Degrees of Entanglement and Contextuality in Quantum Natural Language Processing

Tilen Gaetano Limbäck-Stokin[1], Kin Ian Lo[1], and Mehrnoosh Sadrzadeh[1]

[1]Department of Computer Science, University College London

Abstract

This paper regards an interdisciplinary area of study between physics and linguistics, and uses a setting which has been dubbed Quantum Natural Language Processing (QNLP). So far in QNLP, the focus has been on variational quantum circuits (VQCs) and a succinct translation between linguistic structures and quantum circuits. We add two procedures to QNLP that connects its circuits to quantum resources. These are (i) the degree of entanglement of the semantic tensors, and (ii) the degree of contextuality of the quantum system. We experiment with a large dataset designed for a downstream language task: pronominal anaphora resolution and show that whenever there is information flow between a pronoun and a noun, the degrees of entanglement and contextuality of the corresponding circuits increase. This suggests that semantically informative linguistic relations, such as anaphors, are associated with measures of correlations in quantum systems. To our knowledge, this is a connection that has not been established before. The current paper relies on QNLP, which needs syntax to build VQCs. A desirable result would be finding a direct connection between semantic relations, beyond what is syntactically implied.

1 Introduction

This field of quantum natural language processing (QNLP) has recently garnered interest from several communities due to their discovery that natural language has similarities to quantum mechanics [1]–[3]. The QNLP framework receives sentences of natural language as inputs and turns them into quantum circuits, built by induction over their syntactic and semantic structure. This framework has been implemented in an open source software [4].

The pipeline works well and almost mechanically converts any sequence of words into a quantum circuit. What is less clear, is the conceptual connections between quantum resources such as entanglement, coherence, and contextuality to language constructions. These resources have been understood in quantum computation and communication and have successfully been taken advantage of. For example, in the Schor [5] and Grover [6] algorithms the use of entanglement is intentionalised and well understood [7] [8]. These connections have even led to a new field, that of quantum resource theory [9].

Despite its natural translation between linguistic structures and quantum circuits, QNLP has not yet been connected to quantum resource theory. This paper fills in this gap and builds two bridges. In particular, we ask: "How do or can quantum resources arise in language through quantum computing and what can we infer about language in their presence or lack thereof?". We provide answers to the above by looking at degrees of contextuality and entanglement.

Contextuality is a defining phenomena of quantum theory. It was first formalised via a specific subclass of hidden variable theories, namely non-contextual models [10]. These theories try to explain the behaviour of a quantum system by the lack of knowledge of an underlying classical system, called hidden variables, which by uncovering them the 'true' deterministic nature of the outcomes of the system would reveal themselves. In particular, contextuality states that even identical observables have different values depending on their context, usually different arrangements of measurements. This is intimately linked with efficient quantum computation, and potentially even advantage. Particularly in the context of magic state distillation, but also more generally in many quantum algorithm [11] [12] [13].

In our scenario, entanglement is related to certain models of contextuality and just as entanglement implies superposition, contextuality implies entanglement, but not the other way around [14]. This can be understood as entanglement providing the necessary basis for the contextual dependence of outcomes of measurements that give rise to contextuality.

There have been several attempts to find meaningful connections between contextuality and linguistics [15] [16]. Despite their clear novelty and value, these approaches have a restrictive nature since they fit the linguistic phenomena into a model of contextuality, literally modelling a linguistic phenomena over a contextual phenomena such as the PR-Box or the Bell experiment. Our approach is more straightforward and general: we translate the linguistic phenomena into a variational quantum circuit using the general pipeline of QNLP and then find ways of measuring the degree of contextuality and of entanglement of the tensors therein. A slight issue with QNLP is its reliance on syntactic structure. Other linguistic resources such as bags and sequences of words also exit and are exploited by large language models. Using these more general approaches to language and connecting the same phenomena to quantum resources constitutes a natural future direction.

2 Background

In this section we go through the pipeline of QNLP, which itself is based on a model of meaning called Distributional Compositional Categorical (DisCoCat). DisCoCat maps the syntactic structure of language to semantic embeddings in a finite dimensional vector space [17]. So first we introduce DisCoCat, then the translation between its semantic constructions and variational quantum circuits.

2.1 Syntactic Structure

In the distributional model, meaning of words is represented using vectors in a high-dimensional vector space. The orthogonal basis of this space are *context* words, which define all other words [18]. This also provides a notion of distance between words, the idea being similar words will usually be surrounded by the same basis words and should have a small distance between each other. The symbolic theory of language, modelled by formal grammar and formal semantic theories, uses different mathematical formalisms. Amongst these, categorial grammars such as Lambek Calculus [19], pregroup grammars [20] and Combinatory Categorial Grammar [21]–[23] are based on a set of algebraic axioms and rules and foster a transparent interface between rules of syntax and semantics. CCG has a robust large scale parser that has been trained using deep neural network transformer technology [4], hence it is a natural choice for this paper.

CCG consists of a set of types and inference rules. The basic CCG has the set of atomic types $\{N, NP, S\}$ representing nouns, noun phrases and sentences, and two function types: forward and backward application. The function type $A\backslash B$ outputs type B given A is to the left and B/A outputs B given A is on the right. The formal definitions of these rules are given below [1].

$$< \ \frac{\alpha : X/Y \quad \beta : Y}{\alpha\beta : X} \qquad > \ \frac{\alpha : Y \quad \beta : Y\backslash X}{\alpha\beta : X} \tag{1}$$

Consider the following example sentence "*Alice likes Bob*". The syntactic structure of this sentence is derivable using both of the inference rules [2].

$$\dfrac{\dfrac{\text{Alice}}{NP} \quad \dfrac{\dfrac{\text{likes}}{(S\backslash NP)/NP} \quad \dfrac{\text{Bob}}{NP}}{S\backslash NP}<}{S}> \tag{2}$$

This variant of CCG restricted to the rules above has an expressive power equivalent to context free grammars [24]. CCG has other rules such as type raising T and composition B, with forward and backward application variants for both. The formal definitions of these rules are given below [3].

$$B_< \ \frac{\alpha : X/Y \quad \beta : Y/Z}{\alpha\beta : X/Z} \qquad B_> \ \frac{\alpha : Z\backslash Y \quad \beta : Y\backslash X}{\alpha\beta : Z\backslash X}$$
$$T_< \ \frac{\alpha : X}{\alpha : T/(X\backslash T)} \qquad T_> \ \frac{\alpha : X}{\alpha : (T/X)\backslash T} \tag{3}$$

Composition is used for phrase formation, e.g. in auxiliaries verbs. As an example consider the derivation below [4].

$$\dfrac{\dfrac{\text{Alice}\ \overline{NP} \quad \dfrac{\dfrac{\text{might}\ \overline{(S\backslash NP)/VP} \quad \text{like}\ \overline{VP/NP}}{(S\backslash NP)/NP}\,{>}\mathbf{B} \quad \text{Bob}\ \overline{NP}}{S\backslash NP}\,{<}}{S}\,{>} \tag{4}$$

Additionally, type raising allows for non local dependencies such as those seen in relative clause formation. For instance see below [5].

$$\dfrac{\dfrac{\text{The}\ \overline{NP/N} \quad \text{company}\ \overline{N}}{NP}\,{>} \quad \text{which}\ \overline{(NP\backslash NP)/(S/NP)} \quad \dfrac{\dfrac{\text{Google}\ \overline{NP}}{S/(S\backslash NP)}\,{>}\mathbf{T} \quad \text{bought}\ \overline{(S\backslash NP)/NP}}{S/NP}\,{>}\mathbf{B}}{NP\backslash NP}\,{>}$$
$$\dfrac{}{NP}\,{<} \tag{5}$$

In all of the above, the subscripts $_>$ and $_<$ let us know that an inference rule is forward or backward applied, respectively. Moreover, the CCG can be endowed with a a set of other additional rules to increase its expressivity. For example, there are rules for coordination Φ and cross composition $B_\times$. Coordination is used for conjunctives such as "and, or, because". Cross-composition is applied to complex situations such as the treatment of phrasal verbs, as demonstrated below [6].

$$\dfrac{\dfrac{\text{Alice}\ \overline{NP} \quad \dfrac{\text{put}\ \overline{(NP\backslash S)/NP} \quad \text{on}\ \overline{(NP\backslash S)\backslash(NP\backslash S)}}{(NP\backslash S)/NP}\,{<}\mathbf{B}_\times \quad \text{her hat}\ \overline{NP}}{S/NP}\,{>}\mathbf{B}}{S}\,{>} \tag{6}$$

Finally, let us look at the CCG derivation for the sentence "Alice likes and trusts Bob", as seen below [7].

$$\dfrac{\dfrac{\text{Alice}\ \overline{NP} \quad \dfrac{\text{likes}\ \overline{(NP\backslash S)/NP} \quad \text{and}\ \overline{CONJ} \quad \text{trusts}\ \overline{(NP\backslash S)/NP}}{(NP\backslash S)/NP}\,{<}\Phi{>} \quad \text{Bob}\ \overline{NP}}{NP\backslash S}\,{>}}{S}\,{<} \tag{7}$$

2.2 A String Diagrammatic Semantics for CCG

In order to provide the CCG with a quantum circuit semantics, we first translate it to a diagrammatic form. We then use the conversion between the diagrams and quantum circuits developed in [1] and implemented in [4].

Each string diagrams consists of wires and boxes. The wires depict the formulae. The boxes are the rules. The diagrammatic forms of the forward application, composition, and type raising rules are given in [1].

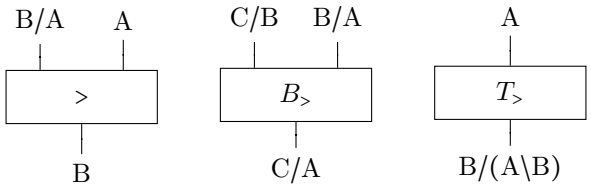

Figure 1: CCG diagrams and corresponding pregroup diagrams for three main inference rules of a CCG rule set.

For instance, the string diagrammatic form of the sentence "Alice likes Bob", follows a similar section as the derivations in the previous section as follows [2].

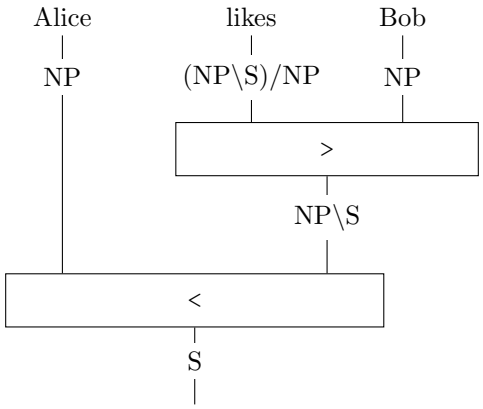

Figure 2: Diagrammatic representation of a CCG parse.

The next step is to convert the CCG diagrams into a more compact form, also known as *string diagrams*. In these diagrams, the only connection between the types is the tensor product. The conversion works by replacing the left and right slashes with left and right adjoints that mark the order of the slashes. The conversion map is called F and is given below in equation [1], where for simplicity we also assume that the words with atomic types N and NP get their semantics from the same vector space (1).

$$\begin{aligned} &F(S) = s, \quad F(N) = F(NP) = n \\ &F(X\backslash Y) = F(X)^r \; F(Y), \\ &F(X/Y) = F(X) \; F(Y)^l \end{aligned} \tag{1}$$

The cancellations between the adjoint types is depicted by a cup or a cap. Let us see how this looks on a very basic example of a single CCG inference rule, for the three basic rules of application, composition, and type raising in their forward form [3].

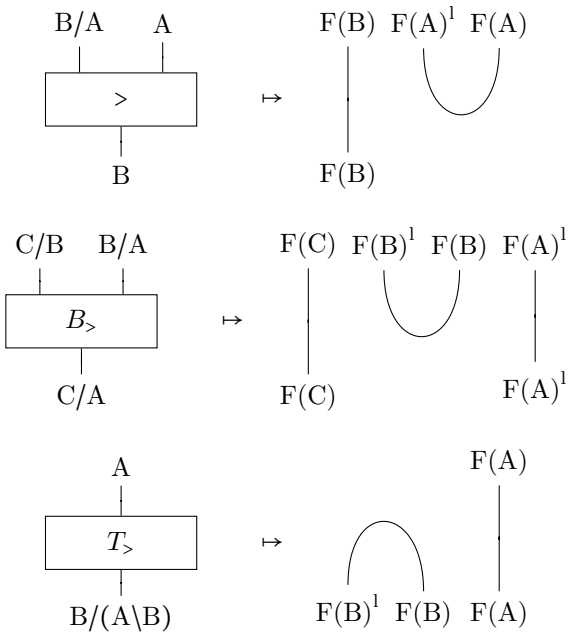

Figure 3: CCG diagrams and corresponding pregroup diagrams for three main inference rules of a CCG rule set.

Performing this operation repeatedly on the CCG diagram, results in obtaining string diagrams for syntactic parses of sentences. For example, applying the above conversions provides us with the following diagram for the sentence “Alice likes Bob” [4].

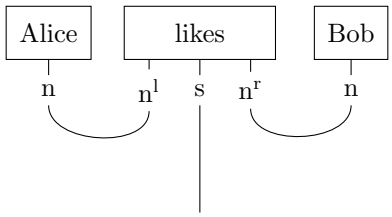

Figure 4: Example of DisCoCat string diagram, where the transitive verb ’likes’ is an order 3 tensor and is contracted with the subject and object nouns ’Alices’ and ’Bob’, both of which are order 1 tensors. This results in a vector representing the sentence semantics.

The specific labelling is not particularly important, so long as the order of cancellation is derived from the grammatical rules of the CCG. For a more complex example, see below for the diagrammatic form of “The company that Google bought” [5].

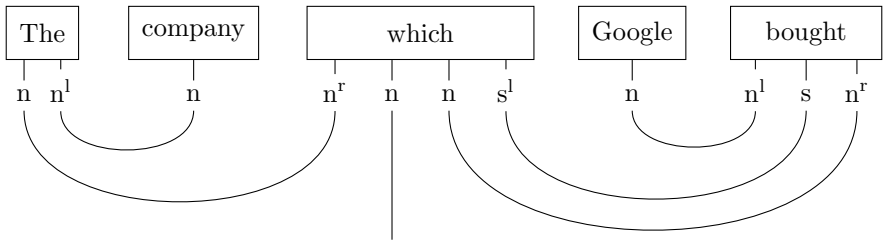

Figure 5: DisCoCat string diagram for the sentence ”The company which Google bough”, where the pronoun ’which’ is an order 4 tensor and is contracted with the subject and object nouns ’Google’ and ’company’, both of which are order 1 tensors. This results in a vector representing the sentence semantics, signified by the single outgoing wire.

2.3 A Tensor Semantics for CCG

A tensor semantics was developed for the CCG in [25], [26]. In this semantics, a word with an atomic CCG type is assigned an atomic vector space. A word with a CCG function type of n arguments is assigned an n-ary linear map $\mathbb{W}$ from the spaces of the arguments to the space of the output (2), that is:

$$f: V_1 \times V_2 \times \cdots \times V_n \to V_{n+1} \tag{2}$$

As our spaces are finite dimensional vector spaces, an n-ary map becomes equivalent to a tensor of rank $n+1$, with n spaces for the arguments and one extra space for the output of the map. We have $f \cong \mathbb{T}^{w}_{i_1 i_2 \cdots i_{n+1}}$, where we have (3).

$$\mathbb{T}^{w}_{i_1 i_2 \cdots i_{n+1}} \in V_1 \otimes V_2 \otimes \cdots \otimes V_{n+1} \tag{3}$$

For example, a noun $Noun$ with an atomic type NP is assigned an atomic space $\mathbb{T}^{w}_{1}$, i.e. it is a tensor of rank 1. An adjective Adj is assigned a linear map, i.e. it is equivalently a rank 2 tensor $\mathbb{T}^{Adj}_{i_1 i_2}$. Similarly, an intransitive verb iTv is a linear map, i.e. a rank 2 tensor $\mathbb{T}^{iTv}_{i_1 i_2}$. A transitive verb Tv is a bilinear map, i.e. a rank 3 tensor $\mathbb{T}^{Tv}_{i_1 i_2 i_3}$ and so on. Words with more complex types such relative pronouns become rank 4 tensors. Ditransitive verbs are also rank 4 tensors. For a general formula of type-driven tensor representations, see [26].

The semantic representation of a string of words $w_{l_1} w_{l_2} \cdots w_{l_m}$, each with a tensor representation $\mathbb{T}_{i_1 i_2 \cdots i_n}$ is a series of tensor contractions. The contractions are mapped from the derivation rules of CCG. As examples, the semantic representations of an adjective noun phrase and of an intransitive sentence are given below [8].

$$\dfrac{\mathbb{T}^{\text{Adj}}_{i_1 i_2} \quad \mathbb{T}^{\text{Noun}}_{i_1}}{\mathbb{T}^{\text{Adj-Noun}}_{i_2}}{>} \qquad\qquad \dfrac{\mathbb{T}^{\text{Noun}}_{i_1} \quad \mathbb{T}^{iTv}_{i_1 i_2}}{\mathbb{T}^{\text{Noun-}iTv}_{i_2}}{<} \tag{8}$$

Below is an example of the tensor semantics of the transitive sentence "Alice likes Bob" is below [9].

$$\dfrac{\mathbb{T}^{\text{Alice}}_{i_1} \qquad \dfrac{\mathbb{T}^{\text{likes}}_{i_1 i_2 i_3} \quad \mathbb{T}^{\text{Bob}}_{i_3}}{\mathbb{T}^{\text{likes-Bob}}_{i_1 i_2}}{>}}{\mathbb{T}^{\text{Alice-likes-Bob}}_{i_2}}{<} \tag{9}$$

2.4 Parametrised Quantum Circuits for CCG

The tensors can be learnt by machine learning over parametrized quantum circuits. For each tensor, a set of different quantum circuits exists. Choosing one relies on fixing a set of initial assumptions, called '*ansatze*'. A well known ansatz is generated using the Instantaneous Quantum Polynomial (IQP) model [27]. Other ansatz also exist, for instance Sim14 and Sim15 from the Sim family [28]. After fixing an ansatze, we build a circuit for each CCG derivation following the conversion steps proposed in [1]. These steps are automated in a software package called *Lambeq*, documented in [4].

The first step of the conversion is the same for all ansatze, a number of qubits is assigned to each atomic type, then each wire with labels of that type is all allocated the same number of qubits. For instance, suppose we assign one qubit to the atomic type NP, then all wires with an NP label will be one qubit. From the next step on, we follow the particular instructions of each ansatze. Each string diagrammatic box with k wires is sent to a tensor of rank k. We call these tensors quantum circuits and their size is given by the total number of qubits assigned to each of the box's outgoing wires. The IQP ansatze creates a commuting ladder of $n-1$ interleaved controlled rotations between the qubits. Then it assigns a Hadamard gate to each qubit. The circuit for $n = 3$ is given in the left hand side of Figure [6]. This generalises to any number of qubits except for $n = 1$, the single qubit case. Here the IQP instead applies a sequence of rotation gates that sweeps the entire Bloch sphere, e.g. X and Z; see the right hand side of Figure [6].

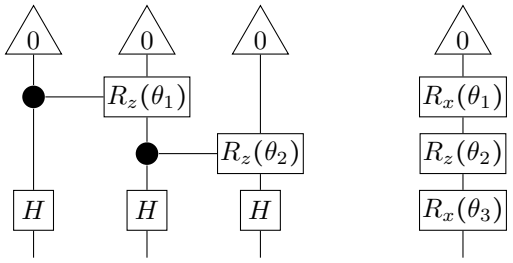

Figure 6: The left diagram is the IQP conversion of a box assigned 3 qubits. Consits of a layer of Haddamards and the ladder of control Z rotations. The right diagram is the case for a box of 1 qubit, consisting of X, Y, and Z rotations performed in sequence.

Here we see that for an n qubit box IQP introduces $n-1$ tunable parameters, except for $n=1$ where it introduces 3 parameters. The other two schemes are similar, applying a commuting ring of interleaved controlled gates. This means that for a control qubit i, we apply an operation to the $i-1 \bmod n$ qubit. This is then repeated a second time such that the second ring is in the opposite direction. Rather, for each control qubit i, we apply a controlled operation on the $i+1 \bmod n$ qubit. This two part sequence of rotations and controlled rings forms one layer of the ansatze. The difference in the schemas here is in the controlled operation. The Sim14 ansatze uses controlled X rotations, which has parameters, while the Sim15 ansatze uses standard CNOT gates, which does not have any parameters [7].

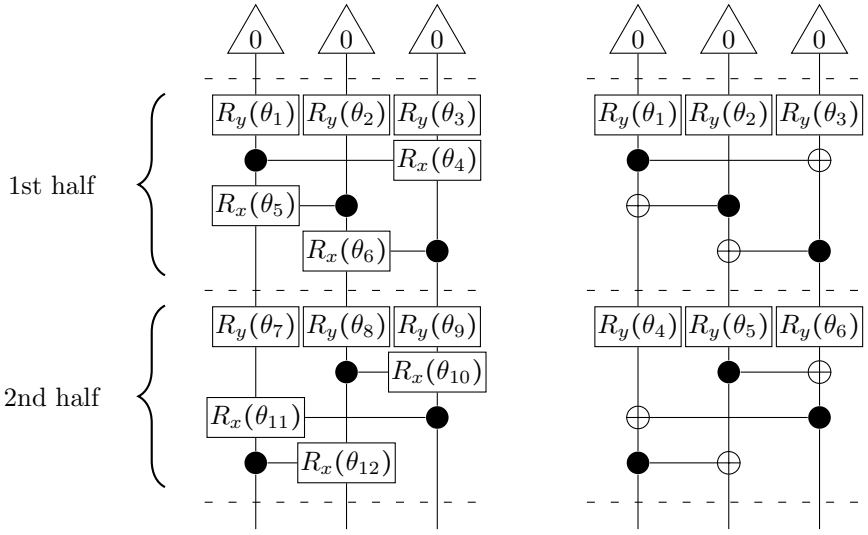

Figure 7: One layer of the Sim14, left, and Sim15, right, ansatze for three qubits. Split into the first and second sublayers.

In this case we can trivially note that these ansatze have a much higher entangling power, but also introduce more depth having two sub-layers. In the case of Sim14 this results in $4n$ tunable parameters for an n qubit box, which is computationally more expensive but also more expressive. Similarly, for Sim15, there are $2n$ tunable parameters. Overall, this results in a much more expensive learning process.

Here it is important to note the hyperparameters for the ansatze, namely the number of qubits for a wire and the number of layers to apply. As we saw earlier, when converting a sentence to a string diagram each wire has one of three types assigned to it. In our experimentation we assign 1 qubit to each type in the grammar. Another hyperparameter is the number of layers to apply. Everything we have seen so far has been a single layer, so if we chose 2 layers the above examples would be applied twice. The number of layers is kept at 1. Let us look at the string diagram in [11] after it is converted using IQP ansatz with the just mentioned hyperparameters, resulting in the following quantum circuit [8].

Note that there is an additional Haddamard gate applied to each wire at the beginning. This is done with the intent to change the basis to the desired one used in the backend for the universal unitary gate set that Lambeq uses, which is derived from Qiskit. This is necessary when running circuits on actual quantum hardware as they choose specific gate sets by which they implement all operations.

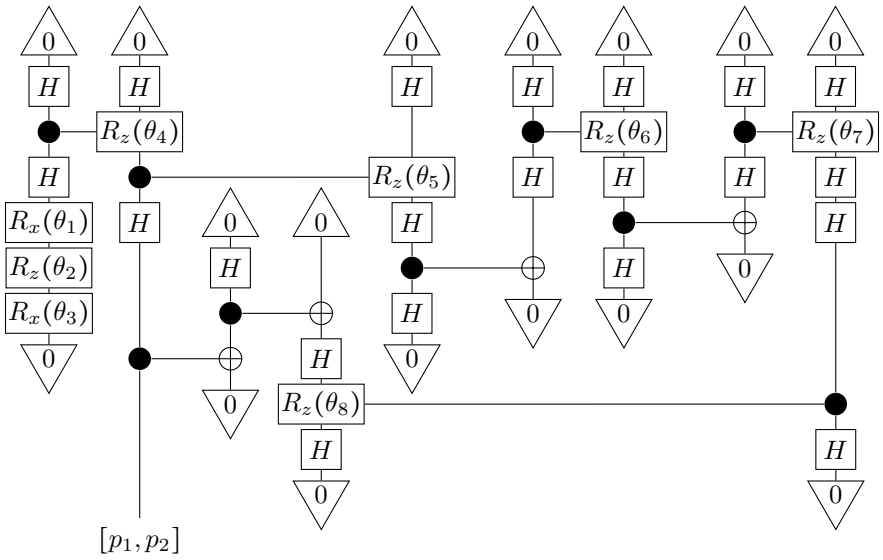

Figure 8: Quantum circuit ansatz generated with IQP from the string diagram in [11], with a single open wire outputting a probability distribution $[p_1, p_2]$

Furthermore, the wires ending in 0 labelled upside down triangles are *post-selections*. When getting results from a quantum circuit it is run several times and measured, storing the outcome results to reconstruct the distribution of the output state later on. This is necessary as there is inherent randomness in the measurements, so in order to reconstruct the output state we have to 'sample' many outputs to get an accurate set of measurements representative of the output state. Post-selections are useful when we want to condition our measurements on the outputs of specific wires. Essentially, we only keep the results when the post-selection is satisfied, the qubits are measured and the desired output is observed. In this example, it would be when the post-selected qubits are measured and return the $|0\rangle$ state. In the case of QNLP we are only interested in the wires containing valid semantic information of the sentence, given the correct grammatical reduction. This proper grammatical reduction occurs precisely when the post-selection is satisfied, the corresponding grammatical ancilla qubits are measured as $|0\rangle$ in our case. The runs where the grammatical reduction is not valid, guaranteed by post-selection, are discarded and not included when reconstructing the final joint probability distribution. Meanwhile, for the runs we do keep, the open wires are always measured and the measurement results are renormalised with respect to the post-selection. This explains why post-selection is expensive, it will take more runs to satisfy stricter post-selections, which needs to occur many times to have an accurate estimate of the open wire.

3 Methodology

The CCG parser does not have the capability of handling pieces of text, or rather parsing, the connections between the different sentences within it. So we had to adopt a slightly modified approach. First, we generated a separate parse for each sentence and created its string diagrams. For instance, for the piece of text Alice likes Bob. He is happy." we obtain the string diagram in Figure [9].

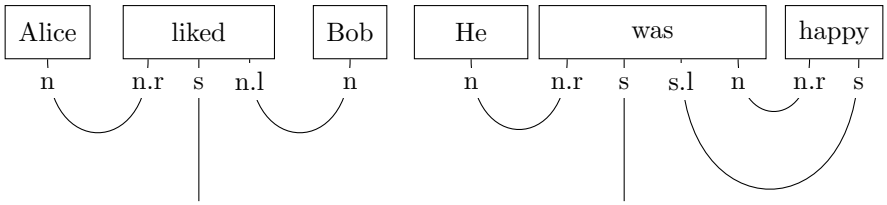

Figure 9: The string diagram for sentences *Alice liked Bob.* and *He was happy.* brought into the same tensor space.

We then had to manually connect the referent with the pronoun, by first changing the types of the pronoun so it can interact with its referent noun, then combining the two sentence output wires via the Forbenius operation over the tensors. This operation is referred to as a "spider" in string diagrams. Moreover, the just mentioned Lambeq package does not support the manual manipulation of diagrams, hence the DisCoPro library [29] built on top of it was used for this step. The diagram of our example text after connecting the referent to the pronoun and composing the sentence outputs is shown below [10].

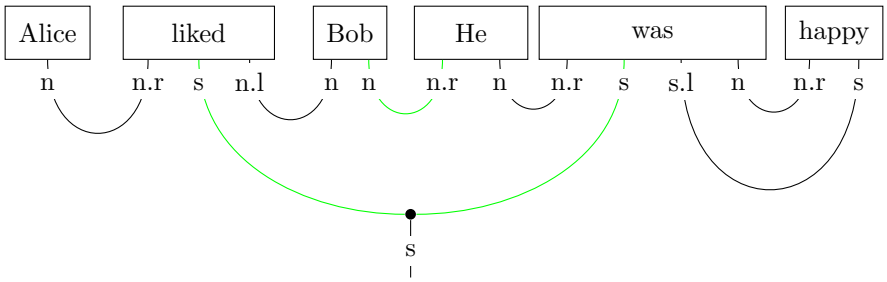

Figure 10: The string diagram for sentence *Alice liked Bob. He was happy.* with *Bob* connected to *He* and the individual sentences joined via a spider operation. The connections between the words represent the specific order in which; they are composed and contraction occurs, annotated with their corresponding types.

3.1 Diagram Rewriting

Another important step is producing simplified yet equivalent diagrams to make computation later on more efficient. String diagrams are very expensive to compute and become very large very quickly as we are working with high dimensional tensor spaces, so it is ideal to simplify the diagrams. This is done primarily by: removing cups and caps by 'straightening' them and 'untangling' wires that cross, called swaps. Cups and caps in the string diagram induce additional post-selection, which are expensive operations that will be discussed later on. Similarly, swaps in the string diagram simply become the swap operation standard in quantum computing which is also expensive.

First we connect the anaphora from the second sentence to the respective referent in the first sentence by composing the sentences on top of each other, which avoids introducing unnecessary swaps. As mentioned earlier this step is done via the DisCoPro library. This is useful as simply performing the connection with the sentences adjacent to each other can create many swaps. Our current example does not suffer this issue, but consider if we change the pronoun to "She" and tried connecting it to "Alice" in [10], we would introduce at least one additional swap. At this point we would also additionally compose the sentence outputs.

The next step is using what Lambeq calls a rewriter to handle cups. There are many different types of rewriters, or rather methods for rewriting, but they can all intuitively be though of as rearranging the boxes in such a way that the wires all get straightened. In particular, we use the aptly named `RemoveCupsRewriter` to eliminate cups and caps, which essentially 'flips' boxes with a cup wire on top such that the wire is straightened.

This also induces a dagger operation on the flipped cups, which is simply the standard Hermitian operator from quantum mechanics. Continuing off the above example, below is the simplified diagram [11], which has a minimised number of swaps and cups in the diagram, greatly increasing the diagram efficiency by reducing the number of expensive operations and circuit width. Note that, although rewriting does remove the amount of post-selection, it also increases circuit depth, but in our case this is a worthwhile trade off.

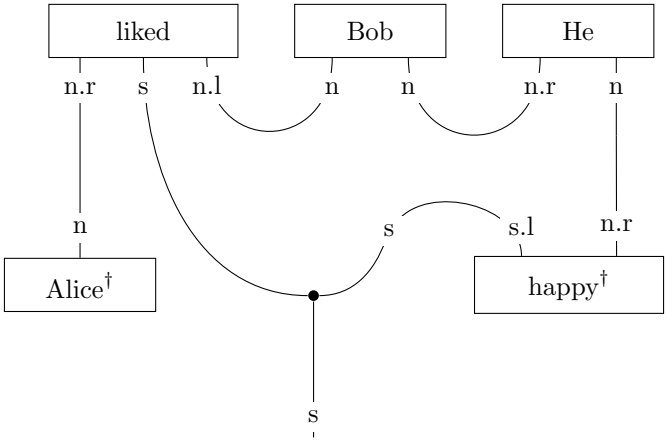

Figure 11: The minimised string diagram of the original from figure [10]

After simplifying the diagrams, we turn them into parametrised quantum circuits. The procedure is explained above. The novel anaphora constructions use the already existing rules and do not need a special treatment. Only note that the presence of anaphors increase the number of tensors in the sentence, which in turn leads to more entanglement after applying the Ansatze.

4 Experiments

We train the quantum circuits generated for the pieces of text of the dataset developed in [3], which has 16,400 entries consisting of two sentences each. In each entry, the first sentence has two nouns and the second sentence a single pronoun. This pronoun refers to one of the nouns of the first sentence.

For each piece of text, we consider two different variants. The first one is about how to connect the pronoun to the referent. Here there are three options: do not connect them at all, connect the right referent to the pronoun, or connect the wrong referent to the pronoun. The second choice is about how to connect the sentences of the text. Again two options arise: do not connect them at all or connect them with a 'spider'.

One might wonder when the other options may be useful. For the different pronoun-referent connection, these diagrams are used to train a binary classifier, where both the 'right' and the 'wrong' answers are used as labels in the training set. As for the sentence composition, the spider has a natural interpretation as a dot product in the tensor space, which has semantically meaningful results from the distributional background. As for the options of not connecting referents to pronouns or not composing sentences, it may seem meaningless to do so, but this has interesting consequences later on as in general fewer connections usually imply less entanglement. This provides a potential insight where the presence of entanglement, or rather, contextuality may have correlation with better results in this scenario.

We experiment with combinations of these different choices to produce different models of string diagrams. This results in four such models: open wire without a pronoun-referent connection, open wire with a pronoun-referent connection, spider without a pronoun-referent connection, and spider with a pronoun-referent connection. Open wire here just means that we do not compose the two outputs sentences into a singular output wire. The goal

is to explore the degrees of contextuality, and moreover entanglement, arising from these structurally varying string diagrams.

The process of converting a sentence pair to a parametrised quantum circuit is performed for all 16400 sentence pairs in the dataset with a training, validation, and testing split of 0.6/0.2/0.2. This is repeated four times for the different combinations of open wire or spider and with or without a pronoun-referent connection. This is done with the IQP ansatze for all four and using the Sim14 and Sim15 ansatze for the standard case of a present pronoun-referent connection and composed sentence outputs. This means we have to generate six sets of circuits for the 16400 sentences. It is important to note that during the pipline of converting the sentences to circuits by using the Bobcat parser and applying an Ansatz, some sentences do not get successfully translated. Although they are very few cases, this is usually due to the ansatze encountering inconsistent types as not all valid Bobcat parses result in a valid DisCoCat derivation and hence diagram.

4.1 Labels

For each circuit generated, a corresponding label is also produced, which is used for training. The label is based on whether the pronoun is connected to the right or wrong referent, which is chosen at random during the data generation step, so that there is a balance of false and positive results. Spider based models have a single outgoing wire, or rather a single qubit output. The output of the circuit is a probability distribution $[p_1, p_2]$, which can also be seen in [8], where the first value p_1 is the probability of measuring the qubit and observing the state $|0\rangle$ and the second value p_2 is the probability of observing $|1\rangle$. Therefore, we assign the output state $|1\rangle$, with a corresponding distribution $[0, 1]$, the case where the anaphora is connected to the right referent. Similarly, the state $|0\rangle$, with distribution $[1, 0]$, is the case where the anaphora is connected to the wrong referent. For the open wire circuits there are two outgoing wires, so we have a two qubit output. We adopt the same labelling approach as in the single qubit case, except we ignore the two central values. Hence, the labels for open wire circuits are $[0, 0, 0, 1]$ for right anaphora connection and $[0, 0, 0, 1]$ for the wrong anaphora connection. Moreover, for the case where the connection between the pronoun and referent is not made, we simply have the same diagram for both right and wrong labellings.

4.2 Optimisation

The resulting circuits do not use a particularly large number of qubits, on average 10 qubits with a max of 28. The depth is also reasonable, with an average of 24 layers and a max of 45. This is reasonable for NISQ era quantum computers that can handle several hundred qubits and depths of up to 100. However, the main issue is that we have just under $6 * 16400$ circuits, which would need to be run several times. Due to the impracticality of this, we use the NumpyModel from Lambeq. This model classically simulates quantum computers using unitaries and densitry matrices by treating the circuit as a tensor network and performing tensor contractions with the `opt_einsum` function from NumPy [4]. We use the binary cross entropy (BCE) as the loss function, computed at each epoch for training and validation sets. The BCE for a batch of size N is defined below [4].

$$H(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} (y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) \tag{4}$$

Where y is a set of N true labels in $\{0, 1\}$ and similarly $\hat{y}$ are a set of N predicted labels in $(0, 1)$, representing the state of the measured output qubit for circuit i. We also compute the accuracy by comparing the true labels with the predicted labels after rounding. As for the optimiser, we use the Simultaneous Perturbation Stochastic Approximation (SPSA) [30]. This method allows for an efficient evaluation of the loss function gradient using perturbations for approximation, performed simultaneously, resulting in fast parameter updates. Given the circuit parameters θ_k, at iteration k, the output of some circuit $U_i(\cdot)$ is the corresponding predicted label $\hat{y}_i = U_i(\theta_k)$. Hence, the estimated gradient at step k is defined as [5].

$$\hat{\nabla} H(y, U(\theta_k)) = \frac{H(y, U(\theta_k + c_k \Delta_k)) - H(y, U(\theta_k - c_k \Delta_k))}{2 c_k \Delta_k} \tag{5}$$

Where Δ_k is the random perturbation vector with entries ± 1 and c_k is the corresponding perturbation constant by which it is scaled. Given this gradient estimation, the corresponding update rule is [6].

$$\theta_{k+1} = \theta_k - a_k \hat{\nabla}_k H(y, U(\theta_k)) \tag{6}$$

Where a_k is the learning rate. The learning rate and perturbation constants at iteration k are defined as [7].

$$a_k = \frac{a}{(k+A)^\alpha} \qquad c_k = \frac{c}{k^\gamma} \tag{7}$$

Where $\gamma = 0.101$ and $\alpha = 0.602$, constants set by Lambeq [4], although other implementations use different values. Therefore, the tunable hyperparameters are a, A, and c defining the initial state of the learning rate and perturbation constant. We chose the hyperparameters suggested by Lambeq: $a = 0.1$, $A = 0.01 \cdot k^*$, and $c = 0.06$ where k^* is the total number of epochs to be run. We also tried hyperparameter optimisation, which resulted in similar hyperparameters to the suggested ones and did not provide a significant improvement. Although, we did not test the hyperparameter optimisation for all circuits, as it was computationally infeasible with the given resources.

Lastly, for the training data we use a batch size of 32 and train the models for 100 epochs. This is largely motivated by constraints of computational resources and time, with these parameters the fastest runtime still took about a day and a half.

4.3 Pipeline overview

Let us review the whole process so far, describe succinctly in the diagram below [12]. We start with a sentence pair and their corresponding referent-pronoun labelling. The first part of the pipeline is *preprocessing*, denoted by the red dashed box. Steps 1 and 2 are handled by the BobCat parser [4], which runs a BERT model in the backend [31], converting a sentence into a DisCoCat diagram using the CCG parse [17] [32]. Step 3 is done using DiscoPro [29], manually composing the two sentences and generating the appropriate binary label, based on whether the pronoun is connected to the right or wrong referent. Step 4 and 5 simplify the diagram and convert them into circuits based on the rewriter and ansatze, respectively. This preprocessing is performed once for each sentence for each of our 6 different diagram composition and ansatze choices.

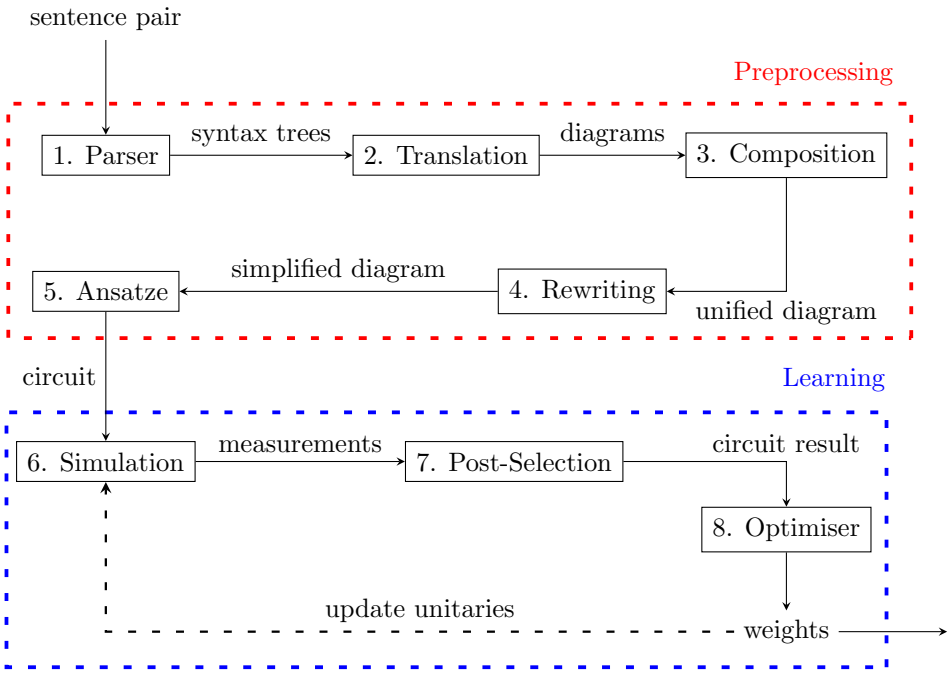

Figure 12: Pipeline for finding solution to anaphora resolution using QNLP. The first 5 steps are for *preprocessing*, where sentences are converted to circuits. The latter 3 steps are for *learning* the free parameters of the circuits.

The latter part, denoted by the dashed blue box, is where we learn the parameters θ of our circuits. The parameters learning is guided with the goal of finding an approximate solution to the anaphora resolution task, which ensures the tensors will be semantically valid [33].

The learning sections starts with step 6, which evaluates the actual circuit, in this case we perform classical simulation of a quantum computer [1]. The next step, 7, post-selects and normalises the outputs provided by the measurement results of the circuit. Then step 8 consists of computing the new weights with the selected optimiser, SPSA here [30], and updating the parametrised unitaries accordingly. Steps 4 to 8 are repeated for a specified number of epochs or until a convergence criterion is met. Note that for the first iteration of the 'learning' process the weights are initialised randomly. Moreover, the loss, and other training metrics, would be computed between step 7 and 8. This second part needs to be done for each set of 6 circuits generated in the first part.

This pipeline was performed on UCL provided remote compute. These clusters consist of GeForce RTX 3090 Ti with 32 Gb of memory. The GPUs ran on CUDA 12.4 with the JAX library used for efficient computation. The just in time compilation method was not used due to the large number of circuits requiring too much memory to precompile. The learned parameters of these models are then used on the test circuits to compute measures of entanglement and contextuality, with the motivation that they have been theoretically proven to be correlated.

5 Quantum Resources

Once the pipeline is completed and we have obtained all the circuits with their corresponding optimised parameters we can use them to measure desired aspects of the output data. The output wires of each sentence should have some semantic representation of them, in the form of two single qubits. This forms a bipartite system and allow us to measure properties of quantum systems. However, we have two sets of circuits that differ in their output. The open wire variety is simple, we already have a bipartite system as an output, so there is nothing more to be done but measure. This can be seen in the left circuit of figure [13].

On the other hand, the case where we compose sentence outputs with a spider requires some post-processing. The spider operation is translated into a CNOT between the output wires of each sentence, with a post selection performed on one of these wires. Therefore, we simply drop the post selection when performing the following measurements, visible in the right circuit of the figure [13].

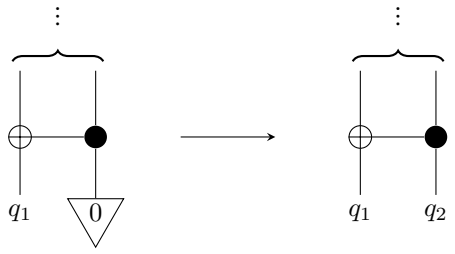

Figure 13: Output wires of the spider based diagrams after the post-selection on the second wire is removed. In the end all circuits used for measurement will have an output structure as seen on the right, although some without the CNOT gate.

5.1 Density Matrices

Measures of entanglement, and much of quantum computation and information in general, works in what is called the density matrix formalism. This is simply an alternate, more general way, of representing qubits and quantum systems. It is based on density matrices defined as [8].

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| \quad \text{such that } p_i > 0 \text{ and } \sum_i p_i = 1 \tag{8}$$

Where $|\psi_i\rangle$ are arbitrary states. Therefore, this defines the possibility for a mixture of states, a so called density matrix. The density matrix will always be positive semi-definite and have a trace of one $\mathrm{Tr}(\rho) = 1$. In this work, we will always be working with pure states, which simply means our density matrix is fully defined by a single state $\rho = |\psi\rangle\langle\psi|$. This is done as it allows us to use many of the known measures of entanglement, practically all it means is that we have to take the outerproduct of the output state from our circuits.

However, the caveat is that the NumpyModel used for classical simulation of the quantum circuits only outputs the real valued probability distributions of the output state. Therefore, we create a wrapper class that inherits the NumpyModel, and create an alternate method for evaluating the circuits that outputs the actual state. This method is very similar to the original and simply skips the conversion from state to distribution.

5.2 Entanglement

In this work we use two measures for entanglement, both are essential criteria of separability as an entangled state is one that cannot be expressed as a tensor product of smaller states. The first measure of entanglement used is the standard entropy of entanglement E [34], based on the Von Neumann entropy with respect to either of the reduced states of a bipartite system ρ_{AB}, [9].

$$E(\rho_A) = -\mathrm{Tr}(\rho_A \log \rho_A) \tag{9}$$

Where ρ_A is the reduced density matrix of subsystem A with respect to subsystem B given by $\mathrm{Tr}_B(\rho_{AB})$, and similarly for ρ_B (where $E(\rho_A) = E(\rho_B)$). Furthermore, it can be computed efficiently via Schmidt decomposition or Eigendecomposition. This is done to avoid the cost and potential errors of computing the matrix logarithm.

The second measure of entanglement used is logarithmic negativity [35], which is an upper limit on the *distillable* entanglement. More importantly for us though, it is a measure that does not reduce to the standard entropy of entanglement just defined. Given a bipartide system ρ_{AB} the logarithmic negativity N is given by the below equation [10].

$$N(\rho_{AB}) = \log_2 \|\rho_{AB}^{\Gamma_A}\| \tag{10}$$

Where ρ^{Γ_A} is the partial transpose with respect to subsystem A and $\|\cdot\|$ is the Schatten norm with $p = 1$ defined as $\|A\| = \mathrm{Tr}(\sqrt{A^* A})$. As before, this can be computed efficiently via Eigendecomposition. Both of these measures E and N are calculated based on the pure density matrix resulting from the two output wires of each sentence in the pair, treating each a subsystem of the individual sentences.

5.3 Measurement Scenario

In order to measure contextuality, we additionally apply a measurement basis to the open wires of the circuits. We choose a set observables that generate a maximal Bell violation, which are $0°, 22.5°, 45°, 67.5°$ [36]; that are not unique. Using these angles we define the four observables we will use with the Y rotation gate [11].

$$R_y(\theta) = \exp\left(-i\frac{\theta}{2}Y\right) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{11}$$

These four observables allow us to create a cyclic Bell type scenario necessary for measurements of contextuality. For some arbitrary circuit with two open wires we generate the following set of measurement contexts [14].

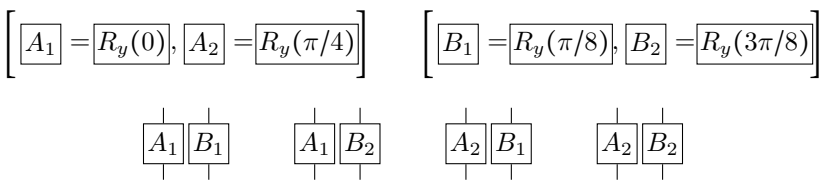

Figure 14: The observables appended to the end of the two output wires of a circuit, resulting in four contexts in different measurement bases.

With the 4 circuits generated via this modification an *empirical table* can be generated upon measuring the outcomes of each context [1].

	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
A_1, B_1	C^1_{00}	C^1_{01}	C^1_{10}	C^1_{11}
A_1, B_2	C^2_{00}	C^2_{01}	C^2_{10}	C^2_{11}
A_2, A_1	C^3_{00}	C^3_{01}	C^3_{10}	C^3_{11}
A_2, B_2	C^4_{00}	C^4_{01}	C^4_{10}	C^4_{11}

Table 1: An empirical table where each row is a context and an entry in a row is the probability of that specific outcome for the bipartide system.

5.4 Incidence Matrix

The contextuality of a measurement scenario can be determined by the existence of a global section, which implies the system is non-contextual. Determining the existence of a global section can be computed using linear algebra via the incidence matrix M of a measurement cover $\mathcal{M}$ and the event sheaf $\mathcal{E}$. First we consider the disjoint union $\bigsqcup_{C \in \mathcal{M}} \mathcal{E}(C)$ of all sections over contexts in $\mathcal{M}$ with an enumeration $s_1, \ldots, s_p$. Sections are enumerated for each of their outcomes, or rather an enumeration of the empirical table. Similarly, we also specify an enumeration $t_1, \ldots, t_q$ of all global assignments $t_j \in O^X$ representing the global sections of $\mathcal{E}$. We then define the incidence matrix M_{pq} as in [37] seen below [12].

$$M[i,j] = \begin{cases} 1, & t_j \mid C = s_i \quad (s_i \in \mathcal{E}(C)) \\ 0, & \text{else} \end{cases} \tag{12}$$

This matrix represents the tuple of restriction maps [13].

$$\mathcal{E}(C) \to \prod_{c \in \mathcal{M}} \mathcal{E}(C) :: s \mapsto (s \mid C)_{C \in \mathcal{M}} \tag{13}$$

Let us derive the specific case for the generalised Bell type scenario (n,k,l) with n parties, each with k measurements to choose from, and each measurement having l possible outcomes. Hence, there are k^n contexts, each having l^n possible outcomes. There are $(kl)^n$ sections over contexts with kn total measurements and l^{kn} global assignments. The resulting incidence matrix will be of size $(kl)^n \times l^{kn}$.

Working from this definition, global sections of the distribution presheaf can be seen as a solution to a linear system of equations involving the incidence matrix. Consider an empirical model $\{e_C\}$ with respect to a distribution $\mathcal{D}_R$, assigning a probability to each section $s_i \in \mathcal{E}(C)$. Instead of the previously encountered table, this can also be treated as a vector v of length p, where $v[i] = e_C(i)$, a sort of flattened empirical table as sections correspond to rows of the empirical table. Moreover, we define another vector x of length q, containing an unknown variable for each global section $t_j \in \mathcal{E}(X)$. Hence, a solution to the linear system $Mx = v$ will be a vector r of weights for each t_j. To ensure that this solution vector r is a probability distribution the matrix M is extended with a row of 1s and the vector v is extended with an element of value 1. This will enforce the following constraint [14].

$$\sum_{i \in q} x_i = 1 \tag{14}$$

This leads to the key result of Sheaf-Theoretic Contextuality used in this thesis. Given the augmented incidence matrix M' and the augmented empirical model vector v' over a distribution functor $\mathcal{D}_R$, the solution to $M'x = v'$ in R is a global section, the existence of which determines the non-contextuality of a scenario. In our specific case we are working in the $(2,2,2)$ scenario, so the incidence matrix will have 256 entries.

5.5 Contextual Fraction

The incidence matrix computation is extended to the contextual fraction (CF) [38] which is defined as the minimal λ that such that $e = (1 - \lambda)e^C + \lambda e^{NC}$, where e is the original empirical model, e^C is the contextual partition, and e^{NC} is the non-contextual partition. This can be solved as an optimisation problem of a set of simultaneous equations, again relying solely on linear algebra of reasonably sized matrices. Hence, the CF is a measure of contextuality and in general a system is considered contextual if the CF is non-zero.

5.6 CbD Measure

Another measure of contextuality comes from the Contextuality-by-Default (CbD) framework [39]. In this framework we say that our measurement scenario lives in the class of cyclic dichotomous systems. This means that our system is consistently connected and we can use the simple inequality [15], to determine the contextuality of a system in the CbD framework.

$$n - D(\mathcal{C}_n) \geq 2 \tag{15}$$

If this inequality does not hold then the system is contextual. All there is to do is compute $D(\mathcal{C}_n)$, which can also be done via optimisation of a linear system of equations. However, it is important to note that this only works for cyclical systems so the ordering of the measurement scenario used is a reordered form of the one in [1], such that the rows are (A_1, B_1), (B_1, A_2), (A_2, B_2), and (B_2, A_1). As we are using the NumpyModel and testing on a bipartite system there is no signalling in these experiments making the computations for the measures of contextuality simplified. Moreover, this also means that its impossible to get maximal contextuality.

6 Results

6.1 Convergence Accuracy during Training

First we look at the performance metrics gathered during the optimisation of the four different models. Below [15] are the per epoch results obtained when optimising the four different kinds of circuits resulting from different diagram structures based on sentence composition.

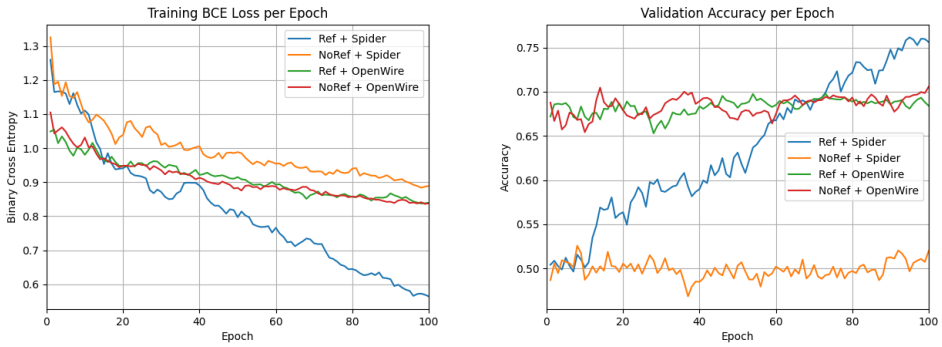

Figure 15: Training data for the four kinds of circuit structures. On the left is the training BCE computed at each epoch. On the right is the accuracy computed on the validation set at each epoch.

Here, if we look at the BCE loss per epoch on the left, we see that the standard model with both sentence and pronoun-referent composition present achieves the lowest loss at the end of the 100 epochs. Moreover, whereas the other three models begin to plateau and stop 'learning', the blue line continues with the same trajectory and would likely continue to decrease with more epochs. Now, focusing on the validation accuracy per epoch, we see that the other three models seem to meander around their starting accuracy set by the randomly initialised parameters. On the other hand, the standard method steadily increases

in accuracy after each epoch. Again we see that the standard model would likely benefit from more epochs. It is unlikely the others would continue to learn, as is suggested by their stagnating loss function.

Moving on, we look at the same results but for different ansatze with the diagram structure fixed, using a spider operation and the pronoun-referent connection present [16]. Out of the three types of ansatze we see that IQP achieves the lowest loss and highest accuracy. Sim14 and Sim15 begin decreasing less sharply earlier. This is interesting in the case of Sim14 as it is more expressive.

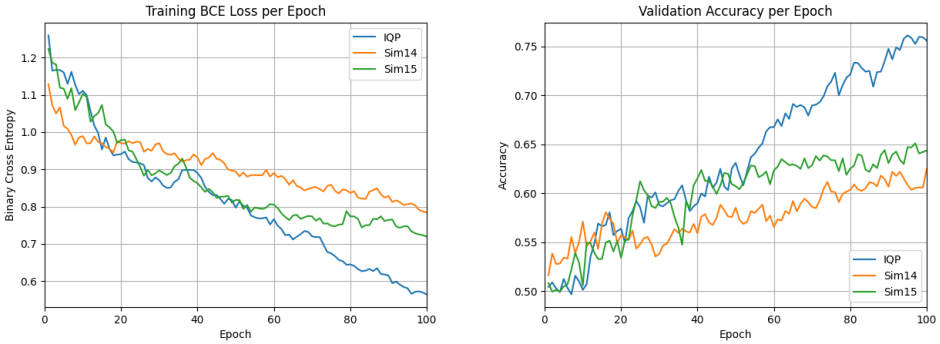

Figure 16: Training data for the four kinds of circuit structures. On the left is the training BCE computed at each epoch. On the right is the accuracy computed on the validation set at each epoch.

These observations are corroborated by the summary of results [2], including the final test accuracy and loss. In the table we denote the four different types of sentence composition as a combination of abbreviations. Either present pronoun-referent connection (r) or not present (nr) and sentences composed with spider (s) or left open (ow). The results in the table further demonstrate the best performance of standard IQP (rs). It also had the fastest runtime and fewest parameters, tied with Sim15 coincidentally. Despite their poor learning the (row) and (nrow) variants of IQP performed quite well, this is likely due to luck with the randomised parameter initialised at the start. As we saw in the graph above [15], the accuracy did not change at all during training and simply fluctuated.

Model	Val. Acc.	Val. Loss	Test Acc.	Test Loss	$\#\theta$	s/ep
IQP (rs)	**0.75613**	**0.52200**	**0.74646**	**0.53297**	**6572**	**1322**
IQP (nrs)	0.52005	0.87348	0.50122	0.87608	6903	1639
IQP (row)	0.68401	0.86517	0.68997	0.86775	6591	1402
IQP (nrow)	0.70572	0.81747	0.70020	0.84626	6920	1582
Sim14 (rs)	0.62500	0.80361	0.63986	0.76044	10104	1569
Sim15 (rs)	0.64373	0.71104	0.64832	0.71228	**6572**	1401

Table 2: Results of the respective models. Validation accuracy and loss are taken from the last epoch. $\#\theta$ is the number of parameters learnt and the s/ep is the average time in seconds per epoch.

The stagnation present in the three models can potentially be explained by barren plateau [40]. This is a phenomenon in quantum machine learning, where due to the exponentially increasing size of the parameter space, the resulting loss landscape is barren. This means that the gradient of the loss function at any give point is, with high probability, near zero. This results in a 'no learning' behaviour, where the optimiser explores the landscape aimlessly at random. This can be caused by several things: larger number of qubits, very deep circuits, random initialisation of PQCs, noisy gradients (SPSA), or a large presence of globally entangling gates. Many of these properties are quite typically induced by ansatze.

6.2 Contextuality and Entanglement Results

In the below graph [17], we have the density and number of contextual cases generated by each of the diagram types for IQP. These were further separated into whether the pronoun was connected to the right or wrong referent for the two models that do have a connection present. Overall, we can see that there are very few contextual cases for both CF and CbD. This is fairly standard as it is a rare phenomenon. Importantly, we note that very intuitively the additional entangling gates generated by the spider operation and connection of the referent to the pronoun result in a higher frequency of contextuality. The standard IQP model generated the most contextual cases, whereas either model that had no pronoun-referent connection had essentially no cases of contextuality.

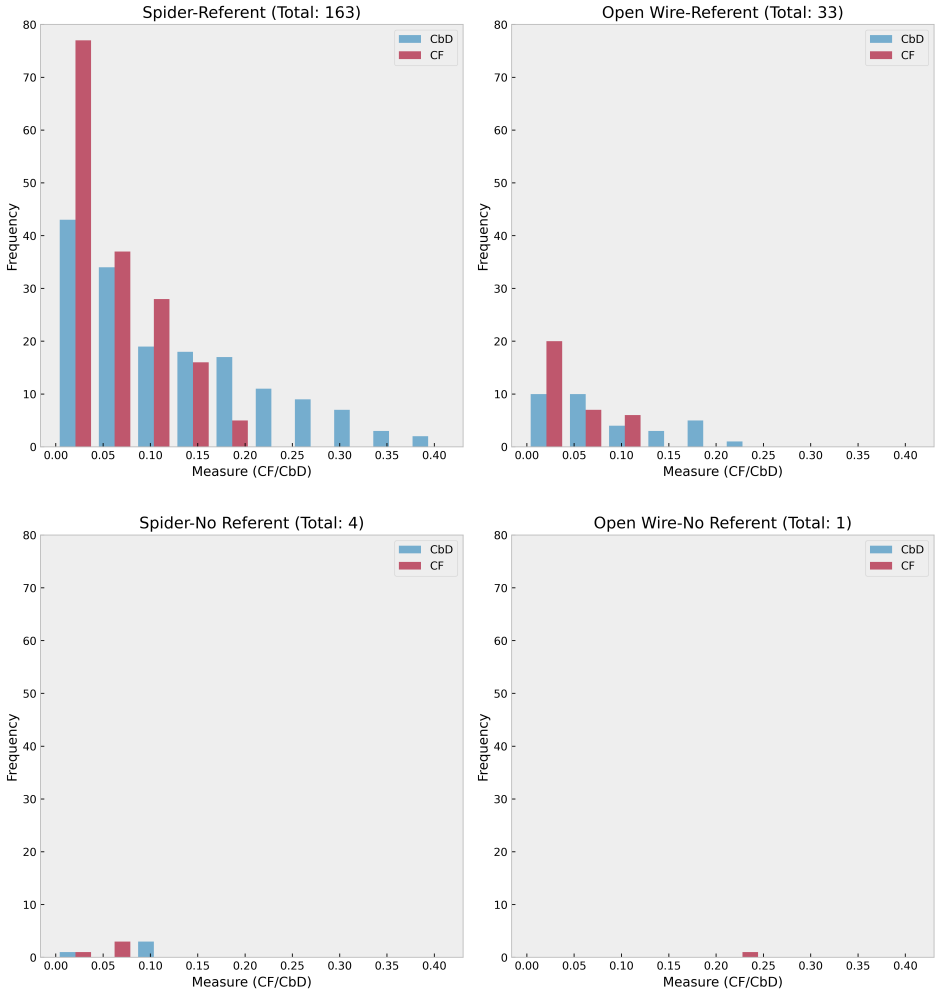

Figure 17: Number of contextual cases in the data. The x-axis is the CF and CbD, blue and red respectively. The y-axis is the density.

Moving on to the entanglement measures [18], where we see similarities to the contextuality graphs. With respect to the entropy of entanglement, for all models the majority of circuits lie near no entanglement present 0 or fully entangled 1, with a smaller peak in between. As before, the cases with no referent connected are generally *less* entangled, concentrating near 0 and moving further away from maximal entanglement. These patterns are also visible with the logarithmic negativity. The main difference is that there is a larger discrepancy between referent connected and unconnected models compared to the entropy of entanglement. Furthermore, the previously seen central peak is gone.

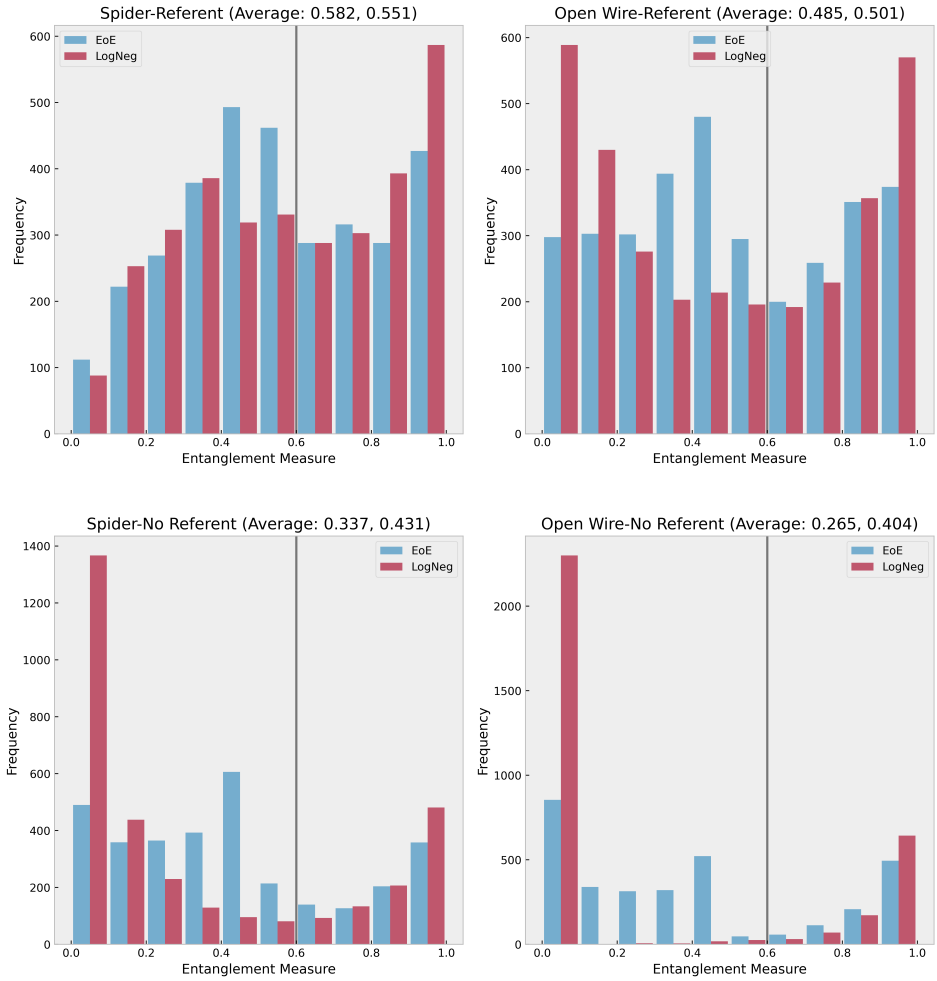

Figure 18: The measures of entanglement on the x-axis and frequency on the y-axis. Blue corresponds to entropy of entanglement and red corresponds to logarithmic negativity.

Based on all this collected data we also trained a standard Support Vector Machine (SVM) to predict the presence of a pronoun-referent connection or right-wrong referent. In total, we trained 4 SVMs two for each problem, one based on the entropy of entanglement and the other based on logarithmic negativity. The prediction accuracy is displayed in the table below [3]. Note that we only did this for the spider model as there is not much difference in results when comparing to the open wire model.

	Entropy	LogNeg
No Referent	55.665%	74.806%
R/W Referent	55.082%	57.049%

Table 3: Prediction results for the presence or not of a pronoun-referent connection and wrong or wrong referent based on entropy and logarithmic negativity.

Lastly, we compared how the circuits of different Ansatze algorithms affect these measures on the standard spider model with referent connected. We considered the standard IQP Ansatze along with Sim14 and Sim15 models. We compare them with regard to both entanglement measures and also the contextuality measures [19]. Here we see that, overall the Sim14 model produces the highest number of contextual cases. This is followed by the Sim15 model, and lastly by the standard IQP model. Another note is that for the Sim14 and Sim15 models the overlap between CbD and CF contextual sentences is full.

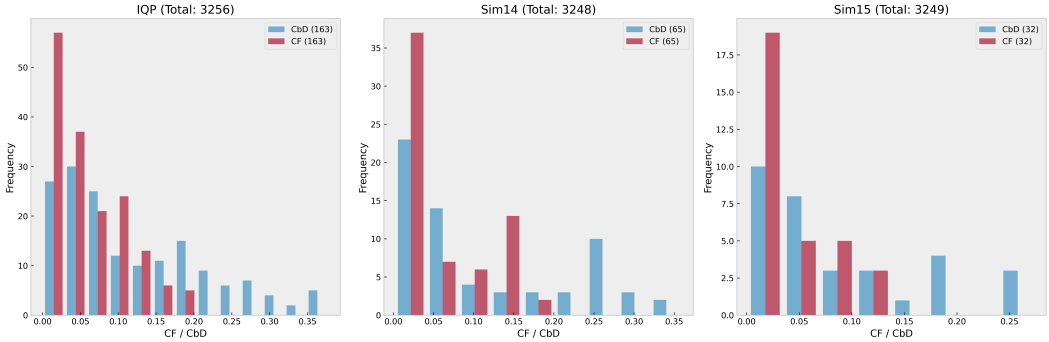

Figure 19: Graphs plotting frequency against CbD with CF of only the contextual (non-zero) cases. The total number of contextual cases is shown in the legend at the top right.

As the Sim14 and Sim15 should be much more entangling, but interestingly have less contextual cases, we look at the correlation between entanglement and contextuality by calculate the Spearman R coefficient between the entropy of entanglement and fraction of contextuality in the data with respect to each Ansatze model. The R coefficients are 0.0373 for IQP, 0.0866 for Sim14, and 0.0733 for Sim15; all with a statistical significance orders of magnitudes below 0.5. This proves that there is a positive correlation, as we observed, but it is a very weak one. This is likely due to the fact that the contextuality data is largely 0. To further explore, this correlation we looked at the average entanglement of the contextual cases. For entropy of entanglement this is 0.580 for IQP, 0.665 for Sim14, and 0.686 for Sim15. Similarly, for the logarithmic negativity we have: 0.631 for IQP, 0.763 for Sim14, and 0.759 for Sim15. We see that overall the contextual cases are definetly not highly entangled but are also not weakly entangled and the relationship between the two is indeed not so linear.

7 Discussion

The small number of contextual cases across all models is likely due to the rarity of contextuality, which is still a largely not well understood resource. There are also two other possibilities. The schema used, and the general pipeline, is likely not optimised for maximising contextuality. This work relied on contextual cases occurring naturally, essentially randomly, which would be unlikely unless there was a unique property of language that would be preserved when mapped to circuits. The first possibility is that there could be certain linguistic structures that would produce much more contextuality. Another possibility is forcing scenarios that would result in high or even maximal contextuality in such a way that the measurement method can be generalised outside the strict structure. The second option is that the mapping from string diagrams to circuits does not produce very expressive circuits. With respect to the different ansatze models, the contextuality for 2 qubits is bounded by $\sqrt{2} - 1 \simeq 0.414$ and the minimum entanglement for contextuality is 0.601 according to [14]. Our results showed that these bounds were respected. In particular, 2 qubit contextuality is certainly bounded and hence so is the entanglement as one is a positive monotone of the other. This would explain why none of the contextual cases found were highly entangled.

The lower number of high entanglement circuits in the model with no connection between the pronoun and the referent is likely due to the fact that the lack of connection leads to fewer controlled gates, which are necessary for generating entangled states. Hence, it is also fairly reasonable that there is a large peak of circuits near 0 entanglement. This leads to question of more 'useful' or 'meaningful' forms of entanglement. For example, the logarithmic negativity quantifies a certain type of entanglement said to be 'distillable' or the just mentioned work by [14] stratifies entanglement further into the class that also generates contextuality. This is a popular and open field of research trying to understand entanglement better and how to use it by identifying certain classes of it. It is important to note that the in the case of logarithmic negativity the results are much more extreme, with all data points being pushed to either note at all or fully entangled. As just mentioned this is likely due to the fact that it

specifically measures *distillable* entanglement which is a stricter class of states. This is also seen in the fact that the model with no referent generally has a much higher concentration of states nearer to 0 compared to other models and in the case of entropy of entanglement. Furthermore, from the results based on the SVM it seems that logarithmic negativity seems to be a better predictor than entropy of entanglement in this context.

Interestingly, we noticed that splitting our data based on semantically desirable properties such as whether the referent is right or wrong; a subject or object, did not yield significant results. The patterns stay the same as before the split, meaning there is likely no importance tied to whether the pronoun refers any of these specific words. At the very least, in the scope of this experimental setting. However, another interesting result is that the only set of circuits not to suffer stagnation when learning the unitary parameters was the standard structure where the pronoun is connected to the right referent and the sentences are composed via a spider. This suggests that the semantically meaningful structure of the circuits resulted in a good optimisation.

8 Summary and Future Work

First and foremost we developed and deployed a robust pipeline for measuring and testing quantum phenomena between two connected sentences. This consisted of combining sentence parsing, different types of composition, diagram rewriting, and multiple ansatze conversions to translate the input sentence into meaningful PQCs. This was achieved by combining various existing methods from QNLP and quantum resource theory.

By converting sentences into individual diagrams and composing them based on linguistic properties, we can then convert the overall diagram into a quantum circuit with an output state representing the joint system of the learnt shared semantic space of the sentences. As this is simply an arbitrary quantum system, any of several measures of quantum phenomena can be used to investigate and explore specific information hidden in the overall system. In our case, we successfully explored entanglement and contextuality present in the output states of 16400 sentences for 6 different models differing in diagrammatic structure and ansatze choice. Intuitively, this displayed a positive correlation between the presence of semantically meaningful composition or the lack thereof. Trivially, in this context the these metrics simply act as quantifiers of the degree of 'connectivity' between the sentences. These results also corroborate recent work connecting sheaf theoretic contextuality and entanglement, experimentally confirming that the contextual fraction behaves as a positive monotone for both entropy of entanglement and logarithmic negativity, in this scenario [14].

Importantly, unlike other previous works that heavily restrict the sentence structure, this approach works for any sequence of sentences which can then be treated as individual subsystem of a larger quantum system. Moreover, the flexibility of the method allows for easy extendability and further improvements.

This work has opened many interesting and fruitful avenues of research to be pursued. In particular, contextuality is a field that has recently been developing swiftly and new measures have been introduces. For instance, some of these do not rely on strict specification of the measurement scenario but rely solely on the collection of the output states obtained in different contexts [41] based on combinations of preparations and measurements.

Moreover, as discussed in this paper the ansatze used in many VQAs are often characterized by their entangling power due to its importance in efficient quantum computation. Recent work has shown the importance of contextuality in efficient quantum algorithms [12], hence it follows naturally to consider the contextual power of ansatze. Similarly, it could be interesting to characterise specific forms of entanglement that are correlated with expressive ansatze and moreover, good optimization. Ideally, these investigations would be interesting in a more linguistic context further exploring the connection between semantically meaningful representations of sentences in the form of quantum circuits and the presence of quantum phenomena.

References

[1] R. Lorenz, A. Pearson, K. Meichanetzidis, D. Kartsaklis, and B. Coecke, "Qnlp in practice: Running compositional models of meaning on a quantum computer," *Journal of Artificial Intelligence Research*, vol. 76, pp. 1305–1342, 2023, ISSN: 1076-9757. DOI: 10.1613/jair.1.14329. [Online]. Available: http://dx.doi.org/10.1613/jair.1.14329.

[2] K. Meichanetzidis, A. Toumi, G. de Felice, and B. Coecke, "Grammar-aware sentence classification on quantum computers," *Quantum Machine Intelligence*, vol. 5, no. 1, p. 10, 2023.

[3] H. Wazni and M. Sadrzadeh, *Towards transparency in coreference resolution: A quantum-inspired approach*, 2023. arXiv: 2312.00688 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2312.00688.

[4] D. Kartsaklis, I. Fan, R. Yeung, *et al.*, "Lambeq: An Efficient High-Level Python Library for Quantum NLP," *arXiv preprint arXiv:2110.04236*, 2021.

[5] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997, ISSN: 1095-7111. DOI: 10.1137/s0097539795293172. [Online]. Available: http://dx.doi.org/10.1137/S0097539795293172.

[6] L. K. Grover, *A fast quantum mechanical algorithm for database search*, 1996. arXiv: quant-ph/9605043 [quant-ph]. [Online]. Available: https://arxiv.org/abs/quant-ph/9605043.

[7] S. Chakraborty, S. Banerjee, S. Adhikari, and A. Kumar, *Entanglement in the grover's search algorithm*, 2013. arXiv: 1305.4454 [quant-ph]. [Online]. Available: https://arxiv.org/abs/1305.4454.

[8] V. M. Kendon and W. J. Munro, *Entanglement and its role in shor's algorithm*, 2006. arXiv: quant-ph/0412140 [quant-ph]. [Online]. Available: https://arxiv.org/abs/quant-ph/0412140.

[9] G. Gour, *Resources of the quantum world*, 2024. arXiv: 2402.05474 [quant-ph]. [Online]. Available: https://arxiv.org/abs/2402.05474.

[10] J. S. Bell, "Introduction to the hidden-variable question," CM-P00058691, Tech. Rep., 1971.

[11] M. Howard, J. Wallman, V. Veitch, and J. Emerson, "Contextuality supplies the 'magic' for quantum computation," *Nature*, vol. 510, no. 7505, pp. 351–355, 2014, ISSN: 1476-4687. DOI: 10.1038/nature13460. [Online]. Available: http://dx.doi.org/10.1038/nature13460.

[12] F. Shahandeh, *Quantum computational advantage implies contextuality*, 2021. arXiv: 2112.00024 [quant-ph]. [Online]. Available: https://arxiv.org/abs/2112.00024.

[13] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, "Quantum entanglement," *Rev. Mod. Phys.*, vol. 81, pp. 865–942, 2 2009. DOI: 10.1103/RevModPhys.81.865. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.81.865.

[14] T. Chan and A. Constantin, *The contextual fraction as a measure of entanglement*, 2024. arXiv: 2403.06896 [quant-ph]. [Online]. Available: https://arxiv.org/abs/2403.06896.

[15] D. Wang, M. Sadrzadeh, S. Abramsky, and V. H. Cervantes, *On the quantum-like contextuality of ambiguous phrases*, 2021. arXiv: 2107.14589 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2107.14589.

[16] K. I. Lo, M. Sadrzadeh, and S. Mansfield, *Quantum-like contextuality in large language models*, 2024. arXiv: 2412.16806 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2412.16806.

[17] B. Coecke, M. Sadrzadeh, and S. Clark, *Mathematical foundations for a compositional distributional model of meaning*, 2010. arXiv: 1003.4394 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1003.4394.

[18] A. Lenci and M. Sahlgren, *Distributional Semantics* (Studies in Natural Language Processing). Cambridge University Press, 2023.

[19] J. Lambek, "The mathematics of sentence structure," *The American Mathematical Monthly*, vol. 65, no. 3, pp. 154–170, 1958. DOI: 10.1080/00029890.1958.11989160. [Online]. Available: https://doi.org/10.1080/00029890.1958.11989160.

[20] J. Lambek, "Type grammar revisited," in *Logical Aspects of Computational Linguistics*, A. Lecomte, F. Lamarche, and G. Perrier, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 1–27.

[21] M. Steedman, "Combinatory grammars and parasitic gaps," *Natural Language & Linguistic Theory*, vol. 5, no. 3, pp. 403–439, 1987. DOI: 10.1007/BF00134555.

[22] M. Steedman, *The Syntactic Process*. MIT Press, 2000.

[23] A. E. Ades and M. J. Steedman, "On the order of words," *Linguistics and Philosophy*, vol. 4, pp. 517–558, 1982.

[24] M. Steedman and J. Baldridge, "Combinatory categorial grammar," in *Non-Transformational Syntax*. John Wiley & Sons, Ltd, 2011, ch. 5, pp. 181–224, ISBN: 9781444395037. DOI: https://doi.org/10.1002/9781444395037.ch5. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781444395037.ch5. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781444395037.ch5.

[25] J. Maillard, S. Clark, and E. Grefenstette, "A type-driven tensor-based semantics for CCG," in *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, 2014, pp. 46–54.

[26] G. Wijnholds, M. Sadrzadeh, and S. Clark, "Representation learning for type-driven composition," in *Proceedings of the 24th Conference on Computational Natural Language Learning*, R. Fernández and T. Linzen, Eds., 2020, pp. 313–324.

[27] D. Shepherd and M. J. Bremner, "Temporally unstructured quantum computation," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 465, no. 2105, 2009, ISSN: 1471-2946. DOI: 10.1098/rspa.2008.0443. [Online]. Available: http://dx.doi.org/10.1098/rspa.2008.0443.

[28] S. Sim, J. Romero, J. F. Gonthier, and A. A. Kunitsa, "Adaptive pruning-based optimization of parameterized quantum circuits," *Quantum Science and Technology*, vol. 6, no. 2, p. 025 019, 2021. DOI: 10.1088/2058-9565/abe107. [Online]. Available: https://dx.doi.org/10.1088/2058-9565/abe107.

[29] K. I. Lo, *Discopro*, https://github.com/kinianlo/discopro, 2023.

[30] J. Spall, "Implementation of the simultaneous perturbation algorithm for stochastic optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 3, pp. 817–823, 1998. DOI: 10.1109/7.705889.

[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1810.04805.

[32] R. Yeung and D. Kartsaklis, *A ccg-based version of the discocat framework*, 2021. arXiv: 2105.07720 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2105.07720.

[33] K. Lee, L. He, M. Lewis, and L. Zettlemoyer, *End-to-end neural coreference resolution*, 2017. arXiv: 1707.07045 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1707.07045.

[34] F. Giraldi and P. Grigolini, "Quantum entanglement and entropy," *Phys. Rev. A*, vol. 64, p. 032 310, 3 2001. DOI: 10.1103/PhysRevA.64.032310. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.64.032310.

[35] M. B. Plenio, "Logarithmic negativity: A full entanglement monotone that is not convex," *Phys. Rev. Lett.*, vol. 95, p. 090 503, 9 2005. DOI: 10.1103/PhysRevLett.95.090503. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.95.090503.

[36] G. Peruzzo and S. Sorella, "Entanglement and maximal violation of the chsh inequality in a system of two spins j: A novel construction and further observations," *Physics Letters A*, vol. 474, p. 128 847, 2023, ISSN: 0375-9601. DOI: https://doi.org/10.1016/j.physleta.2023.128847. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037596012300227X.

[37] S. Abramsky and A. Brandenburger, "The sheaf-theoretic structure of non-locality and contextuality," *New Journal of Physics*, vol. 13, no. 11, p. 113 036, 2011, ISSN: 1367-2630. DOI: 10.1088/1367-2630/13/11/113036. [Online]. Available: http://dx.doi.org/10.1088/1367-2630/13/11/113036.

[38] S. Abramsky, R. S. Barbosa, and S. Mansfield, "Contextual fraction as a measure of contextuality," *Phys. Rev. Lett.*, vol. 119, p. 050 504, 5 2017. DOI: 10.1103/PhysRevLett.119.050504. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.119.050504.

[39] E. N. Dzhafarov and J. V. Kujala, "Context–content systems of random variables: The contextuality-by-default theory," *Journal of Mathematical Psychology*, vol. 74, pp. 11–33, 2016, Foundations of Probability Theory in Psychology and Beyond, ISSN: 0022-2496. DOI: https://doi.org/10.1016/j.jmp.2016.04.010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022249616300207.

[40] M. Larocca, S. Thanasilp, S. Wang, *et al.*, *A review of barren plateaus in variational quantum computing*, 2024. arXiv: 2405.00781 [quant-ph]. [Online]. Available: https://arxiv.org/abs/2405.00781.

[41] F. Shahandeh, T. Yianni, and M. Doosti, *Characterizing contextuality via rank separation with applications to cloning*, 2024. arXiv: 2406.19382 [quant-ph]. [Online]. Available: https://arxiv.org/abs/2406.19382.