Text Classification with DisCoCat and Quantum Support Vector Machines

Internship Report Submitted Under the

IEEE Student Summer Internship Program - 2024 (IS3IP-2024)

30th June 2024 – 15th August 2024

by

Hrithvik Kondalkar
Jadavpur University

Under the Mentorship of

Dr. Partha Pakray

IEEE Silchar Subsection

August-2024

1. Introduction

Quantum computing has garnered increasing attention for its potential to revolutionize various computational tasks, including those in the domain of machine learning. Among these tasks, text classification—a key challenge in natural language processing (NLP)—has emerged as a compelling use case for quantum machine learning techniques. This report explores the application of Quantum Support Vector Machines (QSVM) for text classification, where the encoding of text data is performed using the Distributional Compositional Categorical (DisCoCat) model.

Unlike traditional encoding methods in classical natural language processing (NLP), such as word embeddings or vectorization techniques, which typically represent words or phrases as points in high-dimensional vector spaces, the DisCoCat model captures both the syntactic structure and meaning of sentences in a way that is inherently compatible with quantum computation. This encoding is particularly suitable for quantum machine learning, as it allows for a meaningful and efficient representation of text data in the quantum domain. Lambeq, an open-source python library, facilitates the implementation of the DisCoCat model, enabling the transformation of natural language into quantum circuits.

In this project, we leveraged the Qiskit framework to encode texual data using ansatz generated from Lambeq and implement and train a QSVM for text classification. The integration of the DisCoCat model from Lambeq in Qiskit represents a novel approach to NLP tasks, offering the potential to harness the unique capabilities of quantum computing in processing complex, high-dimensional data like text.

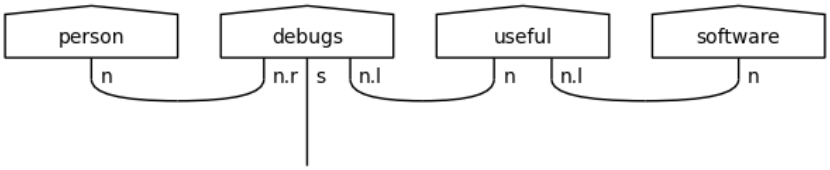
2. Objectives of Internship Program

The primary objective was to explore and demonstrate the practical implementation of a quantum natural language processing (QNLP) pipeline, by encoding text data using the DisCoCat model and classifying it with Quantum Support Vector Machines (QSVM), we aim to uncover the technical requirements, challenges, and potential solutions involved in realizing a functional QNLP system. This report documents the results from our experiments, offering insights into the practical considerations of building a QNLP pipeline, the obstacles encountered during the implementation process and propose directions for future advancements in quantum-enhanced NLP.

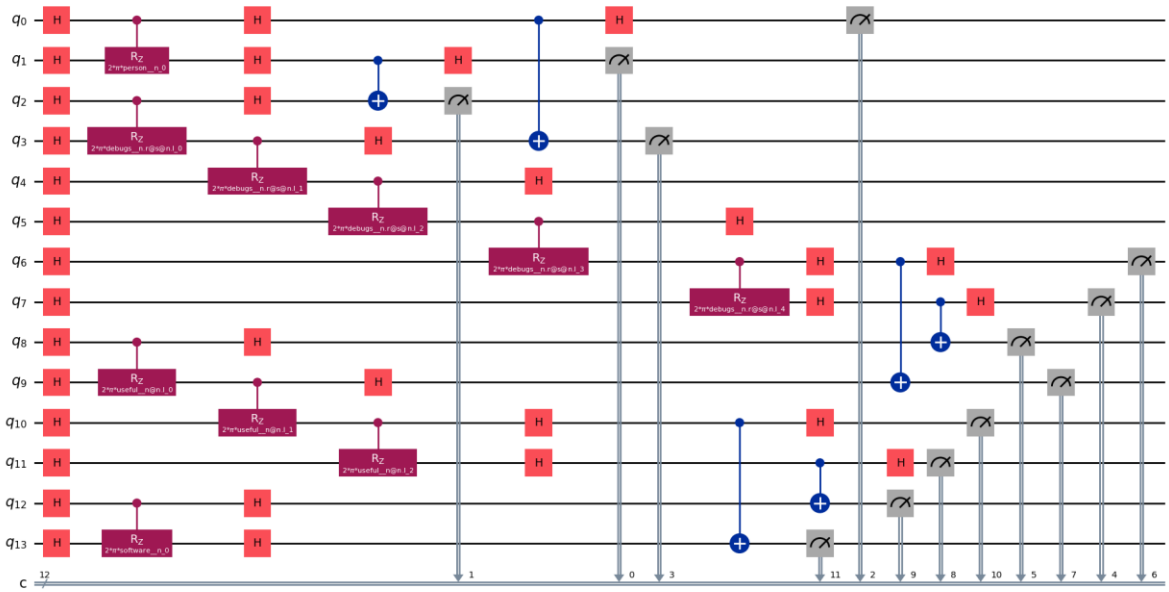
3. Methodology

The dataset selection was based on the cost of simulating the respective ansatz on a classical system. For this purpose, the example dataset provided by Lambeq was used to perform a binary classification task. The data was encoded by parsing the sentences into their respective DisCoCat diagrams, a process carried out using the BobCatParser from Lambeq. To further reduce computational overhead, the diagrams were simplified using the rules 'prepositional_phrase', 'determiner', and 'auxiliary', effectively removing elements like prepositions, determiners, and auxiliary verbs.

The simplified diagrams were then converted into an Instantaneous Quantum Polynomial (IQP) ansatz using the IQPAnsatz function, which associated 2 qubits with nouns and 2 qubits with sentences/verbs. This ansatz was subsequently translated into a Quantum Circuit in Qiskit using the pytket library to leverage the simulators provided by the Qiskit SDK.



The DisCoCat diagram for the sentence “person debugs useful software”

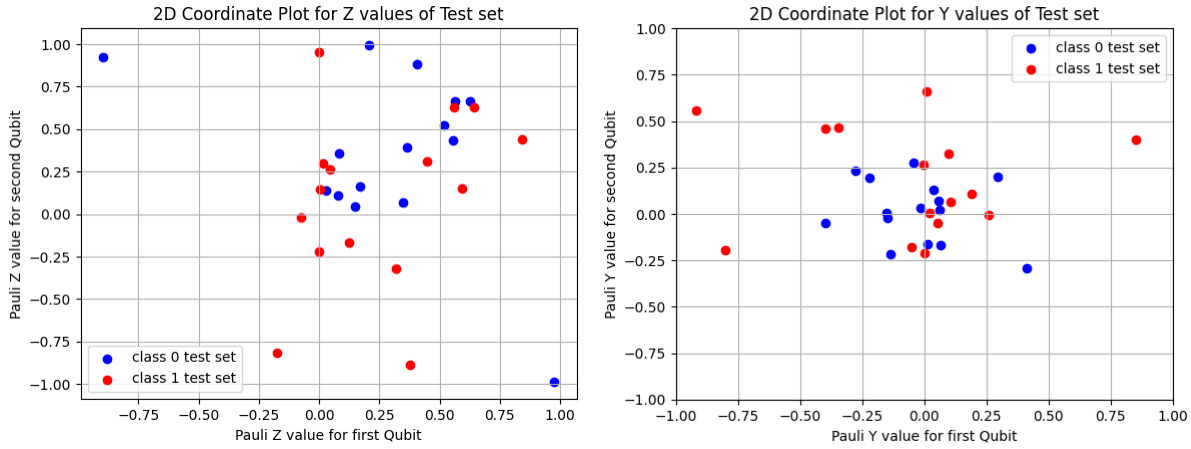


Corresponding Qiskit ansatz generated by IQPAnsatz

It is important to note that the two unmeasured qubits retained the encoded information of the sentence, provided that all parameters of the ansatz were adjusted so that all measured qubits were in the zero state. Since any machine learning algorithm is essentially an optimization technique, the search for optimal parameters involved using a Sampler Quantum Neural Network (QNN) with a custom cost function designed to maximize the probability of all measured qubits being zero. The inputs for the QNN were derived from the vector representation of a word, and the weights were initially randomized. The QNN employed the Constrained Optimization by Linear Approximation (COBYLA) algorithm to adjust the weights and minimize the cost function, which in this case was defined as:

$$CostFunction(weights) = -\log\left(probability\ of\ zero\ state\ (forward\ pass\ of\ neural\ network(weights))\right)$$

Once the weights were optimized, they were fed into the ansatz parameters, and the pair of unmeasured qubits now contained the encoded sentence. These qubits were measured using the Pauli Z and Pauli Y operators, with their values stored for future use as inputs into the Quantum Support Vector Machine (QSVM). This encoding process reduced the sentences into fixed-size vectors of length 4, making them suitable for a feature map requiring exactly 4 inputs.



2D Visualization of encoded test set

A quantum kernel was then defined, incorporating a feature map to encode the real values obtained from the previous step into qubits, and a fidelity measure to assess the similarity between two quantum states (the quantum equivalent of Euclidean distance). This kernel contained parameters optimized by the QSVC (Quantum Support Vector Classifier) instance used to fit the training data. Various hyperparameters related to different kernels, which utilized the ZZFeatureMap and the PauliFeatureMap, were explored to identify optimal values. Additionally, a custom FeatureMap was implemented to enhance the accuracy of the QSVC.

4. Simulation results & Discussion

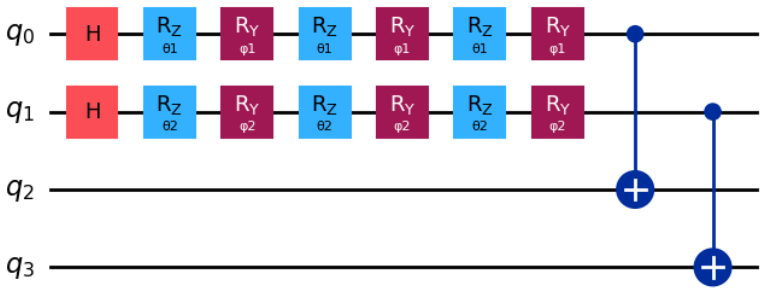
By adjusting the hyperparameters for various feature maps, an accuracy of 0.666 was achieved. This represents the maximum accuracy that the model can attain, as only the Pauli Z and Pauli Y operators were measured, while the Pauli X operator was not. The omission of the Pauli X measurements resulted in a loss of part of the encoded data, which likely contributed to the observed accuracy limit.

FeatureDimensions	Repetitions	Entanglement	Accuracy
4	4	Full	0.433
4	3	Linear	0.533
4	4	Linear	0.666
6	4	Linear	0.666

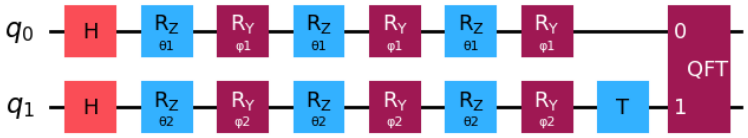
Hyperparameter tuning for ZZFeatureMap as feature map

FeatureDim	Repetitions	Entanglement	Paulis	Accuracy
4	4	Full	['Z','Y','ZZ', 'YY']	0.466
4	4	Linear	['Z','Y','ZZ', 'YY']	0.566
2	4	Linear	['Z','ZZ']	0.666

Hyperparameter tuning for PauliFeatureMap as feature map



Using Custom Feature Map, Produced accuracy score of 0.666



Feature Map reduced to 2 qubits, Produced accuracy score of 0.666

5. Conclusion

In this project, I explored the application of Quantum Support Vector Machines (QSVM) for a binary classification task, leveraging the Distributional Compositional Categorical (DisCoCat) model to encode textual data in a manner compatible with quantum computing. This also demonstrated the powerful encoding capabilities of the DisCoCat model. By integrating the DisCoCat model, facilitated by the Lambeq library, with the Qiskit framework, I was able to transform natural language into quantum circuits and implement a quantum-enhanced text classification pipeline. The experiments performed demonstrated that QSVM, when combined with quantum-native text encoding techniques, effectively classified textual data, offering a glimpse into the potential of quantum computing in the field of Natural Language Processing.

Although the current state of quantum hardware imposes limitations on the scalability and complexity of quantum algorithms, the results indicate promising avenues for future research and development in quantum enhanced NLP.

Throughout the duration of this project, several technical challenges were encountered. However, these challenges also highlighted opportunities for further innovation and optimization in quantum algorithms and hardware. Some suggested improvements in the implementation followed in this project would have been to directly integrate the encoding quantum circuit with the Quantum Support Vector Classifier, which would have eliminated the issues caused by measuring expectancy values of the qubits and then re-encoding them using a feature map. Future research on this topic should focus on exploring larger and more complex datasets to fully appreciate the potential of quantum computing in other NLP tasks.

6. References

Bob Coecke, Mehrnoosh Sadrzadeh, Stephen Clark, *Mathematical Foundations for a Compositional Distributional Model of Meaning,* arXiv:1003.4394

Havlíček V, Córcoles A, Temme K, Harrow A, Kandala A, Chow J, Gambetta J. Supervised Learning with Quantum-Enhanced Feature Spaces. *Nature*, 567, 2019

William Zeng and Bob Coecke, *Quantum Algorithms for Compositional Natural Language Processing*, arXiv: 1608.01406

Shervin Le Du, Senaida Hernández Santana, Giannicola Scarpa, *A gentle introduction to Quantum Natural Language Processing, arXiv: 2202.11766*

Bob Coecke, *The Mathematics of Text Structure, arXiv:1904.03478*

Amin Karamlou, Marcel Pfaffhauser, James Wootton, *Quantum Natural Language Generation on Near-Term Devices, arXiv:2211.00727*