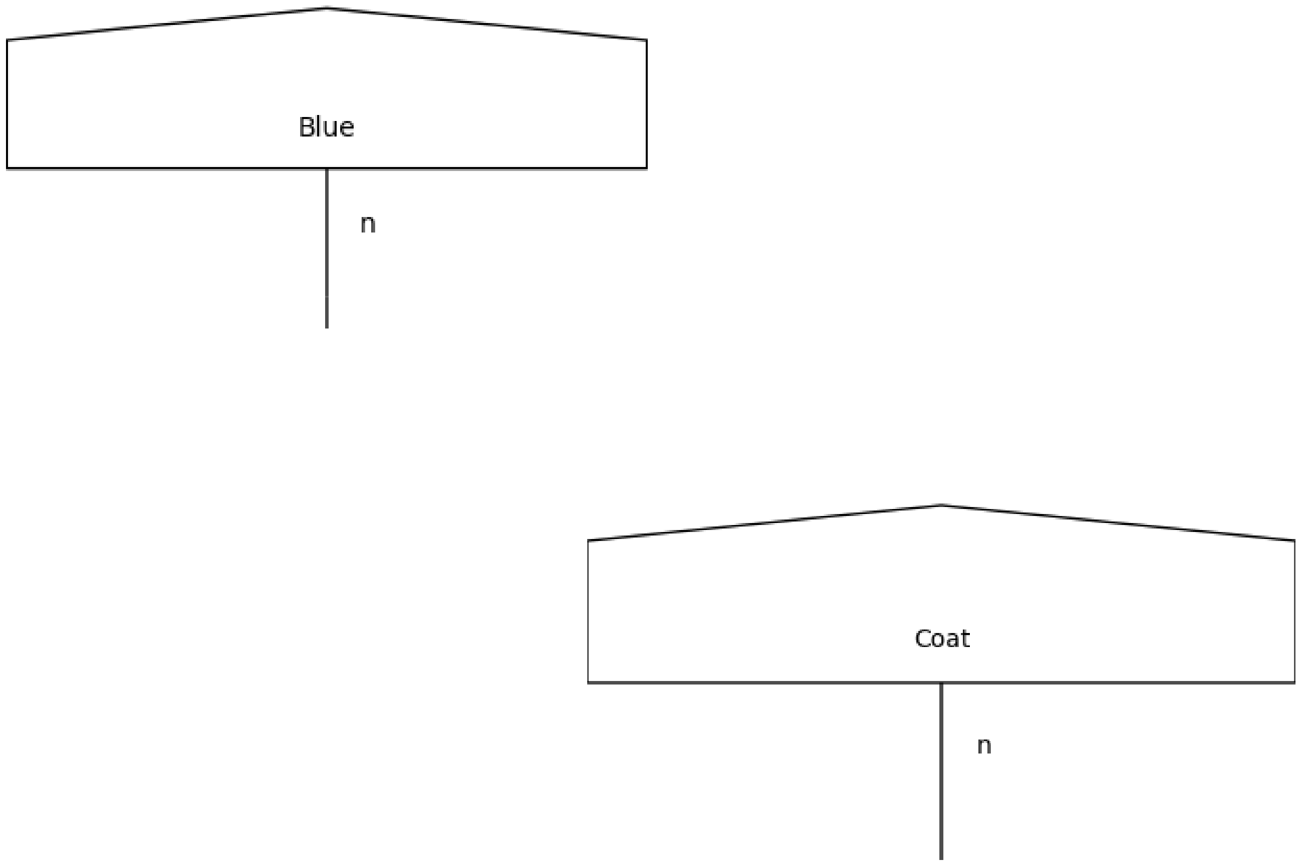


Diagram Generated using https://qnlp.cambridgequantum.com/generate.html

QNLP Basics

Adjective & Noun

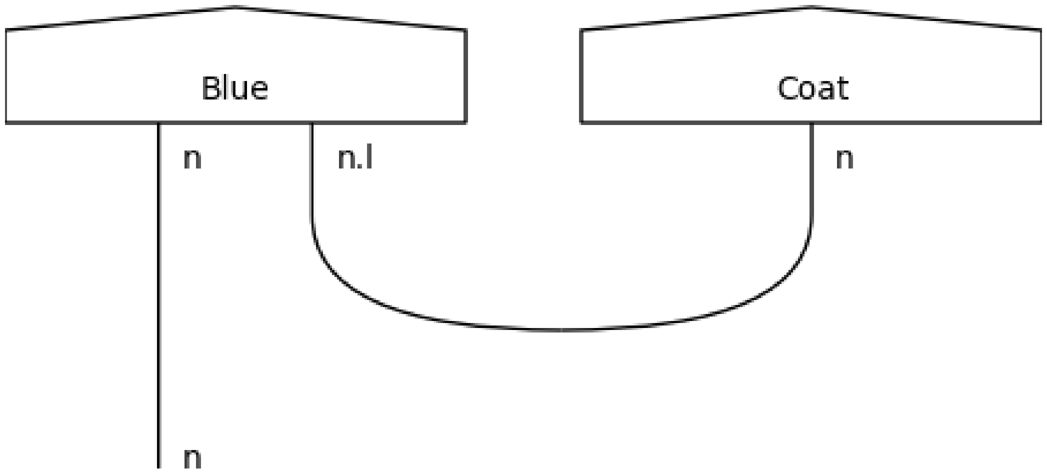
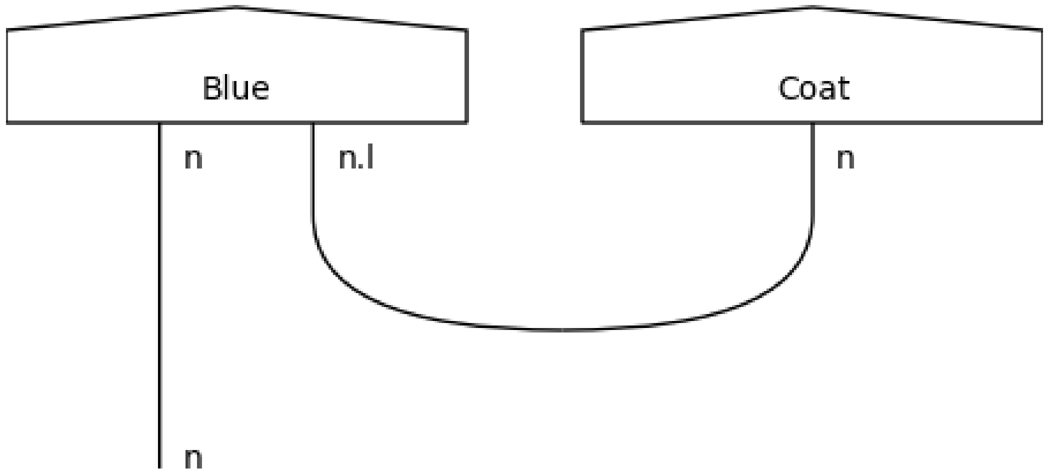


Diagram Generated using https://qnlp.cambridgequantum.com/generate.html

QNLP Basics

Adjective & Noun

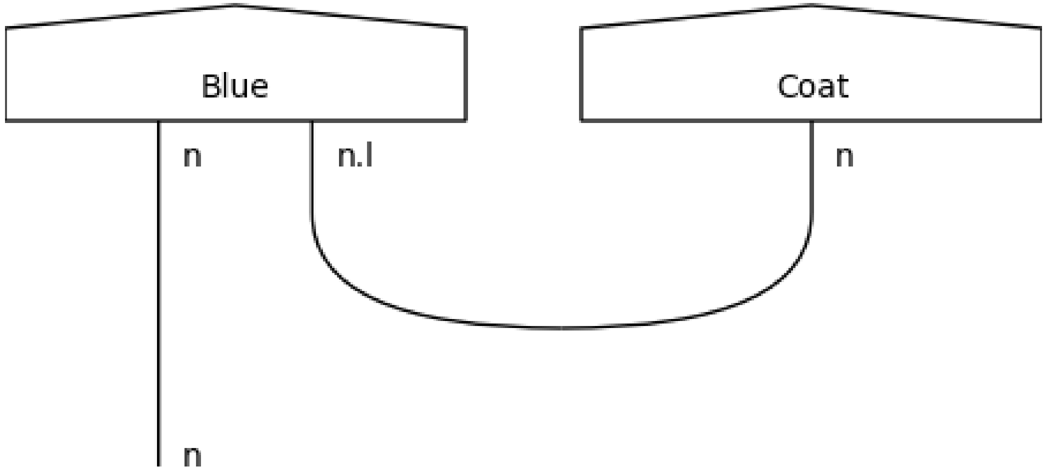


- ➢ For the noun ‘Coat’, the adjective ‘Blue’ is on the left, therefore, the cup connects n to n^l

Diagram Generated using https://qnlp.cambridgequantum.com/generate.html

QNLP Basics

Adjective & Noun



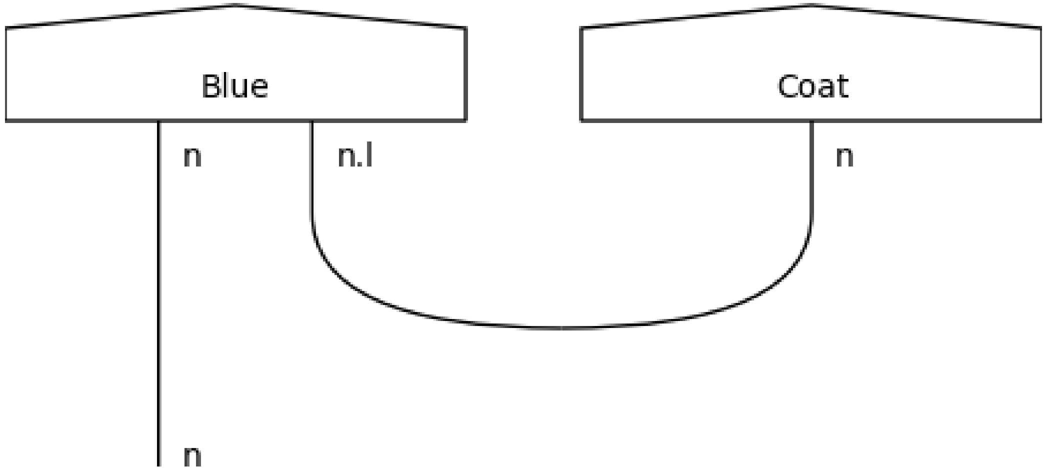
- For the noun ‘Coat’, the adjective ‘Blue’ is on the left, therefore, the cup connects n to n^l

- Consequently, the adjective ‘Blue’ get the compound type $s \cdot n^l$ indicating that it expects n on the right

Diagram Generated using https://qnlp.cambridgequantum.com/generate.html

QNLP Basics

Adjective & Noun



- For the noun ‘Coat’, the adjective ‘Blue’ is on the left, therefore, the cup connects n to n^l

- Consequently, the adjective ‘Blue’ get the compound type $s \cdot n^l$ indicating that it expects n on the right

- Juxtaposition of types - $s \cdot n^l \cdot n \rightarrow s \cdot 1 \rightarrow s$

Diagram Generated using https://qnlp.cambridgequantum.com/generate.html

QNLP Basics

Adjective & Noun

Adjective & Noun Concludes

Subject-Verb-Object Sentence

Subject-Verb-Object Sentence

Let us take the sentence ‘Eric hates Trev’, where ‘Eric’ is the subject, ‘hates’ is the verb and ‘Trev’ is the object

Subject-Verb-Object Sentence

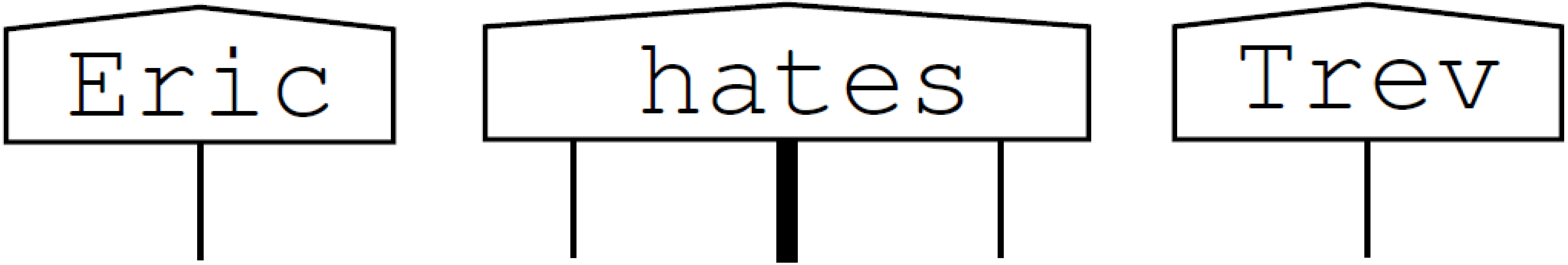
Let us take the sentence ‘Eric hates Trev’, where ‘Eric’ is the subject, ‘hates’ is the verb and ‘Trev’ is the object

Individual diagrams for Eric, hates and Trev

Subject-Verb-Object Sentence

Let us take the sentence ‘Eric hates Trev’, where ‘Eric’ is the subject, ‘hates’ is the verb and ‘Trev’ is the object

Individual diagrams for Eric, hates and Trev

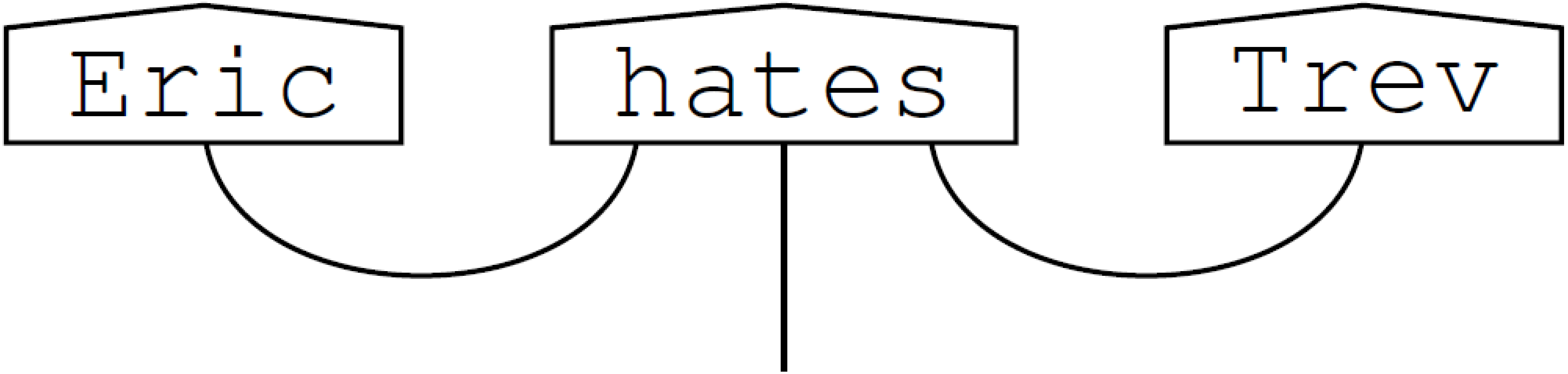


Subject-Verb-Object Sentence

Meaning of the sentence is computed as follows (string diagram)

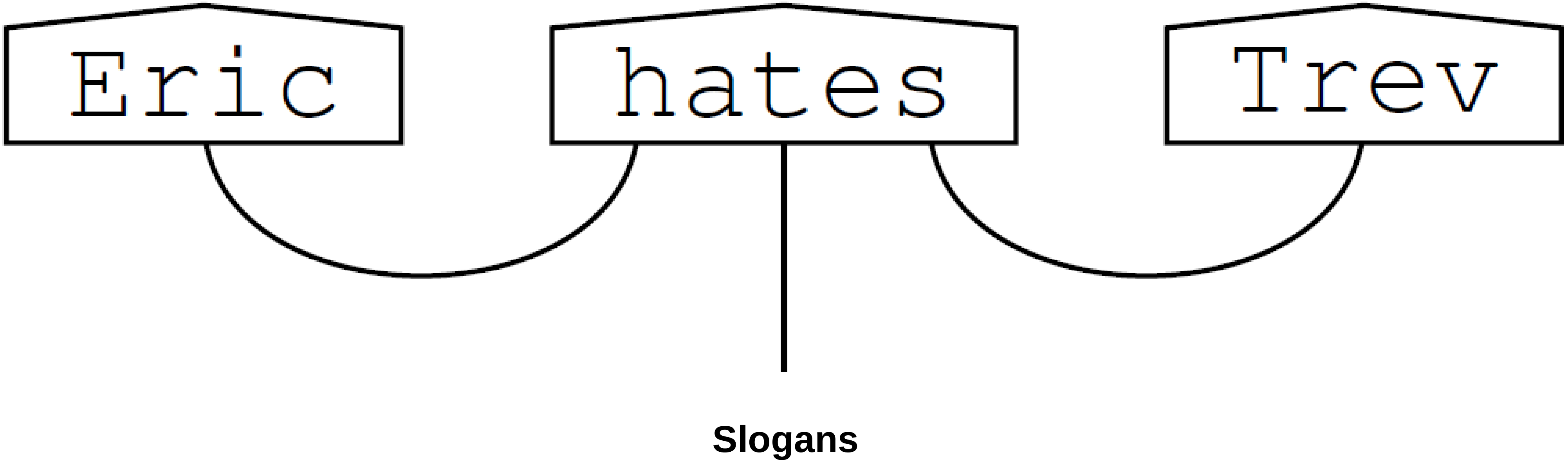
Subject-Verb-Object Sentence

Meaning of the sentence is computed as follows (string diagram)



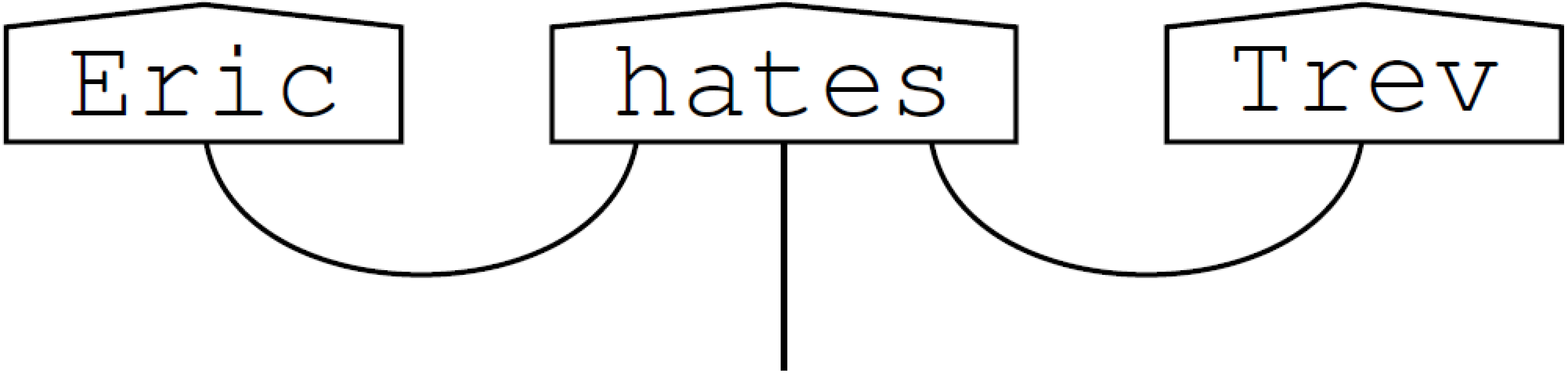
Subject-Verb-Object Sentence

Meaning of the sentence is computed as follows (string diagram)



Subject-Verb-Object Sentence

Meaning of the sentence is computed as follows (string diagram)

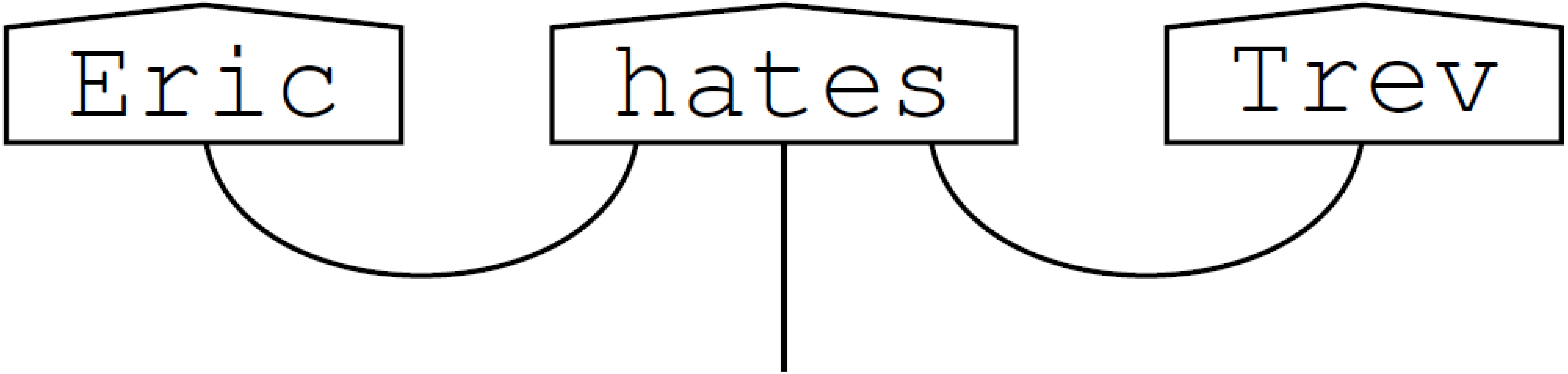


Slogans

A Sentence is a process which alters the meaning of its inhabitants (words)

Subject-Verb-Object Sentence

Meaning of the sentence is computed as follows (string diagram)



Slogans

A Sentence is a process which alters the meaning of its inhabitants (words)

Grammar is what mediates the flow of meanings between words

Subject-Verb-Object Sentence

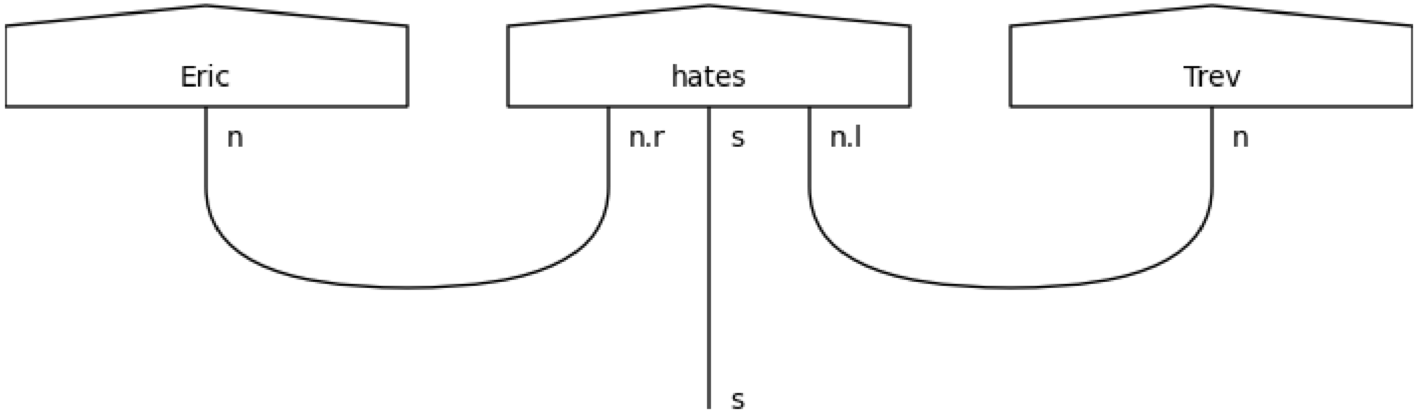
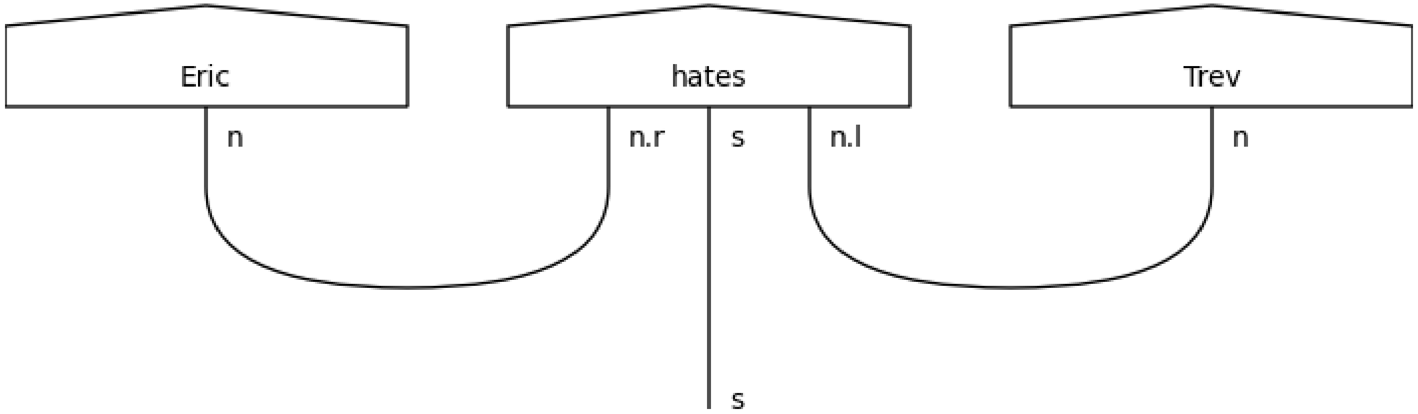


Diagram Generated using https://qnlp.cambridgequantum.com/generate.html

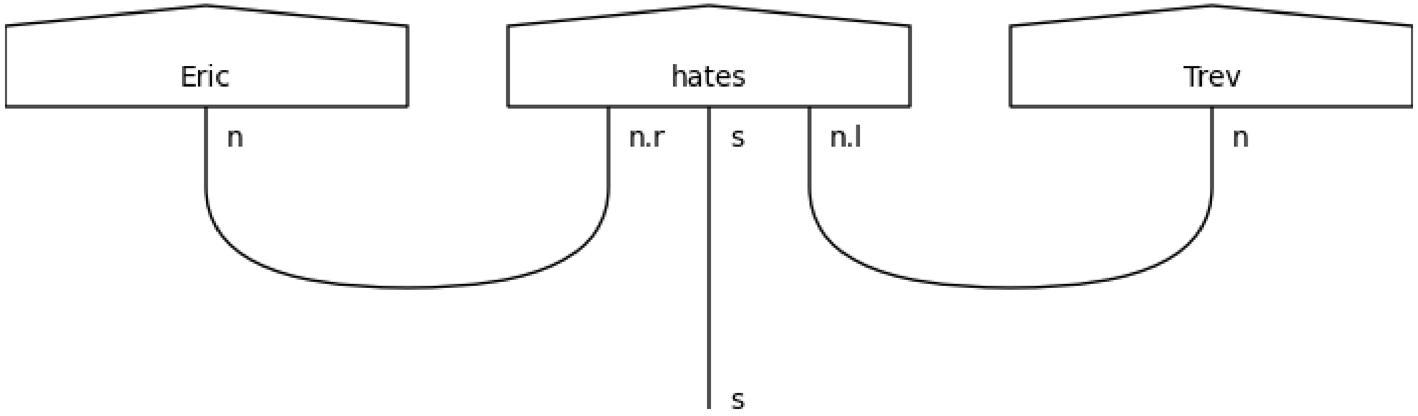
Subject-Verb-Object Sentence



- For the noun ‘Eric’, the verb ‘hates’ is on the right, therefore, the cup connects n to n^r

Diagram Generated using https://qnlp.cambridgequantum.com/generate.html

Subject-Verb-Object Sentence

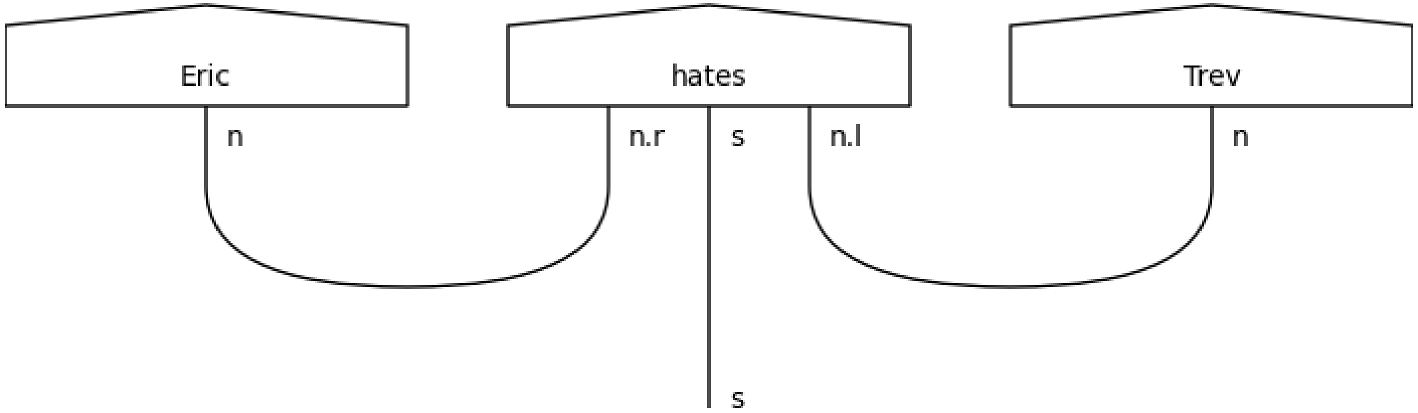


- For the noun ‘Eric’, the verb ‘hates’ is on the right, therefore, the cup connects n to n^r

- For the noun ‘Trev’, the verb ‘hates’ is on the left, therefore, the cup connects n to n^l

Diagram Generated using https://qnlp.cambridgequantum.com/generate.html

Subject-Verb-Object Sentence



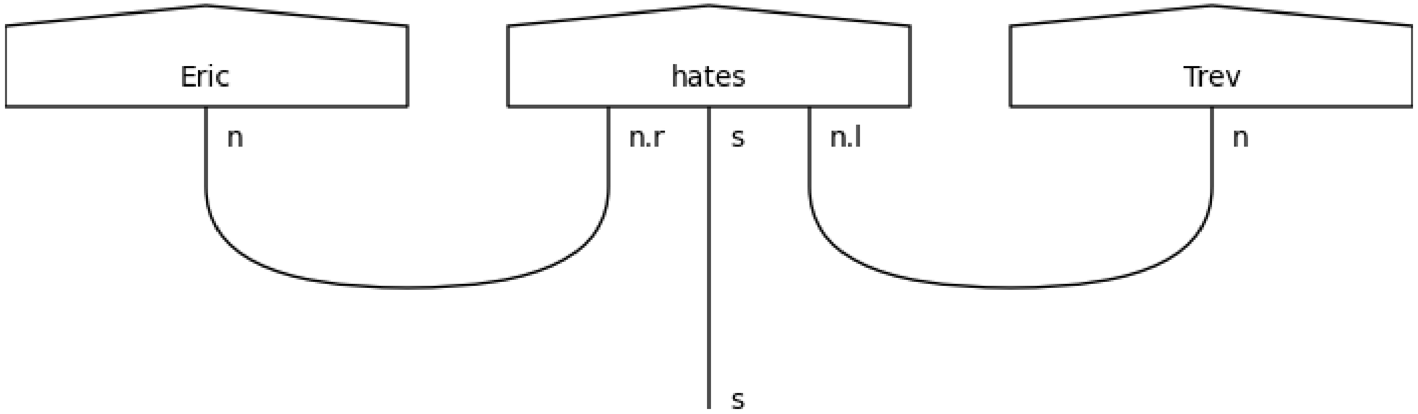
- For the noun ‘Eric’, the verb ‘hates’ is on the right, therefore, the cup connects n to n^r

- For the noun ‘Trev’, the verb ‘hates’ is on the left, therefore, the cup connects n to n^l

- Consequently, the verb ‘hates’ get the compound type $n^r \cdot s \cdot n^l$ indicating that it expects n on the left and right to return s

Diagram Generated using https://qnlp.cambridgequantum.com/generate.html

Subject-Verb-Object Sentence



- For the noun ‘Eric’, the verb ‘hates’ is on the right, therefore, the cup connects n to n^r

- For the noun ‘Trev’, the verb ‘hates’ is on the left, therefore, the cup connects n to n^l

- Consequently, the verb ‘hates’ get the compound type $n^r \cdot s \cdot n^l$ indicating that it expects n on the left and right to return s

- Juxtaposition of types - $n \cdot n^r \cdot s \cdot n^l \cdot n \rightarrow 1 \cdot s \cdot 1 \rightarrow s$

Diagram Generated using https://qnlp.cambridgequantum.com/generate.html

Subject-Verb-Object Sentence

Subject-Verb-Object Sentence Concludes

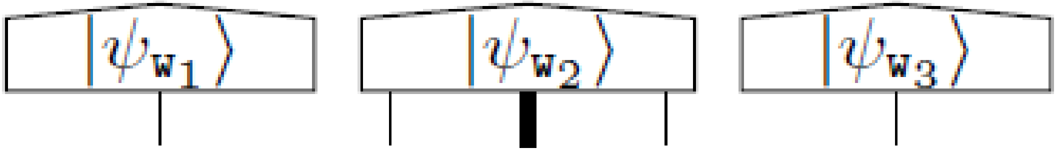
DisCoCat Algorithm

DisCoCat Algorithm

- A set of words which grammatically makes sense are put into the spaces as shown below

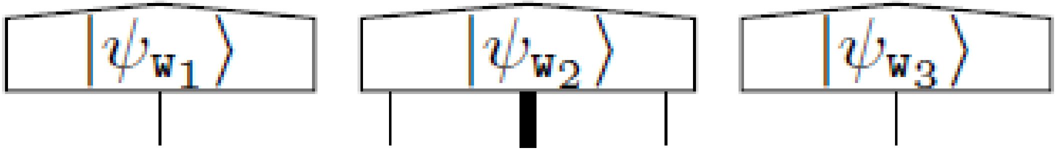
DisCoCat Algorithm

- A set of words which grammatically makes sense are put into the spaces as shown below



DisCoCat Algorithm

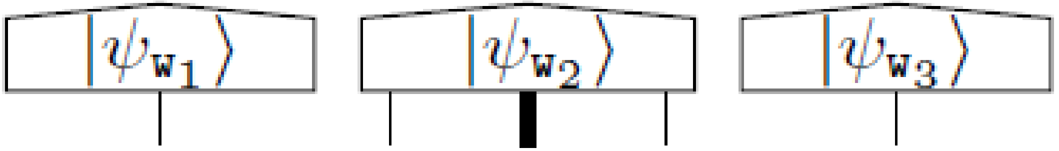
- A set of words which grammatically makes sense are put into the spaces as shown below



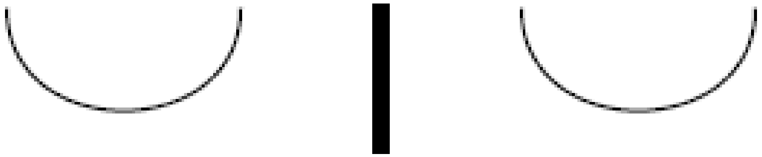
- Build the wire diagram representing the grammar

DisCoCat Algorithm

- A set of words which grammatically makes sense are put into the spaces as shown below



- Build the wire diagram representing the grammar

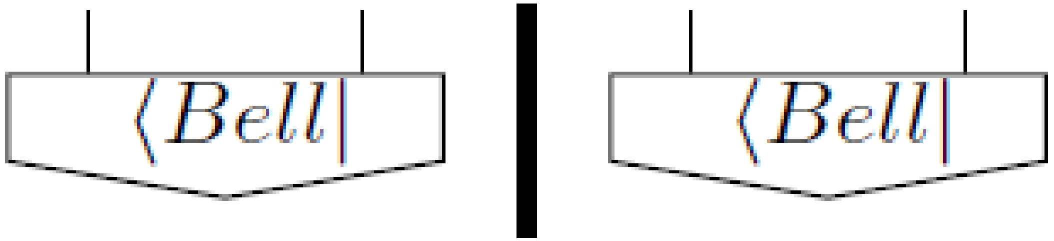


DisCoCat Algorithm

- Replace cups with bell states and all the straight wires with Identities

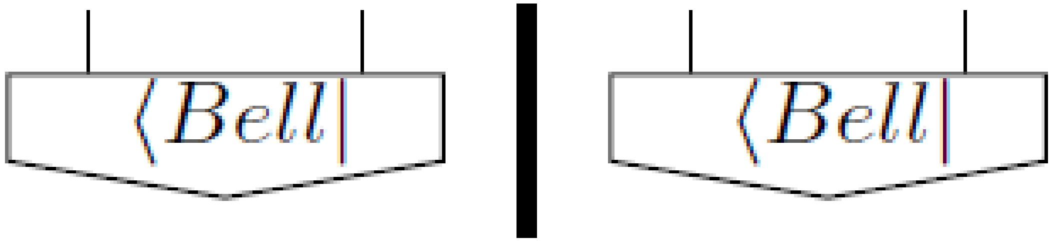
DisCoCat Algorithm

- Replace cups with bell states and all the straight wires with Identities



DisCoCat Algorithm

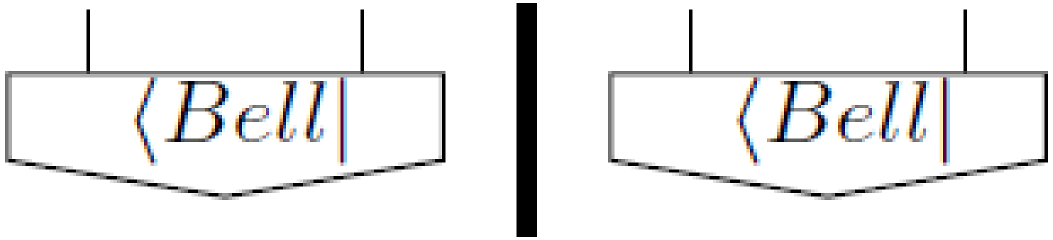
- ❖ Replace cups with bell states and all the straight wires with Identities



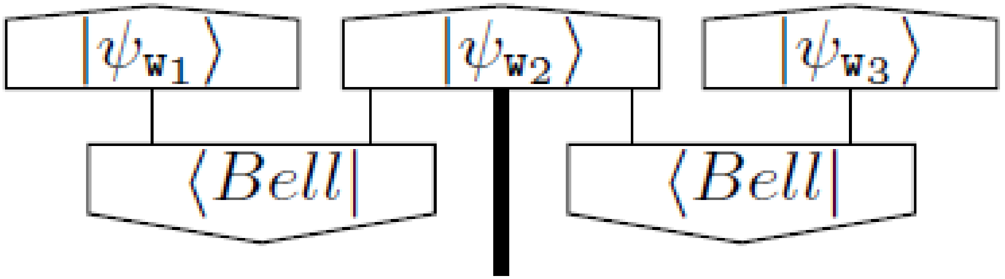
- ❖ Apply this quantum process to the meaning of the parts

DisCoCat Algorithm

- ❖ Replace cups with bell states and all the straight wires with Identities



- ❖ Apply this quantum process to the meaning of the parts



DisCoCat Algorithm

DisCoCat Algorithm Concludes

String Diagrams into ZX Quantum Circuits

String Diagrams into ZX Quantum Circuits

- String diagrams generated by the DisCoCat Algorithm needs to be converted into quantum circuit in order to run them on a quantum computer

String Diagrams into ZX Quantum Circuits

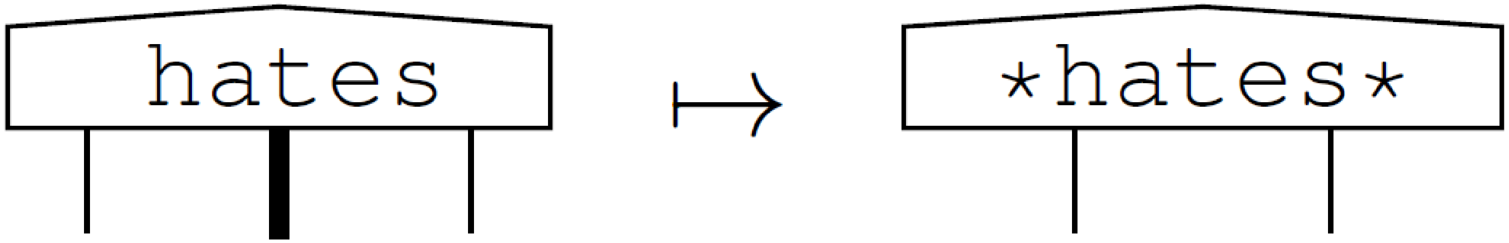
- String diagrams generated by the DisCoCat Algorithm needs to be converted into quantum circuit in order to run them on a quantum computer

- Due to the dimensionality issues the internal wiring of the system is changed

String Diagrams into ZX Quantum Circuits

- String diagrams generated by the DisCoCat Algorithm needs to be converted into quantum circuit in order to run them on a quantum computer

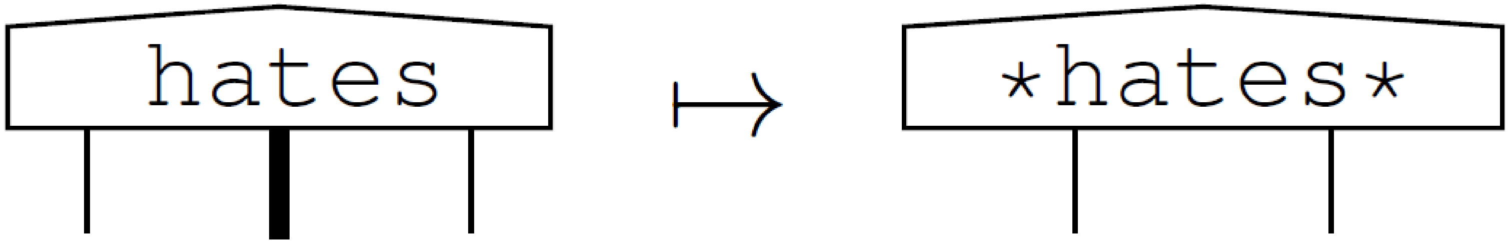
- Due to the dimensionality issues the internal wiring of the system is changed



String Diagrams into ZX Quantum Circuits

- String diagrams generated by the DisCoCat Algorithm needs to be converted into quantum circuit in order to run them on a quantum computer

- Due to the dimensionality issues the internal wiring of the system is changed



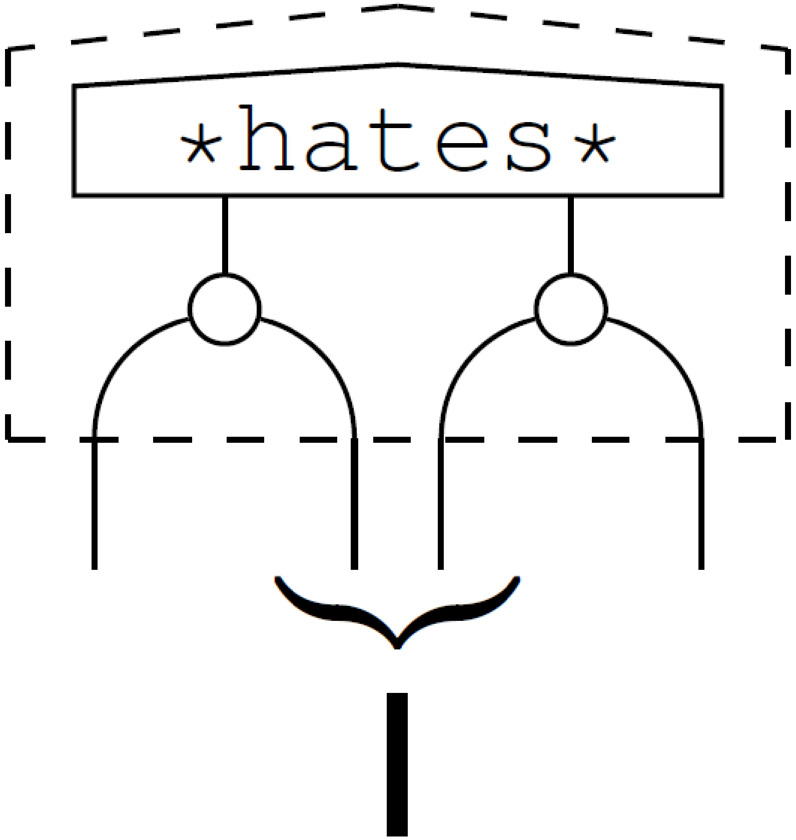
- ZX calculus and Spiders is used to modify the internal wiring of the system.

String Diagrams into ZX Quantum Circuits

- The internal wiring is shown in the following figure which has two Z spiders in them

String Diagrams into ZX Quantum Circuits

- The internal wiring is shown in the following figure which has two Z spiders in them

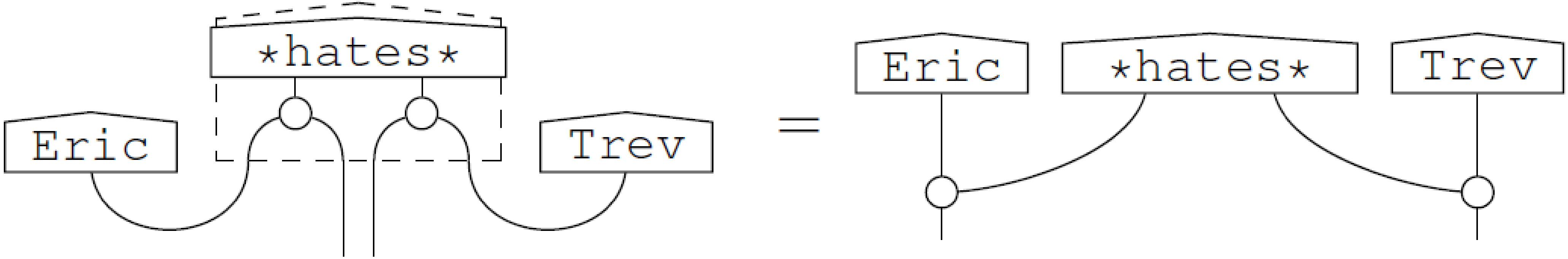


String Diagrams into ZX Quantum Circuits

- With the help of the internal wiring, we can start converting the string diagram into a ZX quantum circuit

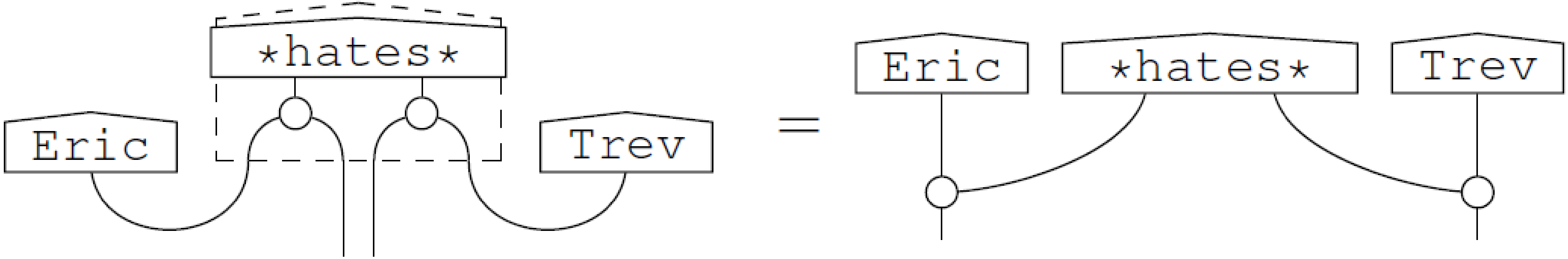
String Diagrams into ZX Quantum Circuits

- With the help of the internal wiring, we can start converting the string diagram into a ZX quantum circuit



String Diagrams into ZX Quantum Circuits

- With the help of the internal wiring, we can start converting the string diagram into a ZX quantum circuit



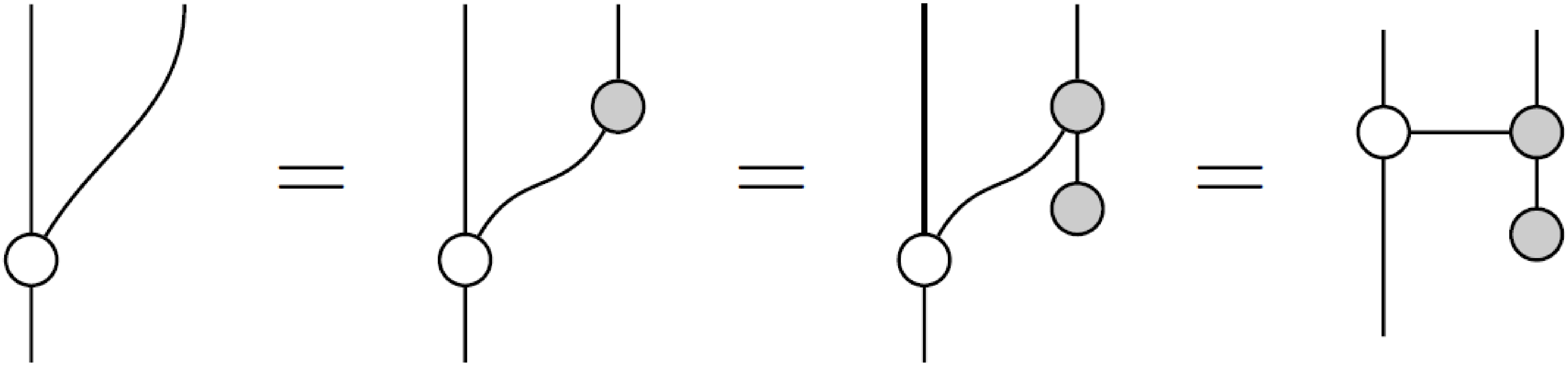
- The Z spiders are moved below the nouns – Eric & Trev and the extra wiring is removed

String Diagrams into ZX Quantum Circuits

- We can now use ZX calculus rewrite rules as shown below to convert the string diagram into a circuit

String Diagrams into ZX Quantum Circuits

- We can now use ZX calculus rewrite rules as shown below to convert the string diagram into a circuit

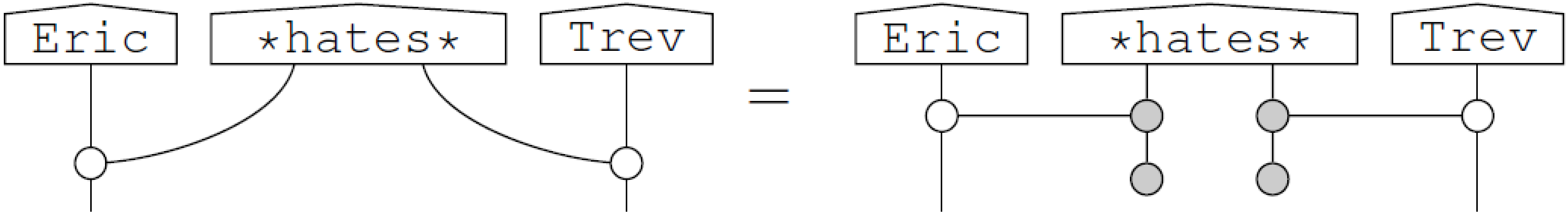


String Diagrams into ZX Quantum Circuits

- Two CNOT gates are being formed when the spiders are pulled out with ZX rules used and last two spiders (grey) are for post-selected measurement

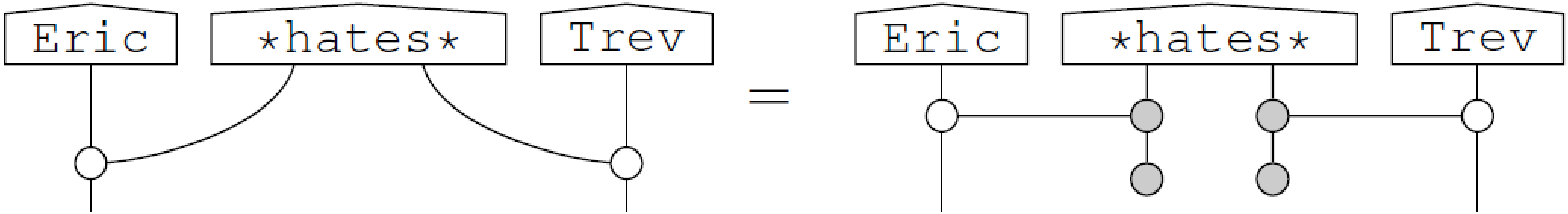
String Diagrams into ZX Quantum Circuits

- Two CNOT gates are being formed when the spiders are pulled out with ZX rules used and last two spiders (grey) are for post-selected measurement



String Diagrams into ZX Quantum Circuits

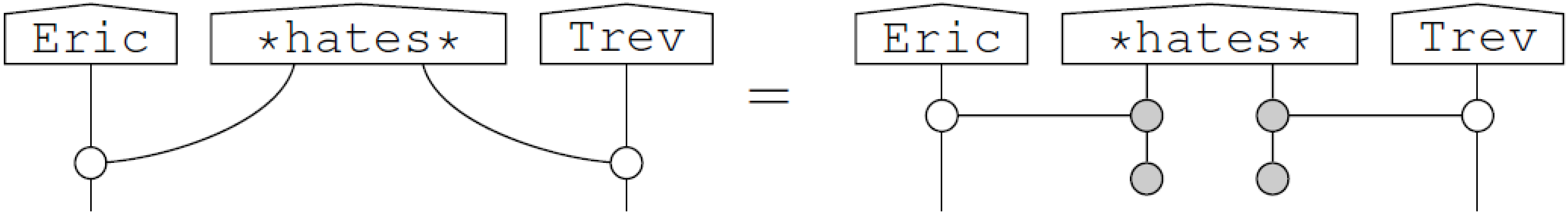
- Two CNOT gates are being formed when the spiders are pulled out with ZX rules used and last two spiders (grey) are for post-selected measurement



- For the complete quantum circuit between the subject and objects - Eric and Trev, the nouns are also converted into parameterized gates

String Diagrams into ZX Quantum Circuits

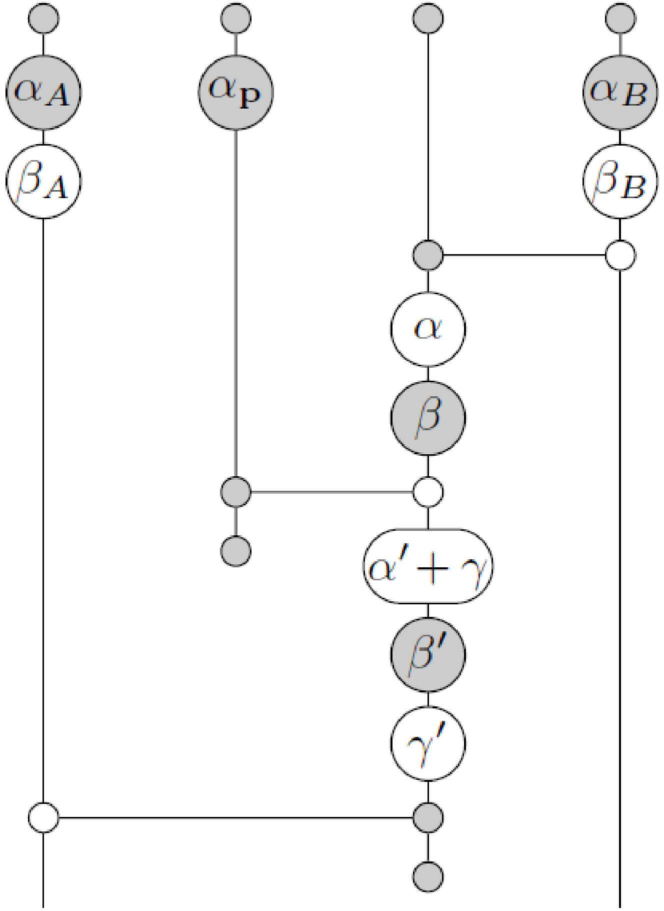
- ➢ Two CNOT gates are being formed when the spiders are pulled out with ZX rules used and last two spiders (grey) are for post-selected measurement



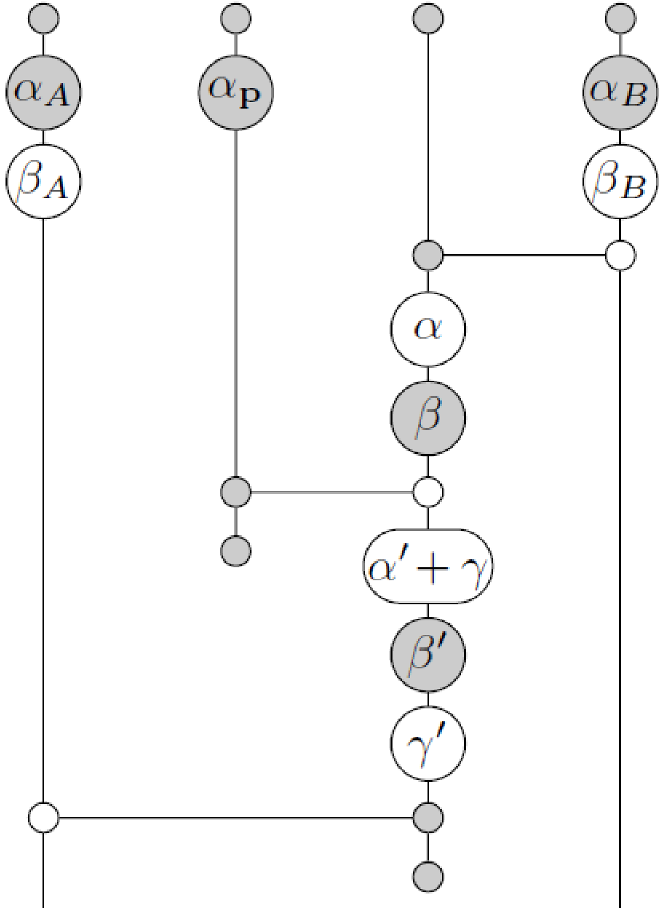
- ➢ For the complete quantum circuit between the subject and objects - Eric and Trev, the nouns are also converted into parameterized gates

- ➢ We represent verbs like ‘hates’ with ZX unitary gates and large space consisting of different verbs can be encoded by α, β, γ

String Diagrams into ZX Quantum Circuits

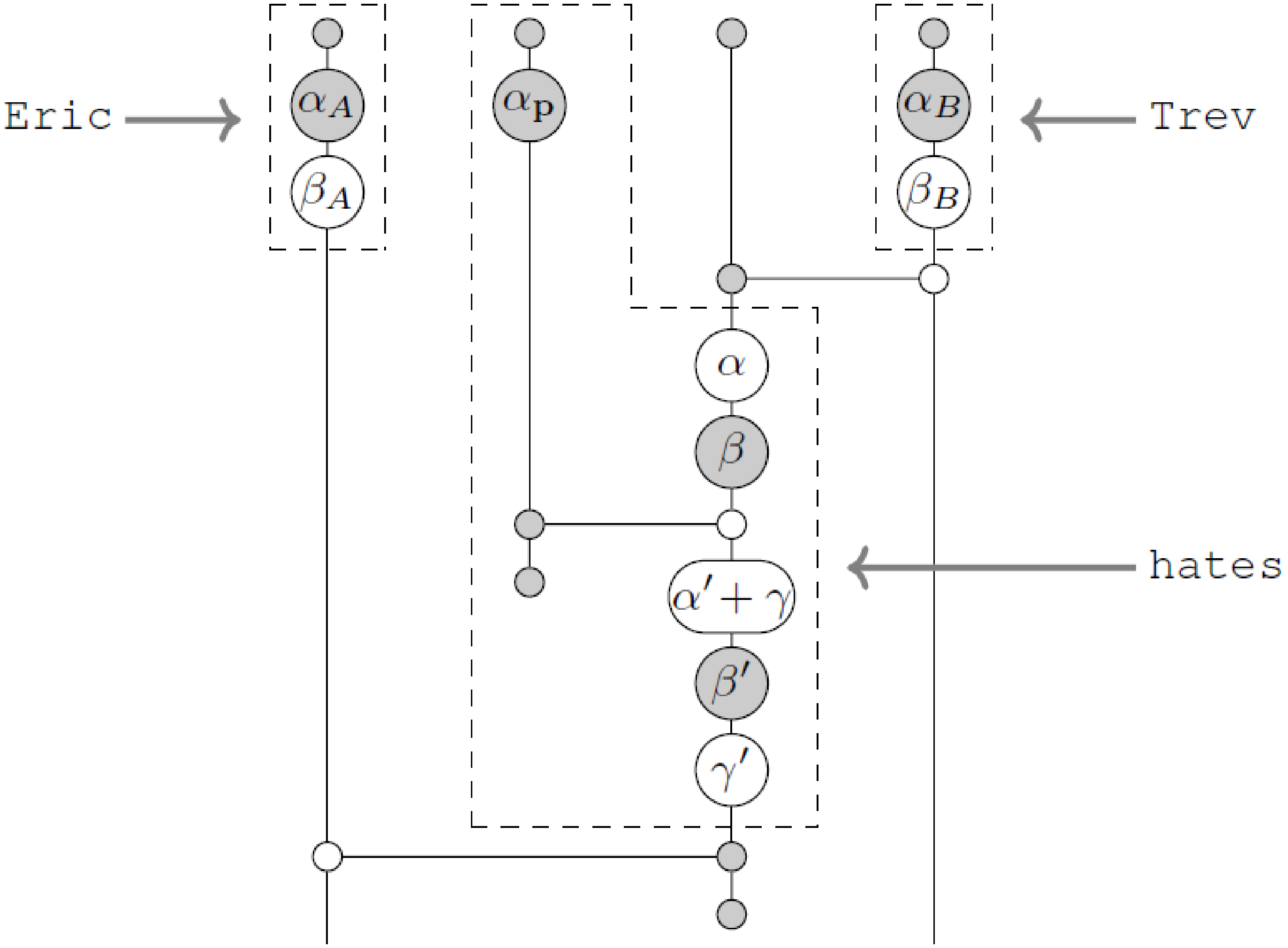


String Diagrams into ZX Quantum Circuits



- ➢ We can depict different verbs by changing the values of α, β, γ

String Diagrams into ZX Quantum Circuits



String Diagrams into ZX Quantum Circuits

String Diagrams into ZX Quantum Circuits Concludes

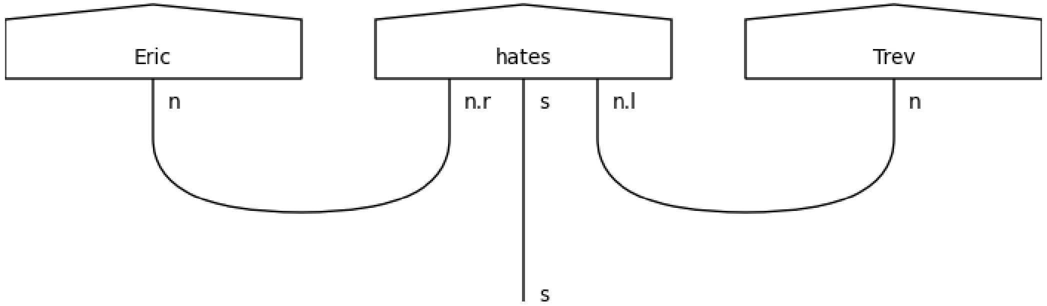
Introducing λambeq & it’s Features

Introducing *λ*ambeq & it’s Features

- ❖ Released by Cambridge Quantum on October 13th, 2021

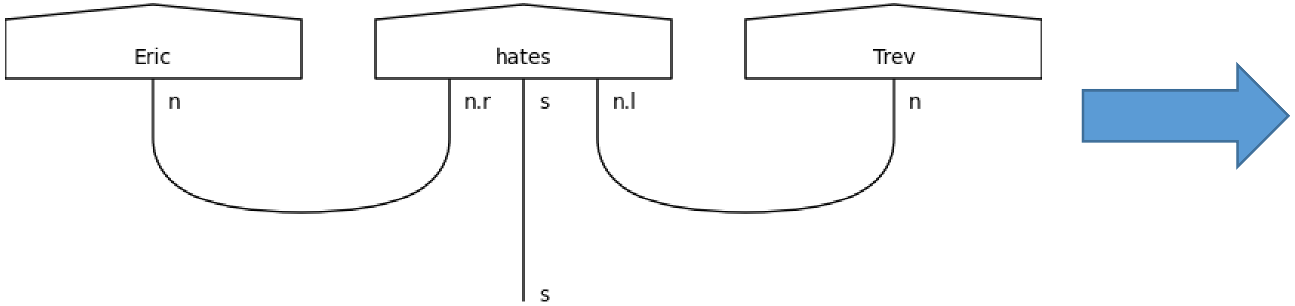
Introducing *λ*ambeq & it’s Features

- ❖ Released by Cambridge Quantum on October 13th, 2021



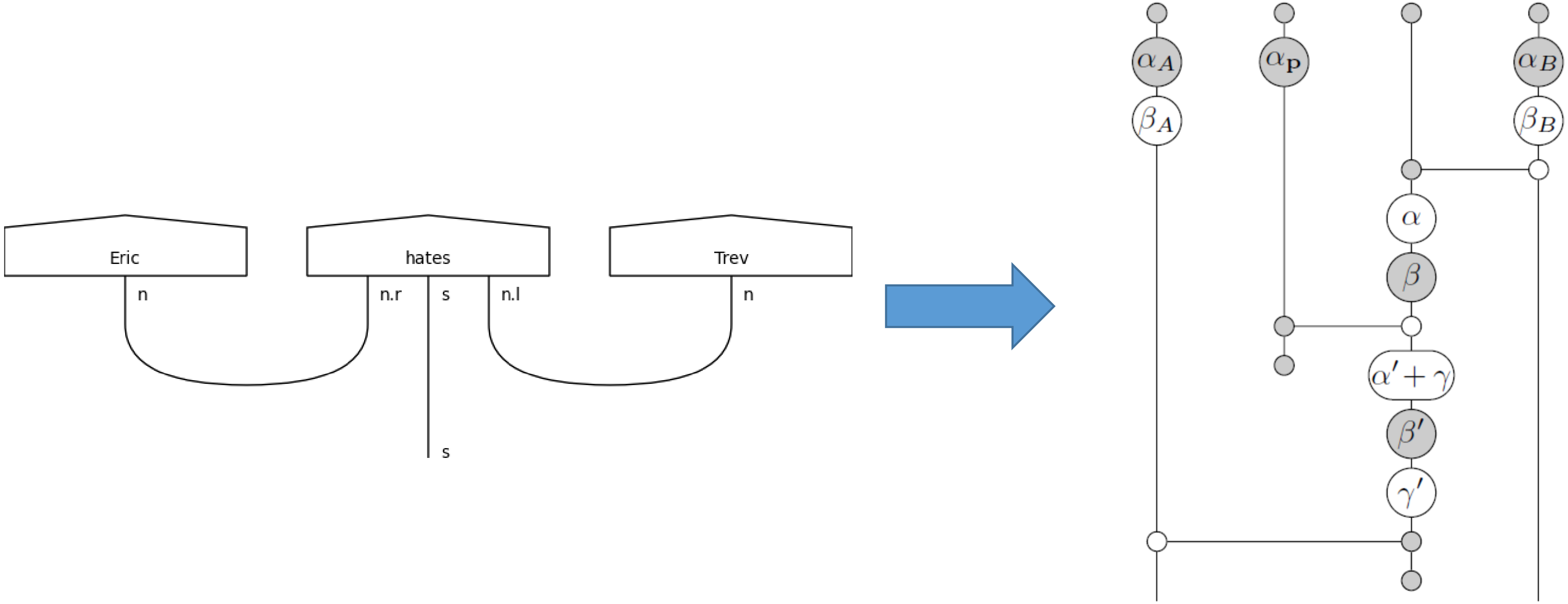
Introducing 𝜆ambeq & it’s Features

- ❖ Released by Cambridge Quantum on October 13th, 2021



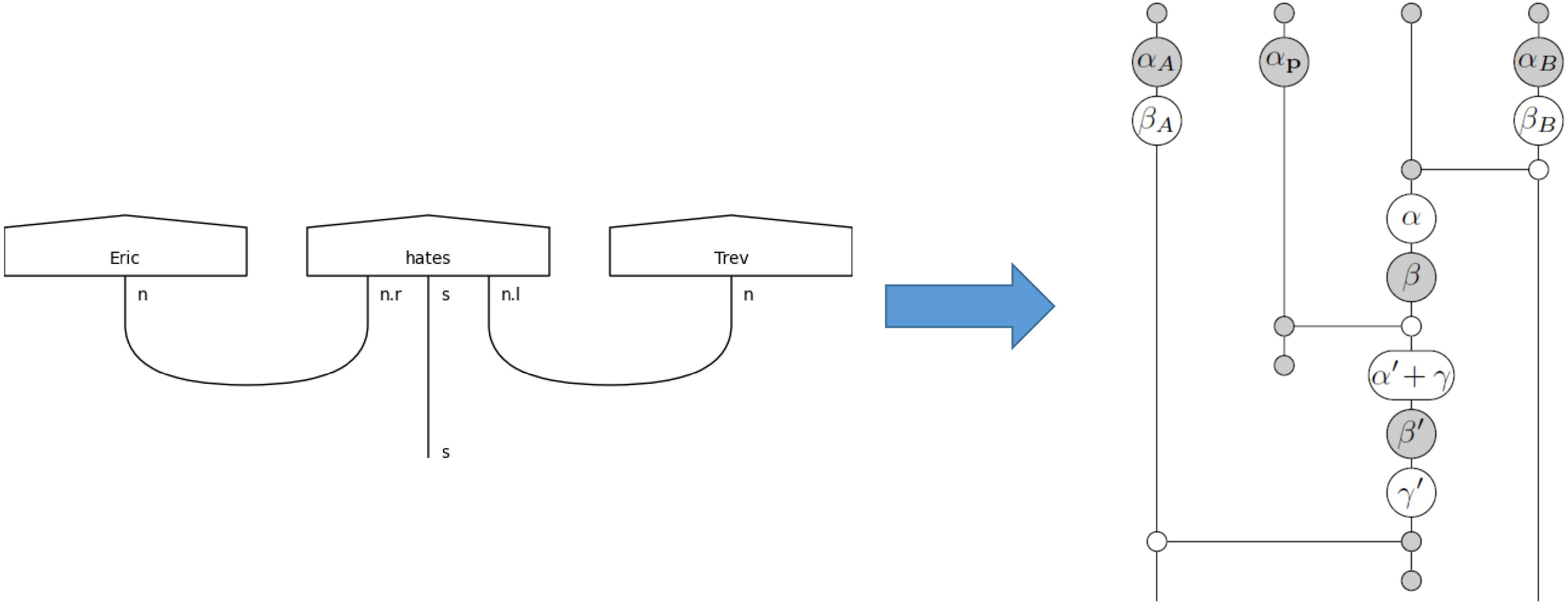
Introducing λambeq & it's Features

❖ Released by Cambridge Quantum on October 13th, 2021



Introducing *λ*ambeq & it’s Features

- ❖ Released by Cambridge Quantum on October 13th, 2021



- ❖ Converts any sentence into quantum circuit based on compositional model (DisCoCat, Bag of Words & Word Sequence) and choices of ansatz (SpiderAnsatz, TensorAnsatz, IQPAnsatz)

Introducing *λ*ambeq & it’s Features

Features of *λ*ambeq

Introducing *λ*ambeq & it’s Features

Features of *λ*ambeq

- ❖ Modular – Independent modules providing flexibility

Introducing *λ*ambeq & it’s Features

Features of *λ*ambeq

- ❖ Modular – Independent modules providing flexibility
- ❖ Open source

Introducing *λ*ambeq & it’s Features

Features of *λ*ambeq

❖ Modular – Independent modules providing flexibility

❖ Open source

❖ Extensive – Object-oriented design

Introducing 𝜆ambeq & it’s Features

Features of 𝜆ambeq

- ❖ Modular – Independent modules providing flexibility
- ❖ Open source
- ❖ Extensive – Object-oriented design
- ❖ High level

Introducing λambeq & it’s Features

Features of λambeq

- ❖ Modular – Independent modules providing flexibility
- ❖ Open source
- ❖ Extensive – Object-oriented design
- ❖ High level
- ❖ Interoperability – Simple communication with other packages

Introducing λambeq & it’s Features

λambeq Architecture

Introducing λambeq & it's Features

λambeq Architecture

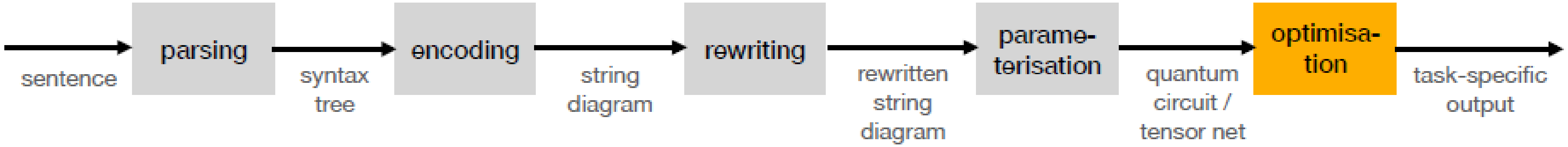
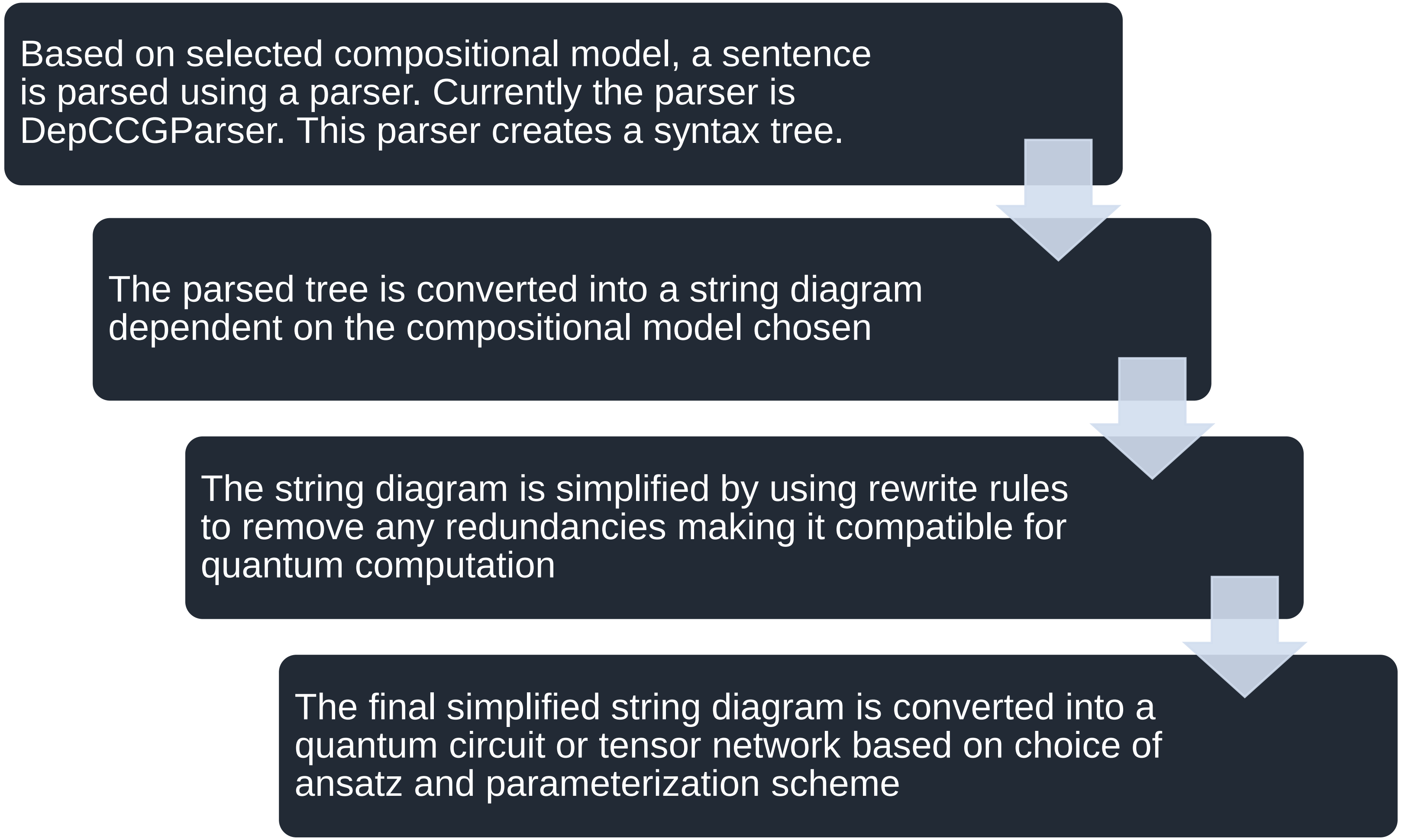


Image Source : lambeq: An Efficient High-Level Python Library for Quantum NLP

Introducing 𝜆ambeq & it’s Features

Working of lambeq :



Based on selected compositional model, a sentence is parsed using a parser. Currently the parser is DepCCGParser. This parser creates a syntax tree.

The parsed tree is converted into a string diagram dependent on the compositional model chosen

The string diagram is simplified by using rewrite rules to remove any redundancies making it compatible for quantum computation

The final simplified string diagram is converted into a quantum circuit or tensor network based on choice of ansatz and parameterization scheme

Introducing λambeq & it’s Features

Qiskit Circuit of ‘Eric hates Trev’

Introducing λambeq & it’s Features

Qiskit Circuit of ‘Eric hates Trev’

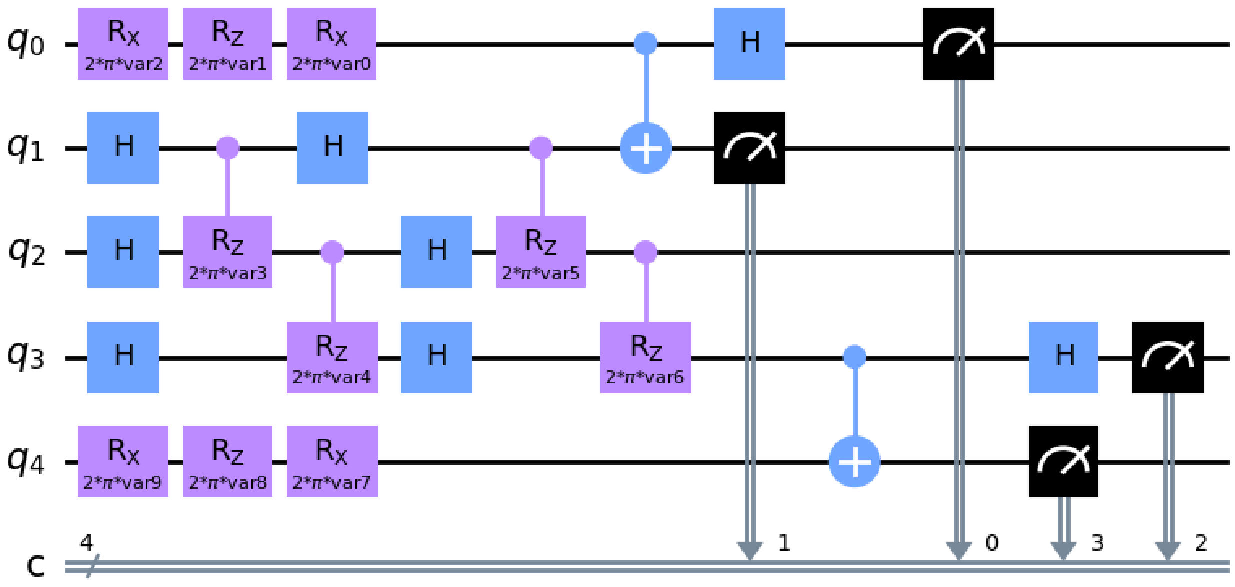


Diagram Generated using https://qnlp.cambridgequantum.com/generate.html

Introducing λambeq & it’s Features

Introducing λambeq & it’s Features Concludes

QNLP Training Process

QNLP Training Process

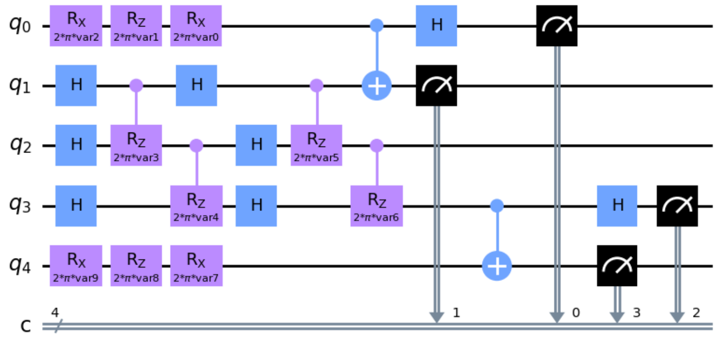


Diagram Generated using https://qnlp.cambridgequantum.com/generate.html

QNLP Training Process

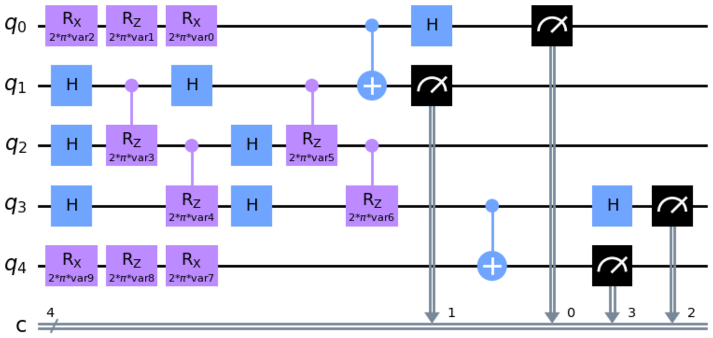
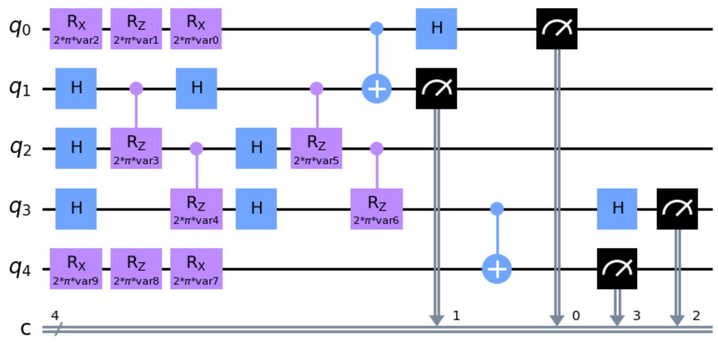


Diagram Generated using https://qnlp.cambridgequantum.com/generate.html

QNLP Training Process



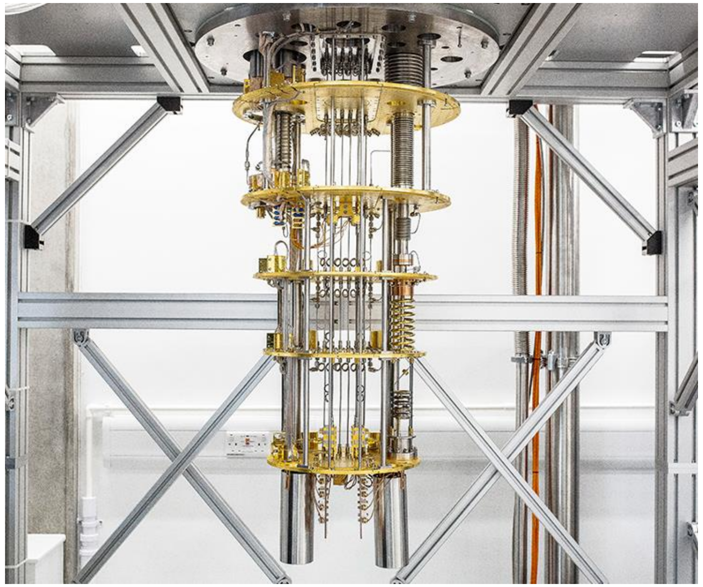
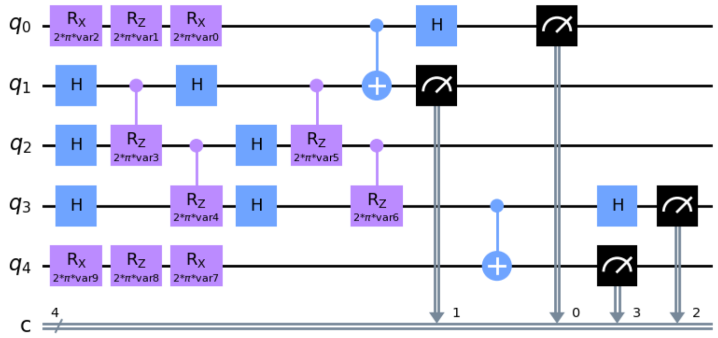


Diagram Generated using https://qnlp.cambridgequantum.com/generate.html Image from https://oxfordquantumcircuits.com/

QNLP Training Process



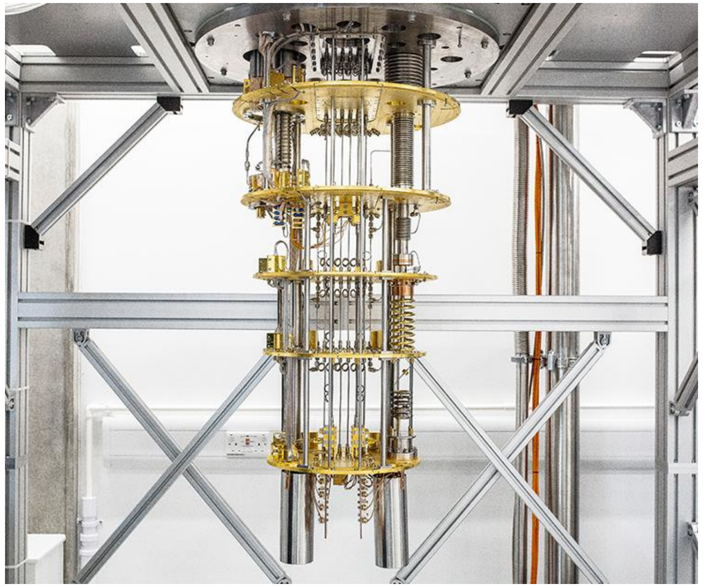
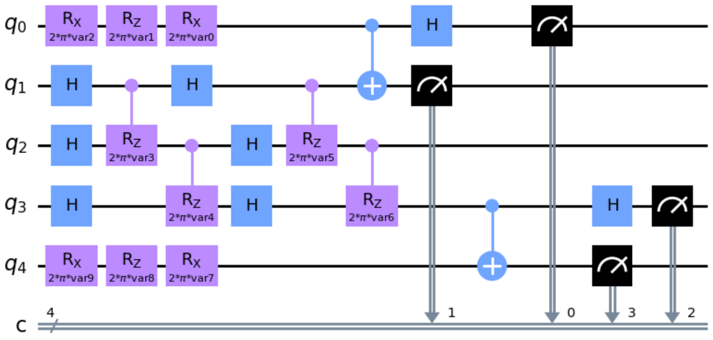
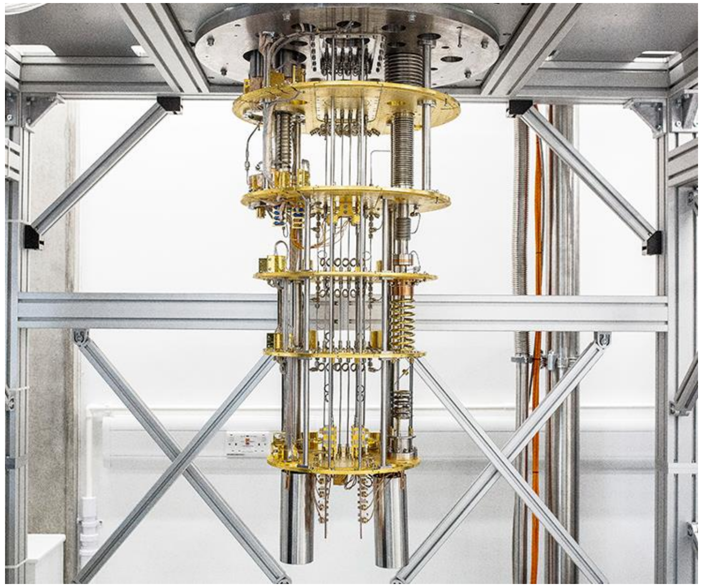


Diagram Generated using https://qnlp.cambridgequantum.com/generate.html Image from https://oxfordquantumcircuits.com/

QNLP Training Process

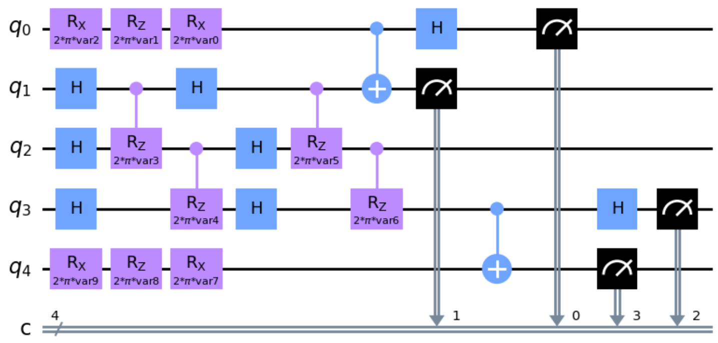


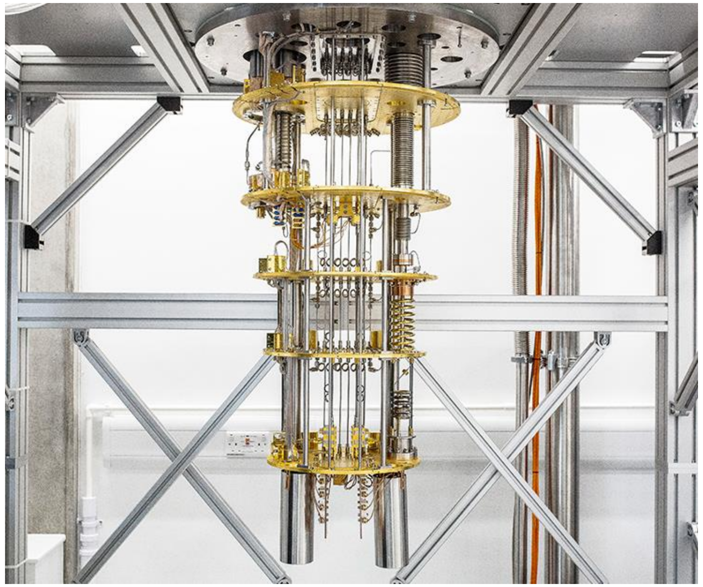


Does Eric hates Trev?

Diagram Generated using https://qnlp.cambridgequantum.com/generate.html Image from https://oxfordquantumcircuits.com/

QNLP Training Process

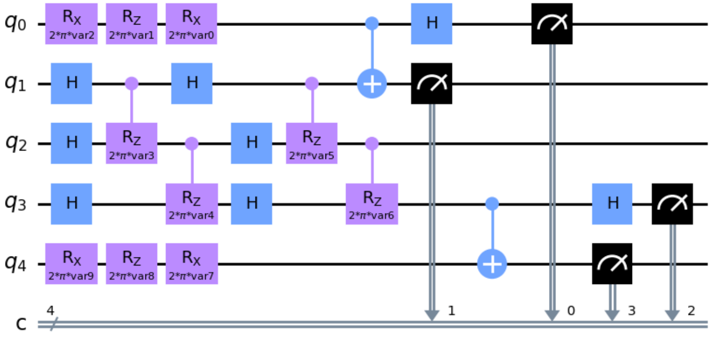


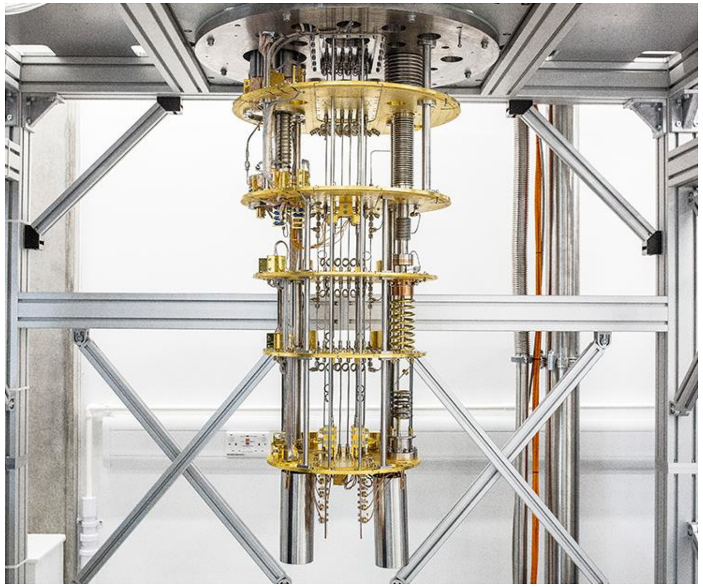


Does Eric hates Trev?

Diagram Generated using https://qnlp.cambridgequantum.com/generate.html Image from https://oxfordquantumcircuits.com/

QNLP Training Process





Does Eric hates Trev?

Yes!

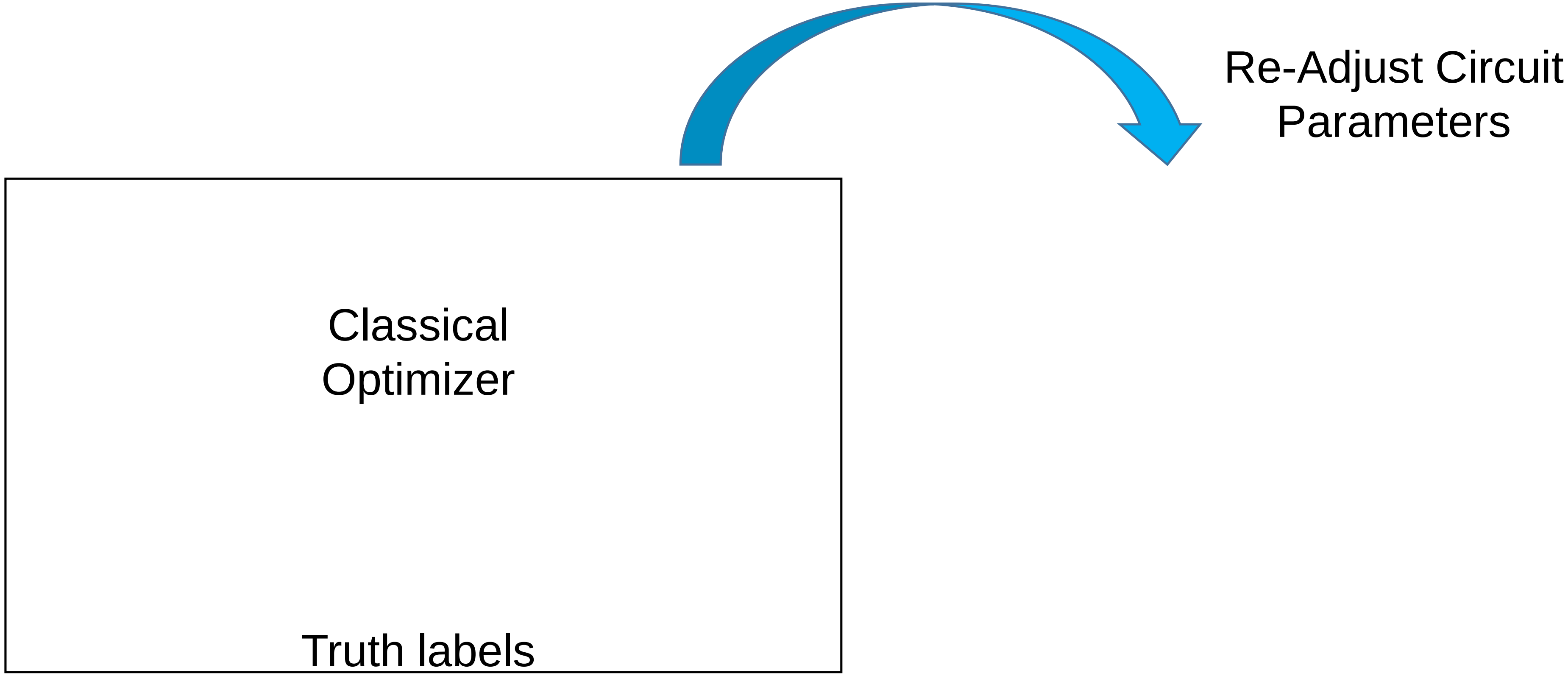
Diagram Generated using https://qnlp.cambridgequantum.com/generate.html Image from https://oxfordquantumcircuits.com/

QNLP Training Process

Classical
Optimizer

Truth labels

QNLP Training Process

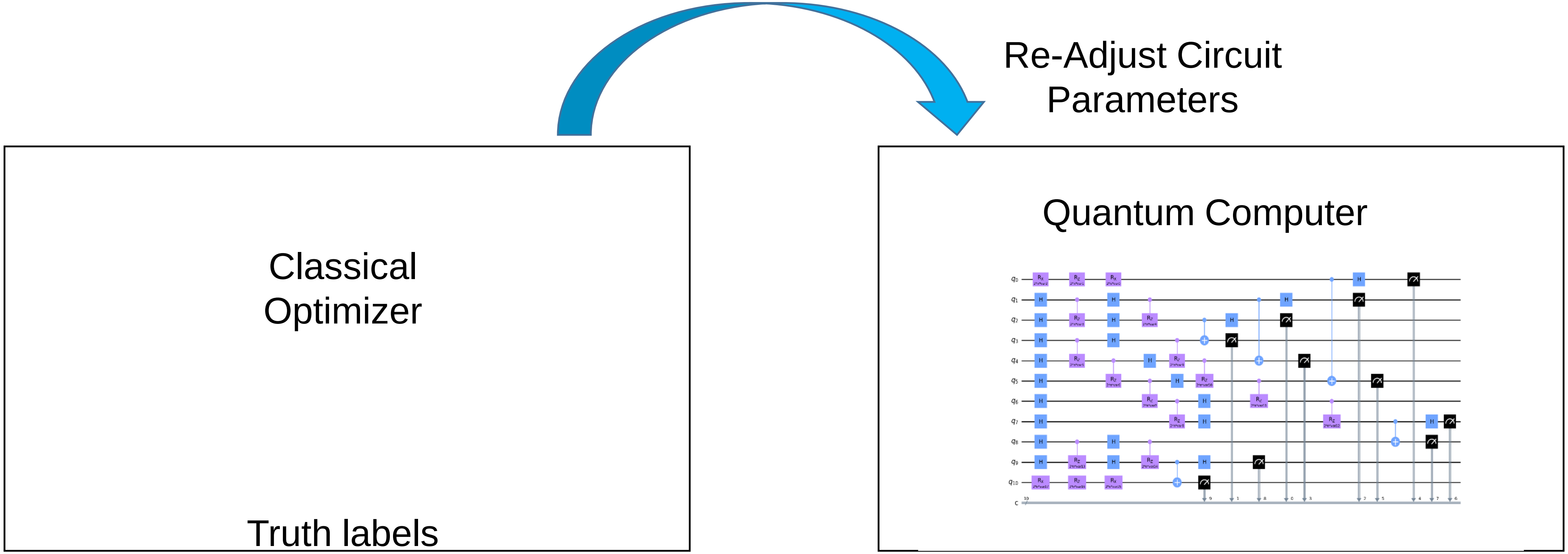


Re-Adjust Circuit Parameters

Classical Optimizer

Truth labels

QNLP Training Process



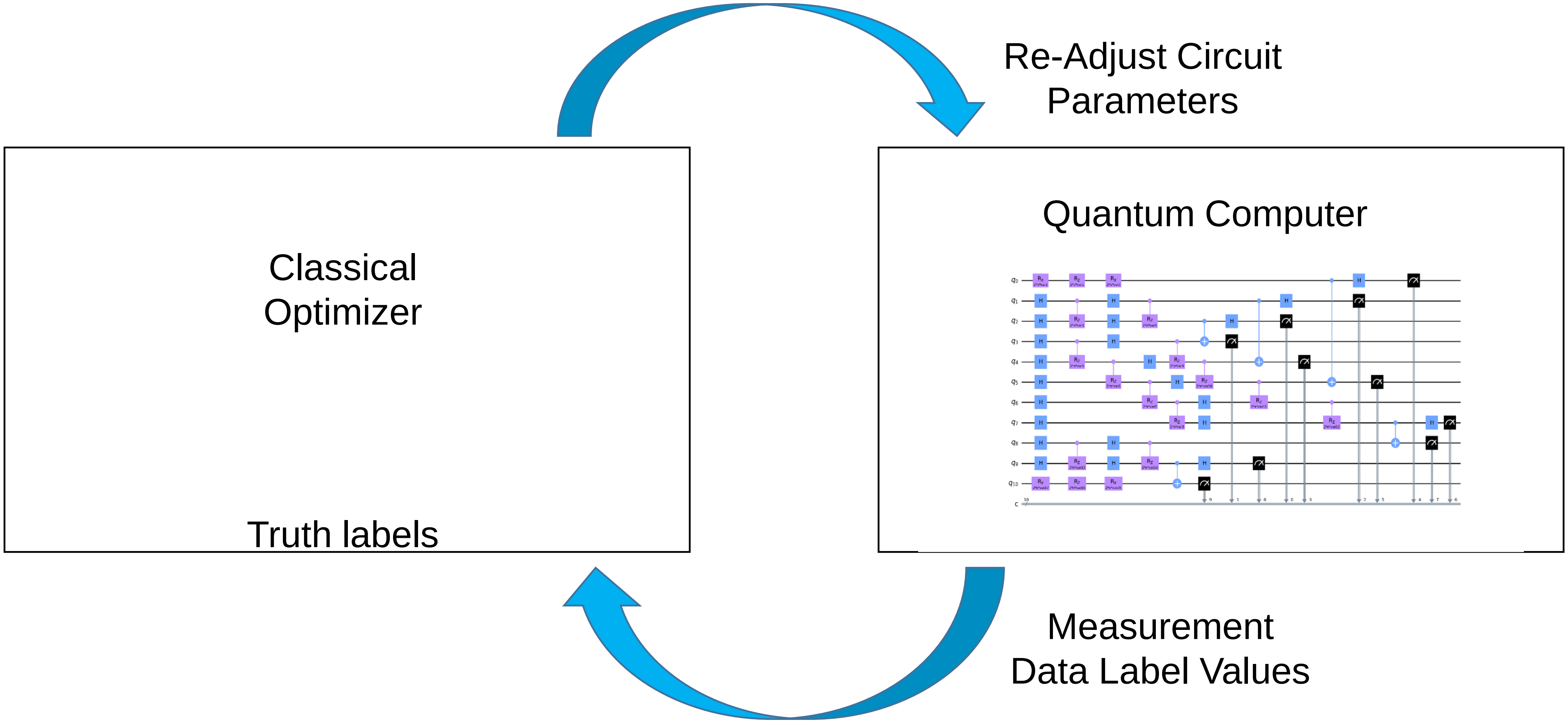
Re-Adjust Circuit Parameters

Classical Optimizer

Truth labels

Quantum Computer

QNLP Training Process



Re-Adjust Circuit Parameters

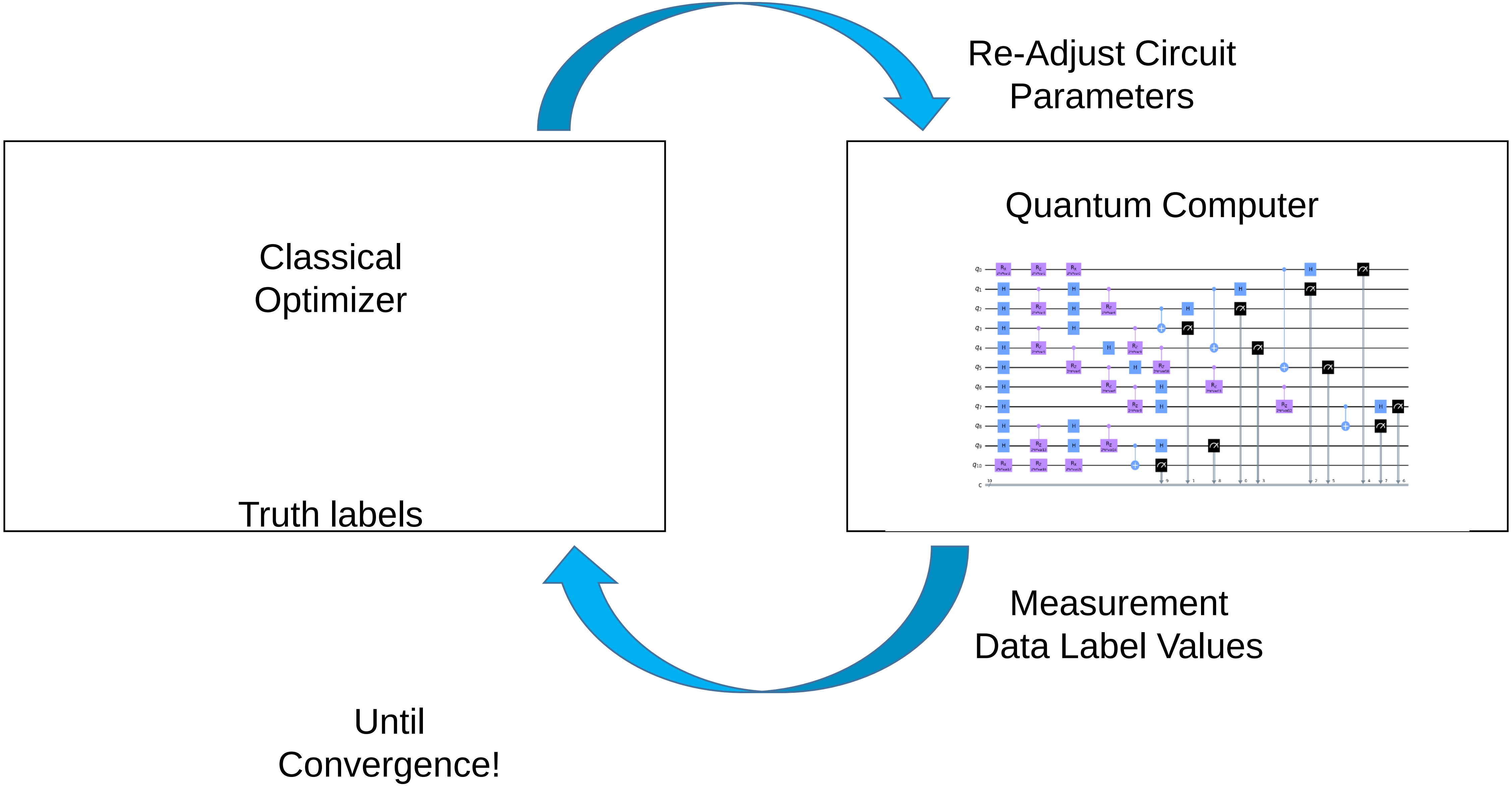
Quantum Computer

Classical Optimizer

Truth labels

Measurement Data Label Values

QNLP Training Process



Re-Adjust Circuit Parameters

Classical Optimizer

Truth labels

Quantum Computer

Measurement Data Label Values

Until Convergence!

QNLP Training Process

QNLP Training Process Concludes

Potential Use Cases/Applications of QNLP

Potential Use Cases/Applications of QNLP

- ❖ Sentiment Classification

Potential Use Cases/Applications of QNLP

- ❖ Sentiment Classification
- ❖ Quantum Language Translation

Potential Use Cases/Applications of QNLP

- ❖ Sentiment Classification
- ❖ Quantum Language Translation
- ❖ Quantum Speech Recognition Systems

Potential Use Cases/Applications of QNLP

- Sentiment Classification
- Quantum Language Translation
- Quantum Speech Recognition Systems
- QNLP for Music Research

Potential Use Cases/Applications of QNLP

Potential Use Cases/Applications of QNLP Concludes

Future Directions for Research

Future Directions for Research

- Developing the lambeq QNLP toolkit further to have more functionalities such as more compositional models and ansatz

Future Directions for Research

- Developing the lambeq QNLP toolkit further to have more functionalities such as more compositional models and ansatz

- Support for quantum friendly optimization and training modules through more research on quantum machine learning and QNLP

Future Directions for Research

- Developing the lambeq QNLP toolkit further to have more functionalities such as more compositional models and ansatz
- Support for quantum friendly optimization and training modules through more research on quantum machine learning and QNLP
- Implementation of diagrammatic differentiation for automated machine learning using lambeq

Future Directions for Research

- ❖ Developing the lambeq QNLP toolkit further to have more functionalities such as more compositional models and ansatz

- ❖ Support for quantum friendly optimization and training modules through more research on quantum machine learning and QNLP

- ❖ Implementation of diagrammatic differentiation for automated machine learning using lambeq

- ❖ Focusing on DisCoCirc algorithm to accommodate sentence to sentence grammatical structure

Future Directions for Research

- ❖ Developing the lambeq QNLP toolkit further to have more functionalities such as more compositional models and ansatz
- ❖ Support for quantum friendly optimization and training modules through more research on quantum machine learning and QNLP
- ❖ Implementation of diagrammatic differentiation for automated machine learning using lambeq
- ❖ Focusing on DisCoCirc algorithm to accommodate sentence to sentence grammatical structure
- ❖ Longer term goal is to scale up and productionization of QNLP models with the help of Quantum DevOps

Future Directions for Research

Future Directions for Research Concludes

References

- Robin Lorenz, Anna Pearson, Konstantinos Meichanetzidis, Dimitri Kartsaklis, Bob Coecke, “QNLP in Practice: Running Compositional Models of Meaning on a Quantum Computer”, https://arxiv.org/abs/2102.12846

- Dimitri Kartsaklis, Ian Fan, Richie Yeung, Anna Pearson, Robin Lorenz, Alexis Toumi, Giovanni de Felice, Konstantinos Meichanetzidis, Stephen Clark, Bob Coecke, “lambeq : An Efficient High-Level Python Library for Quantum NLP”, https://arxiv.org/abs/2110.04236

- Bob Coecke, Giovanni de Felice, Konstantinos Meichanetzidis, Alexis Toumi “How to make qubits speak”, https://arxiv.org/abs/2107.06776

- Richie Yeung, Dimitri Kartsaklis, “A CCG-Based Version of the DisCoCat Framework”, https://arxiv.org/abs/2105.07720

- Bob Coecke, Mehrnoosh Sadrzadeh, Stephen Clark, “Mathematical Foundations for a Compositional Distributional Model of Meaning”, https://arxiv.org/abs/1003.4394

References

❖ Bob Coecke, “Foundations for Near Term Quantum Natural Language Processing”, https://arxiv.org/abs/2012.03755

❖ Bob Coecke, “Compositionality as we see it, everywhere around us”, https://arxiv.org/abs/2110.05327

❖ Stephen Clark, “Something Old, Something New: Grammar-based CCG Parsing with Transformer Models”, https://arxiv.org/abs/2109.10044

❖ Bob Coecke, Aleks Kissinger, “Picturing Quantum Processes”, Cambridge University Press, 2017

❖ Joachim Lambek, “From Word to Sentence: A Computational Algebraic Approach to Grammar”, Polimetrica s.a.s., 2008

❖ PennyLane for Quantum Machine Learning, https://pennylane.ai/qml/index.html

❖ John van de Wetering, “ZX-calculus for the working quantum computer scientist”, https://arxiv.org/abs/2012.13966

Thank you so much!