# Towards Transparency in Coreference Resolution: A Quantum-Inspired Approach

**Hadi Wazni**
University College London
hadi.wazni.20@ucl.ac.uk

**Mehrnoosh Sadrzadeh**
University College London
m.sadrzade@ucl.ac.uk

## Abstract

Guided by grammatical structure, words compose to form sentences, and guided by discourse structure, sentences compose to form dialogues and documents. The compositional aspect of sentence and discourse units is often overlooked by machine learning algorithms. A recent initiative called Quantum Natural Language Processing (QNLP) learns word meanings as points in a Hilbert space and acts on them via a translation of grammatical structure into Parametrised Quantum Circuits (PQCs). Previous work extended the QNLP translation to discourse structure using points in a closure of Hilbert spaces. In this paper, we evaluate this translation on a Winograd-style pronoun resolution task. We train a Variational Quantum Classifier (VQC) for binary classification and implement an end-to-end pronoun resolution system. The simulations executed on IBMQ software converged with an F1 score of 87.20%. The model outperformed two out of three classical coreference resolution systems and neared state-of-the-art SpanBERT. A mixed quantum-classical model yet improved these results with an F1 score increase of around 6%.

## 1 Introduction

Large language models (LLMs), such as GPT-3 (Brown et al., 2020), have achieved impressive success in various NLP tasks and have become increasingly common in everyday life through search engines, personal assistants, and other applications. They are trained on vast corpora of text, which are sourced from books, articles, and websites. LLMs learn complex connections between words and phrases by predicting the likelihood of a word appearing in the context of other words. These learned probability distributions capture the statistical patterns of word co-occurrences in data; due to this, LLMs are also known as distributional language models.

Despite their successes in advancing language understanding and generation, LLMs often face criticism for being black boxes (Buhrmester et al., 2019). This means that it is challenging to understand how they make their predictions, which can in turn make them unreliable and difficult to debug. One way to enhance the transparency and interpretability of these models is to explicitly integrate linguistic structure (Lambek, 1958; Chomsky, 1957) into them.

A notable approach attempting this integration is the Distributional Compositional Categorical (DisCoCat) model (Coecke et al., 2010; Kartsaklis and Sadrzadeh, 2013), which pioneered the paradigm of merging explicit grammatical (or syntactic) structure with distributional (or statistical) data for encoding and computing meanings of sentences. DisCoCat offered tools for a compositional statistical modelling of sentence-level linguistic phenomena, such as lexical entailment and ambiguity, by providing transparent meaning assignments for complex syntactic structures, e.g. relative and possessive clauses (Sadrzadeh et al., 2013, 2014), conjunctive and negation operations (Lewis, 2020). Its underlying theory, however, relied on generalisations of vectors to higher order tensors, which made the framework in need of large computational resources and led to limited scalability.

Conversely, tensors are natural components of quantum systems, and quantum computing resources can efficiently learn them. This idea has led to the development of Quantum Natural Language Processing (QNLP). In QNLP, words are represented as points within a Hilbert space, grammatical structures are represented as Parameterised Quantum Circuits (PQCs), and the learning of circuit parameters is achieved through simulations conducted on accessible quantum computing resources, such as IBMQ quantum computers. QNLP has so far been applied to a variety of tasks, e.g. sentence classification (Lorenz et al., 2021), sentence generation (Karamlou et al., 2022), question answering (Meichanetzidis et al., 2023), sentiment analysis (Ruskanda et al., 2022; Stein et al., 2023; Ganguly et al., 2023), musical composition (Miranda et al., 2021), and language translation (Abbaszade et al., 2023). Moreover, the theoretical underpinnings of QNLP have been extended to model discourse structure and have been tested on a limited toy dataset (Wazni et al., 2022).

In this paper, we expand this dataset by introducing a few-shot prompting technique and generate synthetic Winograd-style ambiguous coreference sentences (Levesque et al., 2012) using GPT-3. We apply this method to a set of initial sentences from (Rahman and Ng, 2012) and create a dataset consisting of 16,400 entries. This dataset have a larger number of data points, longer and more complex sentences, and a broader range of grammatical structures when compared to the dataset in (Wazni et al., 2022), where sentences followed a subject-verb-object structure.

We train a Variational Quantum Classifier (VQC) for binary classification and integrate it into an end-to-end pronoun resolution system. Our system's performance surpasses that of classical coreference resolution systems such as CoreNLP (Manning et al., 2014) and Neural Coreference (Clark and Manning, 2016a,b), and it achieves results that are close with the state-of-the-art SpanBERT (Lee et al., 2018), with an F1 score of 87.20%. Following recent practice in quantum machine learning (QML) (Araujo and da Silva, 2020; Macaluso et al., 2020), we merge our quantum system with classical engines to construct a *mixed quantum-classical* pronoun resolver. In alignment with results observed in QML across various domains (Grossi et al., 2022; Batra et al., 2020; Kerenidis and Luongo, 2020), we find that the classical and quantum results are *complementary*, thus our mixed approach yields a significant performance improvement, resulting in an approximate 6% increase in the F1 score.

## 2 Background and Related Work

In the DisCoCat framework, the grammatical structure of a sentence guides the composition of its word-meanings, leading to the derivation of meaning for the sentence as a whole (Coecke et al., 2020, 2013). The grammatical structures are modelled by proofs derived using the rules of Joachim Lambek's logic of syntax, known as the Lambek Calculus (Lambek, 1988). These proofs are interpreted as *processes* and modelled by morphisms of a monoidal category, which comes equipped with a *string diagrammatic* graphical notation (Piedeleu and Zanasi, 2023). Examples of processes that can be effectively modelled by a monoidal category include linear maps over finite-dimensional vector spaces, and this was the initial concept behind the introduction of DisCoCat. Atomic words like noun phrases are represented as points within finite-dimensional vector spaces, while functional words such as adjectives and verbs are depicted as points within the tensor products of these vector spaces. The interconnection of vector and tensor spaces is facilitated through their grammatical dependencies. By contracting these dependencies, the framework allows for the derivation of the overall meaning of the entire sentence.

In fact, the formulation of vectors and tensors into a monoidal category goes back to a framework known as categorical quantum mechanics (CQM), which reformulated quantum theory in terms of process theories and used string diagrams to describe quantum protocols (Abramsky and Coecke, 2008; Coecke and Kissinger, 2017). For a detailed introduction to quantum computing and CQM, see (Nielsen and Chuang, 2010; Coecke and Kissinger, 2017; Sutor, 2019). As a result, monoidal categories and string diagrams became a common base in which one can use analogical reasoning to relate language with quantum theory. For instance, Hilbert spaces, where quantum states are encoded, are vector spaces, so quantum states are related to word-meanings and grammatical reductions correspond to processes such as quantum maps, quantum effects, and measurements.

### 2.1 Lambek Calculus and its modal extensions

The formulae of *Lambek Calculus* (LC) are generated according to the following BNF:

$$A, B ::= A \in At \mid A \cdot B \mid A\backslash B \mid A/B$$

Atomic types $A \in At$ are atomic linguistic types, e.g. noun phrases $n$ and sentences $s$, multiplication $A \cdot B$ is their composition, and the slashes $A\backslash B$ and $A/B$ build complex types, e.g. for words with function types such as adjectives and verbs.

In (Kanovich et al., 2020), an extension of LC with two operations $!A$ and $\nabla A$ was introduced. The new logic was named *Lambek calculus with soft sub-exponentials* (**SLLM**). In (McPheat et al., 2020), the new modal formulae were used to model the linguistic types found in discourse, e.g. pronouns and other ellipsis markers. The !-modal

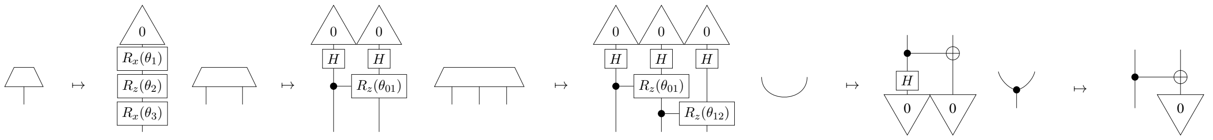

Figure 1: Translation from string diagrams to PQCs using a single-layer IQP ansatz, where each grammatical type is mapped to a 1-qubit space.

types were used for copying referents up to a bound $k$, and the $\nabla$-modal types moved them to the locations of their markers, where they were referred to. The authors showed how the logic could model and reason about definite pronoun discourse ambiguities, such as the Winograd schema examples, and sloppy vs strict readings of elliptic sentences.

In (Coecke et al., 2013), the following vector space semantics was proposed for LC:

$$[\![A]\!] = V \in FdVect, \qquad [\![A \cdot B]\!] = [\![A]\!] \otimes [\![B]\!]$$

$$[\![A \backslash B]\!] = [\![B/A]\!] = [\![A]\!]^* \otimes [\![B]\!]$$

In this semantics, atomic linguistic types are interpreted as finite-dimensional vector spaces and their multiplication as the tensor product of spaces; the slash types are interpreted as the set of all linear maps between their two spaces, via the dual vector space denoted by $(-)^*$. Words are interpreted as elements of the vector spaces associated to their types. This semantics was extended to **SLLM** in (McPheat et al., 2020), by interpreting the copiable linguistic categories as *k-truncated* Fock spaces, defined as follows:

$$[\![!^k A]\!] = \bigoplus_{i=0}^{k} [\![A]\!]^{\otimes i} = k \oplus [\![A]\!] \oplus ([\![A]\!] \otimes [\![A]\!]) \cdots$$

$$\cdots \oplus ([\![A]\!] \otimes [\![A]\!] \otimes [\![A]\!]) \oplus \cdots \oplus [\![A]\!]^{\otimes k}$$

A Fock space closes its base vector $A$ under an infinite number of tensor products, and a $k$-truncated version of it only looks at the first $k$ tensors. Access to any copies of a linguistic category (less than the bound $k$) is facilitated by projecting to that layer. Movable categories take advantage of the commutativity of the tensor product between finite-dimensional vector spaces. The direct sum operation $\oplus$ cannot be directly represented using the quantum gates available in QNLP, which corresponds to the gates provided by IBMQ. We thus translate it into a PQC after projecting it to the desired layer.

A summary of the translation between our Fock space semantics and PQC is provided in Figure 1. Due to space restrictions, we present the translation for the case where only a single qubit is allocated to each atomic linguistic type. In theory, the translation is easily extendible to larger numbers of qubits, but in practice one will face computational limitations. There are two types of diagrams: those on the left, which represent string diagrams associated with vector spaces, and the ones on the right, which depict diagrams used for quantum circuits. On the string diagrammatic side, a parallelogram box with one leg depicts words with an un-copied atomic types. A parallelogram with many legs either depicts a words with a copied type or a functional type. Cupped lines depict the application of a linear map. The concatenation of two atomic sentence types has a conjunctive (rather than tensorial) interpretation, and this is modelled by the Frobenius multiplication between vector spaces. This multiplication is diagrammatically denoted by a bullet symbol (•).

In Figure 2, an example of a string diagram, where *"books"* and *"learning"* are depicted without being copied, which is indicated by their parallelograms having one leg each. *"The students"* is copied and has a parallelogram with two legs. The pronoun *"They"* is shown with one input and one output, giving it two legs. The verbs *"were"* and *"read"* are represented with two inputs and one output, resulting in three legs each. Cupped lines in the diagram illustrate the application of verbs to their subjects and objects, while a bullet symbol (•) is used to connect *"The students read the books"* with *"They were learning"*.

On the circuit side, a triangle labeled with 0 represents a qubit state in the zero computational basis. A box labeled with $H$ signifies a Hadamard gate. A CNOT gate is denoted by a dot connected horizontally to $\oplus$. A controlled-Z-rotation gate with angle $\alpha$, depicted as a box labeled with $R_\alpha(\theta_i)$, is connected horizontally to a control qubit, where $\alpha$ can be $x$, $y$, or $z$, and $\theta$ ranges from 0 to $2\pi$. An upside-down triangle labeled with 0 signifies a measurement in the computational basis, post-selected to be zero.

## 3 Methodology

We build upon the steps in (Lorenz et al., 2021) to represent an entire discourse as a PQC.

**Parsing and Diagram Generation:** The first step involves parsing a discourse into a proof in **SLLM**. We do this via a translation to Combinatory Categorial Grammar (CCG)[1], which enables the use of the state-of-the-art parser (Clark, 2021; Yeung and Kartsaklis, 2021). The parse trees are then transformed to string diagrams through *DisCoPy* (de Felice et al., 2021).

**Diagram Optimisation:** The number of qubits available on contemporary quantum computers is restricted. For instance, IBM's largest superconducting quantum computer, as of now, has a maximum of 433 qubits[2]. Publicly accessible devices typically offer fewer qubits, often less than 10. Consequently, in the second step, the string diagrams are optimised to minimise the number of qubits associated to them after the translation. QNLP diagrams are composed of a layer of tensors, followed by a layer of applications between the tensors. One approach to reduce the number of qubits is elimination of cups through the transformation of states into effects. Another approaches aims for stretching and reordering them. *Lambeq* (Kartsaklis et al., 2021a) supports additional rewriting rules. An example of an optimised diagram is provided in Figure 2.

**Quantum Circuit Transformation:** In the last step, the optimised string diagrams are transformed into quantum circuits. This conversion relies on a parameterisation scheme, known as an *ansatz*. An *ansatz* serves as a mapping that determines the quantity of qubits linked with each wire in the string diagram, along with a distinct variational quantum circuit associated with each word. In this study, we choose the popular *Instantaneous Quantum Polynomial* (IQP) ansatz, developped in (Shepherd and Bremner, 2009; Havlíček et al., 2019). The resulting quantum circuits are ready for execution on either a quantum computer or a simulator. The details of training these circuits can be found in Section 4.3. Figure 3 illustrates the circuit derived from the diagram presented in Figure 2.

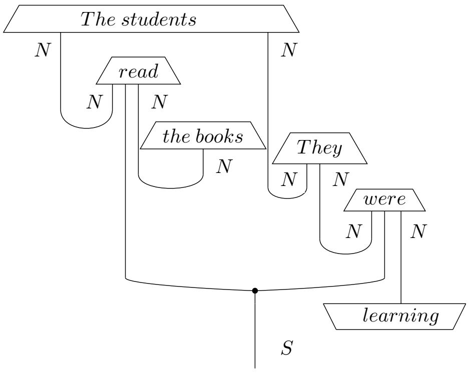

Figure 2: An optimized **SLLM** diagram for a pair of sentences *"The students read the books. They were learning."* To enhance clarity, we treat the determiner-noun phrases *"The students"* and *"The books"* as single units, as determiners are eventually discarded in the rewriting process.

## 4 Classification Task

Pronoun resolution is a computational linguistic process that involves identifying the antecedent of a pronoun within a text. In our experiment, we consider pronoun resolution as a supervised binary classification task. Given a sentence containing a pronoun, the goal is to determine whether a potential antecedent (such as a noun or noun phrase) in the preceding sentence is the correct referent for the pronoun or not. This task requires training a variational quantum classifier with labeled data, where each pronoun-noun pair is classified as *non-coreferent* or *coreferent*. The code and data used in this paper are available at the following link: `https://github.com/hwazni/Qcoref`

### 4.1 Dataset

The process of training PQCs involves optimising multiple parameters associated with each word in a given dataset, with the objective of minimising the loss value on the training set. When it comes to predicting the output for a test sample, a PQC is constructed based on the input sentence. Each word in the sentence is associated with a specific set of parameters learned during the training process. A significant challenge arises when an out-of-vocabulary word is encountered during inference, which includes testing or using the model for predictions. These words lack a predefined parameter assignment. To address this issue, there are several approaches, including random initialisation, replacement with a special token like "UNK" for

[1] A grammar formalism inspired by combinatory logic and developed in (Steedman, 2001)

[2] https://www.ibm.com/quantum/roadmap

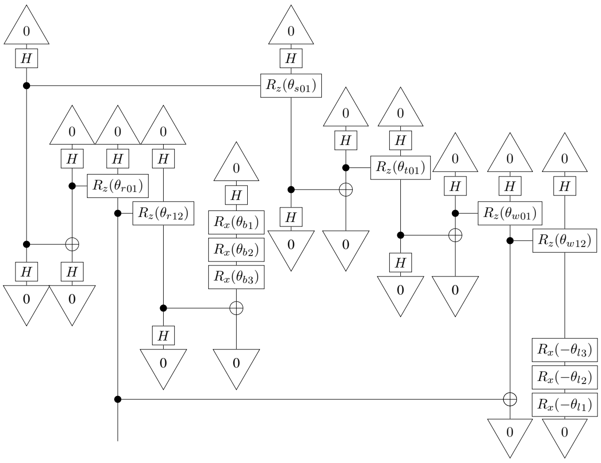

Figure 3: A PQC using the IQP ansatz, transformed from the string diagram presented in Figure 2. The parameters $\{\theta_{s01}\}$, $\{\theta_{r01}, \theta_{r12}\}$, and $\{\theta_{b1}, \theta_{b2}, \theta_{b3}\}$ are associated with the terms *the students*, *read*, and *the books* respectively, while $\{\theta_{t01}\}$, $\{\theta_{w01}, \theta_{w12}\}$ and $\{-\theta_{l3}, -\theta_{l2}, -\theta_{l1}\}$ are associated with *They*, *were*, and *learning* respectively.

unknown words, or establishing an overlap between the test and training vocabularies. In our case, we fix a set of words with grammatical relations between them, then use these and prompt the GPT-3 model to generate pairs of sentences that exhibit a substantial overlap in vocabulary.

In the initial step, we selected entries from the definite pronoun resolution dataset introduced in (Rahman and Ng, 2012), an extension of the Winograd Schema Challenge dataset (Levesque et al., 2012). We excluded sentences containing proper nouns and negation, and gave preference to shorter sentences. This process resulted in a total of 10 entries. Each entry was a pair of sentences. The first sentence, exemplified by $E_1$ : *The students read the books*, contains two referent nouns, namely, *the students* and *the books*. In the second sentence, an ambiguous pronoun is introduced, referring to one of the referents in $E_1$. For instance, it could be either $E_2$ : *They were learning* or *They were interesting*. Notably, the pronoun aligns with gender, number, and semantic class concerning each of the candidate referents mentioned in the first sentence. For each initially selected pair $(E_1, E_2)$, we created an additional set of pairs $(S_1, S_2)$ incorporating a more diverse range of grammatical structures. In these template pairs, $S_1$ retained the same referents as $E_1$, and $S_2$ maintained the same co-reference relation with $E_1$. Below is the list of template pairs for the *student-book* example.

1. The students *(verb, phrasal verb, verb phrase)* the books. They were *(adjective, gerund phrase)*.
2. The *(adjective)* students *(verb, phrasal verb, verb phrase)* the books. They were *(adjective, gerund phrase)*.
3. The students *(verb, phrasal verb, verb phrase)* the *(adjective)* books. They were *(adjective, gerund phrase)*.
4. The *(adjective)* students *(verb, phrasal verb, verb phrase)* the *(adjective)* books. They were *(adjective, gerund phrase)*.

The templates replace the verb *"read"* by another *verb*, *phrasal verb* or a *verb phrase*. Similarly, the adjectives *"learning"* and *"interesting"* can be replaced by another *adjective* or *gerund phrase*. Sample templates for different examples are listed in section 5.4.

Next, we utilize the prompt provided in the box below in GPT-3 along with template pairs. This technique referred as few-shot prompting, where we provide examples in the prompt to steer the model to better performance (Brown et al., 2020; Kaplan et al., 2020; Touvron et al., 2023). Note that the red tokens are modified for each example.

**Provide alternative sentences by replacing the words or phrases inside the brackets for each statement. Utilize different verbs, phrasal verbs, verb phrases, adjectives, or gerund phrases to create new sentences based on the given structure. Ensure that the pronoun 'they' in the second sentence refers to 'students' / Ensure that the pronoun 'they' in the second sentence refers to 'books'**

From the GPT-generated output, we eliminated incorrect referent sentences and duplicate examples, retaining only well-formed sentences that possess meaningful content. We carefully handpicked between 300 to 400 examples for each entry, ensuring a balanced distribution of pronoun references. Then we used the generated linguistic elements, including *verbs, phrasal verbs, adjectives, adverbs, nouns, compound nouns, verb phrases, adverbial phrases, gerund phrases, and prepositional phrases*, with 8 distinct structural patterns to generate over 8 million diverse combinations. We randomly choose 1800 pairs for each example, with one example with 200 pairs. This ended

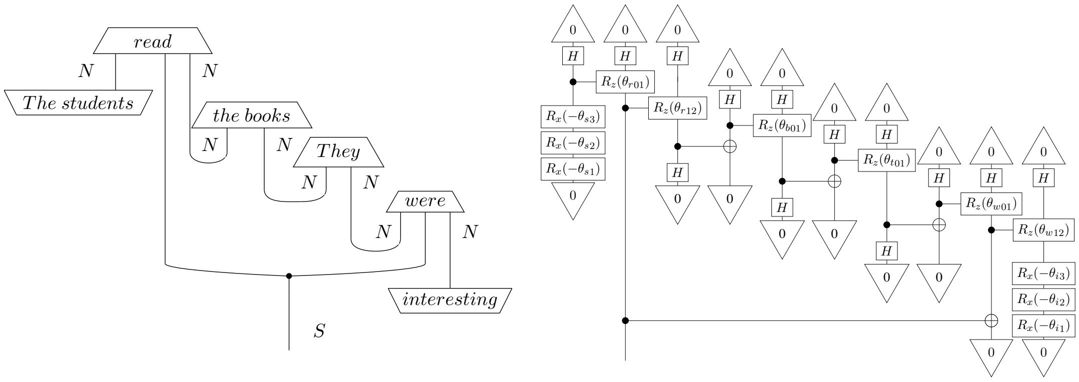

Figure 4: An optimised **SLLM** diagram where the pronoun refers to the object: *"The students read the books. They were interesting."* The diagram along with its transformation into a PQC.

| **Sentence Pair** $(S_1, S_2)$ | **Pronoun** | **Noun** | **Label** |
|---|---|---|---|
| The students researched the books.<br>They were seeking new insights. | They | students | 1 |
| The massive storm cancelled the flight.<br>It was full of passengers. | It | storm | 0 |
| The precise sniper eliminated the ruthless terrorist.<br>He was a vicious dealer. | He | terrorist | 1 |
| The exhausted sailors threw themselves off the boats.<br>They were in poor condition. | They | sailors | 0 |

Table 1: Dataset entries: each sentence pair is labeled with a "0" signifying that the pronoun do not refer to the candidate noun. Conversely, a "1" label indicates that the pronoun and the noun are co-referential.

up with 16,400 (0.2%) examples, comprising approximately 200,000 words, with 1,214 unique vocabulary. Through this approach, we achieved the generation of coherent sentences that uphold grammatical correctness and preserve semantic consistency, as a result a high quality was ensured for the dataset. The dataset was subsequently split into three subsets: 10,496 pairs (∼60%) for training, 2,624 pairs (∼20%) for validation, and 3,280 pairs (∼20%) for testing. The training and testing datasets share a common vocabulary of 95%, while none of the sentence pairs in the testing set appears in the training or validation sets. Some examples of the dataset are provided in Table 1.

### 4.2 Simulating the quantum circuits

Computation using currently available quantum computers, which are called NISQ for Noisy Intermediate-Scale Quantum, is slow, noisy and limited. They lack the practicality needed for extensive training and comprehensive comparative analyses (Preskill, 2018). For this reason, and especially at the early stages of modelling, proofs-of-concept are obtained by running simulations. A simple way to simulate a quantum computation is to use linear algebra; since quantum gates correspond to complex-valued tensors, each circuit can be represented as a tensor network where computation takes place as a result of a series of tensor contractions. The output of these contractions is the ideal probability distribution of the measurement outcomes on a noise-free quantum computer, i.e. an idealistic approximation of the sampled probability distribution obtained from a NISQ device. We conduct our experiments using noiseless non-shot-based simulations utilizing the *NumPyModel* of *Lambeq* (Kartsaklis et al., 2021b) with a JAX backend (Frostig et al., 2018).

### 4.3 Training

We implement a hybrid classical-quantum training approach in which the quantum computer is responsible for computing the meaning of the sentence by connecting the quantum states in a quantum circuit and the classical computer is used to calculate the training's loss function. During each iteration, a new set of quantum states is generated, driven by the loss function's outcome from the preceding iteration. This iterative procedure ensures that the quantum states are continually refined to enhance the model's performance and accuracy.

Specifically, the sentence pair $(S_1, S_2)$ within each dataset entry are combined to create a single output quantum state. These resultant states are the inputs to our binary classifier. In principle, they can be any quantum map that take two sentences as input and produce a sentence as the output (recall the whole circuit is represented by an open sentence wire). A CNOT gate is used to combine the two sentences, as it encodes a commutative Frobenius multiplication ($\bullet$) and acts similar to a logical conjunction. The resulting quantum circuit is denoted by $S_1 \bullet S_2$ and evaluated for an initial set of parameters $\Theta = (\theta_1, \theta_2, ..., \theta_k)$ on a quantum computer giving an output state $|S_1 \bullet S_2\,(\Theta)\rangle$. The expected prediction is given by the Born rule, i.e. as follows:

$$l^i_\Theta(S_1 \bullet S_2) := |\langle i|S_1 \bullet S_2(\Theta)\rangle|^2 + \epsilon$$

where, $i \in \{0, 1\}$, $\epsilon$ is a smoothing term with the value $10^{-9}$, and $l_\Theta(S_1 \bullet S_2)$ is the following probability distribution:

$$l_\Theta(S_1 \bullet S_2) := \frac{(l^0_\Theta(S_1 \bullet S_2), l^1_\Theta(S_1 \bullet S_2))}{\sum_i l^i_\Theta(S_1 \bullet S_2)}$$

The predicted label is obtained by rounding the probability distribution to the nearest integer $\lfloor l_\Theta(S_1 \bullet S_2)\rceil$ and represented as one-hot encoding. This means if $\lfloor l_\Theta(S_1 \bullet S_2)\rceil < 0.5$, the predicted label $[0, 1]$ corresponds to *non-coreferent* mentions, and if $\lfloor l_\Theta(S_1 \bullet S_2)\rceil \geq 0.5$, the predicted label $[1, 0]$ corresponds to *coreferent* mentions.

To find the optimal parameters for our model, the predicted label is compared with the training label using a binary cross-entropy loss function and minimised using a non-gradient-based optimisation algorithm known as SPSA (Simultaneous Perturbation Stochastic Approximation) (Spall, 1998).

For the hyper-parameters, we set the initial learning rate *a* to 0.1, the initial parameter-shift scaling *c* to 0.06, and the stability constant *A* to 20. We run for 2000 epochs of SPSA during which we evaluate the training loss and accuracy. This process is repeated 15 times with random seed values. This is essential since the gradient computed by the SPSA procedure is an approximation and the performance in QML is known to be very sensitive to the initial parameter assignment (Holmes et al., 2022; Grant et al., 2019; McClean et al., 2018).

## 5 Results and Discussion

### 5.1 Quantum Approaches: SLLM vs Bag-of-Words

The graphs in Figure 5 illustrate how the models converged smoothly. Across different runs, a common trend emerges—training loss decreases and training accuracy increases steadily. Initially, the average training loss is 1.144, which drops to 0.483 after 2000 iterations. Minimum and maximum values range from 0.369 to 0.571 for different runs. Similarly, the average training accuracy starts at 0.514 and ends at 0.752 after 2000 iterations. The highest recorded accuracy is 0.827, and the lowest is 0.682 amongst all the runs. The testing accuracy rates vary between 0.628 and 0.782, averaging around 0.70. These results demonstrate that the model is able to generalise its predictions beyond training, with well-balanced performance levels.

To understand whether the promising performance of the **SLLM** classifier is due to the structural symbolic type-driven representations or the use of PQCs, we conducted a comparative analysis with quantum circuits generated from a simple bag-of-words diagram (see section 5.4). In this approach, each word is represented with a single qubit, regardless of its grammatical type (e.g., noun, adjective, or verb). Consequently, this model disregards sentence structure and connects all qubits using CNOT gates (the simplest counterparts to addition in quantum circuits). We trained the model under identical hyper-parameters and the same number of training runs. However, its performance fell short, yielding an average testing accuracy of 0.557.

### 5.2 Classical Approaches: SVM, CoreNLP, Neural Coreference, SpanBERT

We implemented a Support Vector Machine (SVM) for a binary classification task and evaluated its performance in comparison to our VQC. The inputs to the SVM were pre-trained Sentence-BERT embeddings (Reimers and Gurevych, 2019), one per each dataset entry. We also experimented with a compositional model, by adding SBERT word embeddings of each entry, as shown below:

**SVM Full** : $\overrightarrow{E}$
**SVM Add** : $(\overrightarrow{w_1} + \overrightarrow{w_2} + \overrightarrow{w_3}...) + (\overrightarrow{w_1} + \overrightarrow{w_4} + \overrightarrow{w_5}...)$

In the above, $E$ is an entry such as: *"The students researched the books. The students were seeking*

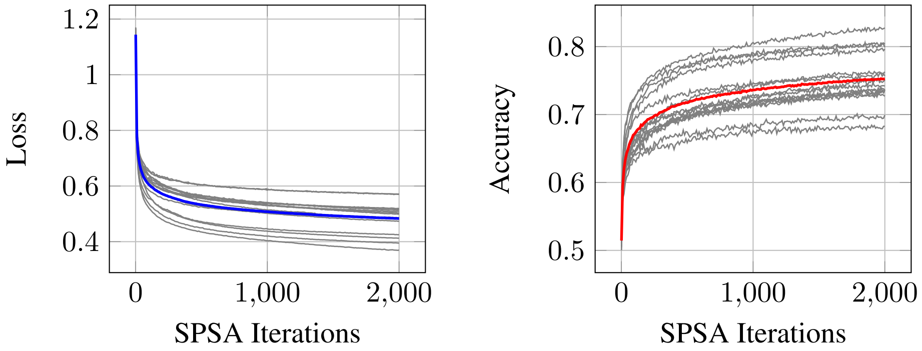

Figure 5: Performance of 15 different runs of a classical simulation of the training set showing the average training loss (blue) and the average training accuracy (red).

*new insights."* labeled as 1 or *"The massive storm cancelled the flight. The storm was full of passengers."* labeled as 0. In **SVM Add**, $\overrightarrow{w_1}$ is a candidate referent, e.g. *students* or *storm*, and $\overrightarrow{w_2}$, $\overrightarrow{w_3}$, $\overrightarrow{w_4}$, $\overrightarrow{w_5}$ are all the other words.

The objective here was to assess the discourse relation within each entry. We achieved this objective by replacing the pronoun with either the correct or the incorrect referent, thereby evaluating the the discourse relation between them. The training process involved optimising two hyper-parameters: the regularisation parameter $c$ and the choice of kernel type, which could be either linear or a radial basis function (RBF). We leveraged a grid search technique with a 10 fold cross-validation scheme to identify the most suitable combination of hyper-parameters. The resulting SVM model with the best-tuned hyper-parameters was used for evaluation on the testing dataset. The results in Table 2 show that **SVM Add** achieved a lower F1 score of 0.821 in comparison to **SVM Full**, which achieved a solid F1 score of 0.914.

| **Model** | **F1 Score** |
|---|---|
| **SVM Full** | 0.914 |
| **SVM Add** | 0.821 |

Table 2: Evaluation performance of classical compositional and non-compositional SVM models

Additionally, we evaluated CoreNLP (Manning et al., 2014), Neural Coreference (Clark and Manning, 2016a) (Clark and Manning, 2016b), and SpanBERT (Lee et al., 2018). CoreNLP combines rule-based techniques with statistical models to resolve coreference; Neural Coreference employs deep learning to capture patterns and dependencies in text, and SpanBERT is a specialised version of BERT (Devlin et al., 2019) fine-tuned for coreference resolution. We ran the pre-trained models using Stanza[3], HuggingFace[4], and AllenNLP[5] libraries respectively. The outcomes are presented in Table 3.

| **Model** | **F1 Score** |
|---|---|
| **CoreNLP** | 0.563 |
| **Neural Coreference** | 0.585 |
| **SpanBERT** | **0.927** |
| **QuantumCoref** | 0.872 |

Table 3: Evaluation performance of classical neural models

The performance levels amongst these systems were diverse. CoreNLP achieved the lowest F1 score of 0.563, while SpanBERT demonstrated the highest score of 0.927. Neural Coreference achieved a moderate score of 0.585, trailing behind SpanBERT but outperforming CoreNLP.

To facilitate the use of our approach, we implemented an end-to-end system named *QuantumCoref* that consists of two sub-modules: (a) a mentions-detection module that uses SpaCy's[6] part-of-speech parser to identify a set of potential coreference mentions, and (b) our highest-accurate trained **SLLM** classifier, which computes coreference scores for each pair of potential mentions. It achieved an F1 score of 0.872 near SpanBERT.

### 5.3 Mixed Quantum + Classical Models

To maximize the strengths of quantum and classical systems, we combine their predictions in the following manner: when a classical system predicts an incorrect referent, we opt for the prediction of

[3] https://corenlp.run/

[4] https://huggingface.co/coref/

[5] https://demo.allennlp.org/coreference-resolution/

[6] https://spacy.io/usage/linguistic-features

| Model | F1 Score |
|---|---|
| **CoreNLP + QuantumCoref** | 0.930 |
| **Neural Coreference + QuantumCoref** | 0.946 |
| **SpanBERT+ QuantumCoref** | **0.986** |
| **SVM Full + QuantumCoref** | 0.959 |
| **SVM Add + QuantumCoref** | 0.910 |

Table 4: Evaluation performance of mixed quantum + classical models

*QuantumCoref.* Similarly, when a classical model fails to identify a referent, resulting in an empty cluster, we rely on *QuantumCoref* for classification. As an example, consider the discourse *"The students learned from the books. They were filled with knowledge."* In this scenario, while SpanBERT detected that the pronoun *"they"* refers to *"students"*, *QuantumCoref* correctly identified the coreference relationship as *"they-books"*. As a result, this mixed quantum-classical approach recognised *"they"* and *"books"* as co-referent entities. By combining the two approaches, we were able to extract the best outcomes from each model, thus enhancing the overall performance. CoreNLP improved from 0.563 to 0.930, Neural Coreference from 0.585 to 0.946, and SpanBERT from 0.927 to 0.986. The SVM models reacted in a similar fashion: the performance of SVM Add increased from 0.821 to 0.910 and that of SVM Full from 0.914 to 0.959.

### 5.4 Discussion

In a more detailed analysis, among the incorrect predictions, SpanBERT identified pronouns referring to the first noun in 95% of the cases and to the second noun in 5% of the cases. This highlights how SpanBERT struggles in identifying the correct referent, particularly when it's positioned towards the end of the sentence, leading to a higher preference for selecting the first noun.

In situations characterised by linguistic ambiguities, SpanBERT struggles in recognising referential connections. Notably, in instances where multiple plausible nouns could serve as antecedents for pronouns, SpanBERT returns an empty cluster. For instance, in *"The productive bee flew over the flower. It was magnificent."* the complexity arises from the fact that both *"productive bee"* and *"flower"* are reasonable candidates for the antecedent. Similarly, in *"The sailors jumped from the boats. They were having technical problems."*, the ambiguity arises from the potential referents for the pronoun *"They"* which could be either the *"sailors"* or the *"boats"*. In contrast, *QuantumCoref* relies on sentence structure and the connections between entities and their referents. Impressively, *QuantumCoref* solves 319 examples where SpanBERT misclassified, showcasing a success rate of 81.37% and handled 35 examples where SpanBERT returned empty clusters, with a success rate of 68.62%. When our dataset was converted into CoNLL format and SpanBERT was fine-tuned on it, unsurprisingly, it achieved an F1 score of 0.998.

We would like to emphasise that these experiments were not specifically aimed at showcasing *quantum advantage* over classical coreference resolution systems. Our aim was to demonstrate the capabilities of our quantum-based approach, which also offers transparency. Furthermore, SpanBERT, with its exceptional coreference resolution capabilities, requires high computational resources. The fine-tuned SpanBERT model comprises a total of 366 million parameters, which is substantially larger compared to *QuantumCoref*, with a total of 2693 parameters. This highlights the efficiency of the quantum-based approach. There is potential for further improvements, especially when a greater number of qubits are used in modelling. Our setting can resolve general coreference relations in the same way as anaphoric ones. When multiple expressions co-refer, the main entity becomes a Fock space and the rest are pronoun types. We leave experimentation in this direction to future work.

## Limitations

We classify the limitations into the following items:

- **Syntax.** It would be tempting to call **SLLM**, the logic of discourse. It, however, does not have a connective for conjoining sentences. In this paper, we resolved the problem in the semantics, by using the Frobenius multiplication for conjoining sentences. A better logic for discourse should include this connective in its syntax.

- **Semantics.** The vector space semantics of **SLLM** over unifies the types, e.g. its copiable and functional types are assigned the same vector space semantics, e.g. two copies of a noun phrase and an adjective both have the same $[\![N \otimes N]\!]$ semantics.

- **Automated Parsing.** **SLLM** does not have an automatic parser and at the moment its use implies manual type annotations to words. LC has an automatic parser that can be extended to the new types introduced in **SLLM**. An automatic learning procedure for types, however, requires a corpus annotated with **SLLM** types. At this stage, we foresee any co-reference annotated corpus can easily be transferred to an **SLLM** annotated one.

- **Quantum Computation.** We relied on simulations for training circuit parameters instead of using real quantum computers. Currently, we are experimenting with a shot-based simulation with an incorporated noise model. This approach takes into consideration critical factors such as quantum gate errors, decoherence, and shot noise, all of which affect practical quantum computing. It can be ported for execution on a quantum computer.

- **Different Types of Anaphora.** In this paper, we focused on definite pronoun resolution and identity anaphora. Non-definite and non-identity anaphora cases, such as bridging and event anaphora, pose challenges and require further theoretical work.

- **OntoNotes.** Our original goal was to run the model on OntoNotes. This turned out to be impossible due to two main reasons. One was that we needed a large overlap between the vocabularies used in training and testing. Secondly, the entries of OntoNotes consist of long complex sentences, which would lead to large quantum circuits. These could not even be efficiently simulated with the current technology.

## Acknowledgement

The authors gratefully acknowledge the three anonymous reviewers for their valuable comments. Mehrnoosh Sadrzadeh is grateful to the Royal Academy of Engineering Research Chair/Senior Research Fellowship RCSRF2122-14-152 on Engineered Mathematics for Modelling Typed Structures. Hadi Wazni would like to express gratitude for support by UCL CS department for the PhD scholarships. Both authors would like to thank Lo Ian Kin for many helpful discussions and some help in implmentation.

## References

Mina Abbaszade, Mariam Zomorodi, Vahid Salari, and Philip Kurian. 2023. Toward quantum machine translation of syntactically distinct languages.

Samson Abramsky and Bob Coecke. 2008. Categorical quantum mechanics.

Ismael C. S. Araujo and Adenilton J. da Silva. 2020. Quantum ensemble of trained classifiers. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Kushal Batra, Kimberley M. Zorn, Daniel H. Foil, Eni Minerali, Victor O. Gawriljuk, Thomas R. Lane, and sean ekins. 2020. Quantum machine learning for drug discovery. *ChemRxiv*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Vanessa Buhrmester, David Münch, and Michael Arens. 2019. Analysis of explainers of black box deep neural networks for computer vision: A survey.

Noam Chomsky. 1957. *Syntactic Structures*. Mouton and Co., The Hague.

Kevin Clark and Christopher D. Manning. 2016a. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas. Association for Computational Linguistics.

Kevin Clark and Christopher D. Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Berlin, Germany. Association for Computational Linguistics.

Stephen Clark. 2021. Something old, something new: Grammar-based ccg parsing with transformer models.

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

Bob Coecke, Giovanni de Felice, Konstantinos Meichanetzidis, and Alexis Toumi. 2020. Foundations for near-term quantum natural language processing.

Bob Coecke, Edward Grefenstette, and Mehrnoosh Sadrzadeh. 2013. Lambek vs. lambek: Functorial vector space semantics and string diagrams for lambek calculus. *Ann. Pure and Applied Logic*, 164:1079 – 1100.

Bob Coecke and Aleks Kissinger. 2017. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press.

Giovanni de Felice, Alexis Toumi, and Bob Coecke. 2021. DisCoPy: Monoidal categories in python. *Electronic Proceedings in Theoretical Computer Science*, 333:183–197.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Roy Frostig, Matthew Johnson, and Chris Leary. 2018. Compiling machine learning programs via high-level tracing.

Srinjoy Ganguly, Sai Nandan Morapakula, and Luis Miguel Pozo Coronado. 2023. Quantum natural language processing based sentiment analysis using lambeq toolkit.

Edward Grant, Leonard Wossnig, Mateusz Ostaszewski, and Marcello Benedetti. 2019. An initialization strategy for addressing barren plateaus in parametrized quantum circuits. *Quantum*, 3:214.

M. Grossi, N. Ibrahim, V. Radescu, R. Loredo, K. Voigt, C. von Altrock, and A. Rudnik. 2022. Mixed quantum–classical method for fraud detection with quantum feature selection. *IEEE Transactions on Quantum Engineering*, 3(01):1–12.

Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. 2019. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212.

Zoë Holmes, Kunal Sharma, M. Cerezo, and Patrick J. Coles. 2022. Connecting ansatz expressibility to gradient magnitudes and barren plateaus. *PRX Quantum*, 3:010313.

M. Kanovich, S. Kuznetsov, V. Nigam, and A. Scedrov. 2020. Soft Subexponentials and Multiplexing. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models.

Amin Karamlou, Marcel Pfaffhauser, and James Wootton. 2022. Quantum natural language generation on near-term devices.

D. Kartsaklis and M. Sadrzadeh. 2013. Prior disambiguation of word tensors for constructing sentence vectors. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1590–1601.

Dimitri Kartsaklis, Ian Fan, Richie Yeung, Anna Pearson, Robin Lorenz, Alexis Toumi, Giovanni de Felice, Konstantinos Meichanetzidis, Stephen Clark, and Bob Coecke. 2021a. lambeq: An efficient high-level python library for quantum nlp.

Dimitri Kartsaklis, Ian Fan, Richie Yeung, Anna Pearson, Robin Lorenz, Alexis Toumi, Giovanni de Felice, Konstantinos Meichanetzidis, Stephen Clark, and Bob Coecke. 2021b. lambeq: An Efficient High-Level Python Library for Quantum NLP. *arXiv preprint arXiv:2110.04236*.

Iordanis Kerenidis and Alessandro Luongo. 2020. Classification of the mnist data set with quantum slow feature analysis. *Phys. Rev. A*, 101:062327.

J. Lambek. 1988. *Categorial and Categorical Grammars*, pages 297–317. Springer Netherlands, Dordrecht.

Joachim Lambek. 1958. The mathematics of sentence structure. *The American Mathematical Monthly*, 65(3):154–170.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of the International Workshop on Temporal Representation and Reasoning*.

Martha Lewis. 2020. Towards logical negation for compositional distributional semantics.

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

Antonio Macaluso, Luca Clissa, Stefano Lodi, and Claudio Sartori. 2020. Quantum Ensemble for Classification.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. 2018. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1).

L. McPheat, H. Wazni, and M. Sadrzadeh. 2020. Vector space semantics for lambek calculus with soft subexponentials. In *Proceedings of the tenth international conference on Logical Aspect of Computational Linguistics*.

Konstantinos Meichanetzidis, Alexis Toumi, Giovanni de Felice, and Bob Coecke. 2023. Grammar-aware sentence classification on quantum computers. *Quantum Machine Intelligence*, 5(1).

Eduardo Reck Miranda, Richie Yeung, Anna Pearson, Konstantinos Meichanetzidis, and Bob Coecke. 2021. A quantum natural language processing approach to musical intelligence.

Michael A. Nielsen and Isaac L. Chuang. 2010. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press.

Robin Piedeleu and Fabio Zanasi. 2023. An introduction to string diagrams for computer scientists.

John Preskill. 2018. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79.

Altaf Rahman and Vincent Ng. 2012. Resolving complex cases of definite pronouns: The winograd schema challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 777–789, Jeju Island, Korea. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Fariska Z. Ruskanda, Muhammad Rifat Abiwardani, Muhammad Akram Al Bari, Kinantan Arya Bagaspati, Rahmat Mulyawan, Infall Syafalni, and Harashta Tatimma Larasati. 2022. Quantum representation for sentiment classification. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 67–78.

M. Sadrzadeh, S. Clark, and B. Coecke. 2013. The frobenius anatomy of word meanings i: subject and object relative pronouns. *Journal of Logic and Computation*, 23(6):1293–1317.

Mehrnoosh Sadrzadeh, Stephen Clark, and Bob Coecke. 2014. The frobenius anatomy of word meanings II: possessive relative pronouns. *Journal of Logic and Computation*, 26(2):785–815.

Dan Shepherd and Michael J. Bremner. 2009. Temporally unstructured quantum computation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, (2105):1413–1439.

J.C. Spall. 1998. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on Aerospace and Electronic Systems*, 34(3):817–823.

Mark Steedman. 2001. *The Syntactic Process*. The MIT Press.

Jonas Stein, Ivo Christ, Nicolas Kraus, Maximilian Balthasar Mansky, Robert Müller, and Claudia Linnhoff-Popien. 2023. Applying qnlp to sentiment analysis in finance.

R.S. Sutor. 2019. *Dancing with Qubits: How Quantum Computing Works and how it Can Change the World*. Expert Insight. Packt Publishing.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

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

Richie Yeung and Dimitri Kartsaklis. 2021. A ccg-based version of the discocat framework.

## Appendix

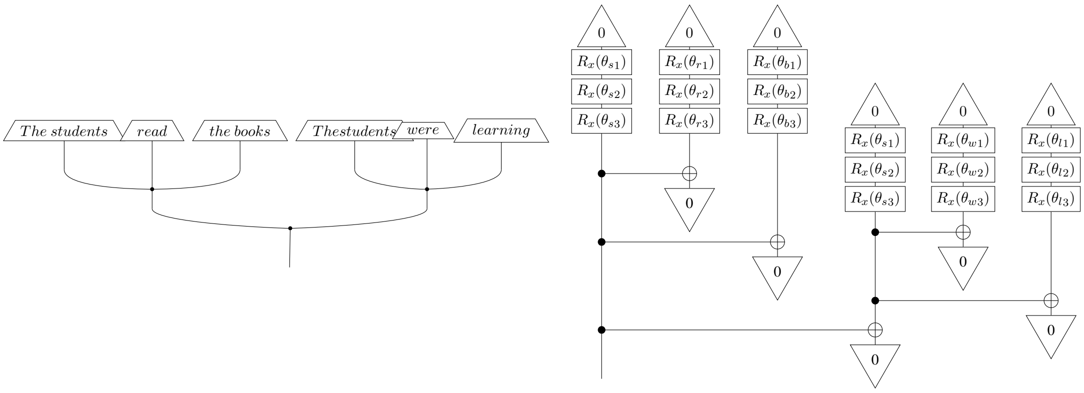

Figure 6: A Bag-of-Words diagram representing the discourse: *"The students read the books. They were learninig."* The diagram along with its transformation into a PQC.

- Template Example 1:
  - The sailors *{verb, phrasal verb, verb phrase}* the boats. They were *{adjective, gerund phrase}*
  - The *{adjective}* sailors *{verb, phrasal verb, verb phrase}* the boats. They were *{adjective, gerund phrase}*
  - The sailors *{verb, phrasal verb, verb phrase}* the *{adjective}* boats. They were *{adjective, gerund phrase}*
  - The *{adjective}* sailors *{verb, phrasal verb, verb phrase}* the *{adjective}* boats. They were *{adjective, gerund phrase}*
- Template Example 2:
  - The storm *{verb, verb phrase}* the flight. It was *{gerund phrase}*
  - The *{adjective}* storm *{verb, verb phrase}* the flight. It was *{gerund phrase}*
  - The storm *{verb, verb phrase}* the *{adjective}* flight. It was *{gerund phrase}*
  - The *{adjective}* storm *{verb, verb phrase}* the *{adjective}* flight. It was *{gerund phrase}*
- 8 distinct structural patterns for Template Example 2:
  - The storm *{verb, verb phrase}* the flight. It was *{gerund phrase (storm)}*.
  - The storm *{verb, verb phrase}* the flight. It was *{gerund phrase (flight)}*.
  - The *{adjective (storm)}* storm *{verb, verb phrase}* the flight. It was *{gerund phrase (storm)}*.
  - The *{adjective (storm)}* storm *{verb, verb phrase}* the flight. It was *{gerund phrase (flight)}*.
  - The storm *{verb, verb phrase}* the *{adjective (flight)}* flight. It was *{gerund phrase (flight)}*.
  - The storm *{verb, verb phrase}* the *{adjective (flight)}* flight. It was *{gerund phrase (storm)}*.
  - The *{adjective (storm)}* storm *{verb, verb phrase}* the *{adjective (flight)}* flight. It was *{gerund phrase (storm)}*.
  - The *{adjective (storm)}* storm *{verb, verb phrase}* the *{adjective (flight)}* flight. It was *{gerund phrase (flight)}*.