iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** I **Month of publication:** January 2022

DOI: https://doi.org/10.22214/ijraset.2022.40103

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue I Jan 2022- Available at www.ijraset.com

A Review Article on Quantum Natural Language Processing

Anshuman Rai[1], Amey Talekar[2], Javed Khan[3], Prof. Ahlam Ansari[4]
[1, 2, 3]*Students,* [4]*Assistant professor, Department of Computer Engineering, M. H. Saboo Siddik College Of Engineering*

Abstract: ***Quantum Natural Language Processing is the implementation of NLP algorithms on quantum hardware or alternatively on hybrid quantum-classical hardware. NLP has been a heavily researched and implemented topic of the past few decades and the most recent developments using new techniques and the power of deep learning have made huge strides in the field. But for all this new development, there is a looming possibility of greater achievements in the form of the rising field of quantum computing which is yet to see its potential come to fruition. A gaping hole in the implementation process of NLP systems is the computing power required to train deep learning and Natural Language Processing models which makes the development of such models time consuming and power hungry. The huge leap in parallel computing power that quantum computers provide gives us immense opportunities to accelerate the training of deep and complex models. Such techniques will help organizations with access to quantum hardware to be able to use quantum circuits to either train a complete model or use a classical system like the norm but outsource all of the most computationally heavy part of the process to quantum hardware which will provide exponential speed up to the development of conversational AI models.***

Keywords: Quantum computing, Natural Language Processing, Quantum Machine Learning, Quantum Natural Language Processing, Noisy Intermediate-Scale Quantum systems, Lambeq, hybrid classical-quantum systems, DisCoCat

I. INTRODUCTION

The development and use of conversational AI systems has been ubiquitous over the previous two decades. Natural Language Processing and the development of conversational systems is a very active research topic. The most common techniques of developing AI based conversational systems have been deep networks and other ML methodologies using NLP techniques to get meaning out of a dataset of sentences. Such training techniques involve defining a grammar for the language which is used by the NLP model to learn to mimic human interaction.

The acceleration of the process of developing an AI model using quantum theory, on the other hand, is a fairly new field of research with only a handful of experimental work done to back up the theoretical work done by physicists and computer scientists over the past decade.

This is an extremely promising and exciting topic of research for the future of computers as the rising power of quantum computers will, sooner rather than later, overtake classical computers in terms of feasibly quick completion of computationally heavy tasks like training deep neural networks on a grammar with thousands of rules and datasets of millions upon millions of sentences and dialogues for a reliably intelligent conversational AI to be developed.

It is important to note that quantum computing is a fairly nascent technology and only a handful of organizations in the world have an actual working quantum computer while others have simulated quantum computers. Even so, the future prospects of quantum computing, if the number of qubits can be scaled, covers almost every field of research. The parallel processing power of quantum computers is unique to its quantum nature and can simply not be achieved with a feasibly sized classical computer.

While classical computers work with bits which are represented as either 0 or 1 at a time, quantum computers work with qubits which qubits which are represented as a vector $(a\ b)^T$ where a and b are complex numbers and $|a|^2$ and $|b|^2$ are the probabilities of the qubit taking on the value 0 or 1 respectively.

This uncertain nature of quantum systems along with principles like quantum entanglement, where two qubits even if separated by large distances have an effect on the state of the other, is the reason behind the parallel computing power of quantum computers. These unique properties allow for quantum algorithms to be developed which are exponentially faster than classical computers on computationally complex problems and the aim of quantum computing is to apply these algorithms to problems like NLP that are resource-costly for classical computers to work out on their own. We will be exploring the different aspects of QNLP on near-term quantum computers, hybrid quantum-classical hardware approaches and discuss important libraries for QNLP implementation.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue I Jan 2022- Available at www.ijraset.com

II. LITERATURE REVIEW

A. QNLP on Near-term Quantum Computers

Quantum computers, unlike classical computers, are very prone to errors and need to be extremely precise. The overhead of keeping continuous quantum states and manipulating precisely varying parameters makes it so that quantum computers with a large number of qubits have exponentially high error rates. This is one of the biggest hurdles in quantum computing and has caused the current generation of quantum computers to be limited to a small number of qubits. The term "Noisy Intermediate-Scale Quantum (NISQ)"[1] was introduced to represent the current era of quantum computers.The source of noise in computing can be anything from dirt on the hardware to heat or noise in electronics. But for quantum computing to be of any use we need devices that can correct these errors very precisely. For maintaining the precision required to be able to use algorithms like Shor's and Grover's search algorithm, we need quantum computers that can manage the erroneous states of all the qubits effectively. It is generally accepted that systems with more than 50 qubits are extremely complex due to the enormous overhead of quantum states and thus cannot be effectively simulated on classical computers. These devices can perform computations that would be infeasible for a classical computer. Thus, the NISQ-era is made up of devices that are not fault-tolerant because there is not enough qubits to error-correct (thus "noisy"), but there is enough for it to do things classical computers cannot (thus "intermediate-scale").

One of the most important aspects of NLP is to find the meanings of words and figure out if we can compute the meaning of a sentence that is made up of these words. To overcome this problem of extending the meanings and sentiments of individual words to phrases and sentences, representation of sentence meanings is required which would be independent of the grammatical structure.This kind of structure was first introduced as the DisCoCat (Categorical Compositional Distributional) model[2]. This model has been successful in modeling meaning for the most part but fails to account for the fact that language can change i.e. the contextual nature of meaning is unhandled by this model. DisCoCat combines linguistic meaning and linguistic structure such as grammar into one structure. The model finds its origins in categorical quantum mechanics (CQM) formalism. Thus, it is natural to assume that it is perfectly suited for quantum hardware. DisCoCat and CQM, find their implementation on quantum hardware as forms of turning diagrammatic representations to quantum processes. The resulting quantum representation of the language is then morphed using ZX-calculus[3] into quantum circuits which can be implemented on quantum hardware. Accounting for the linguistic on classical hardware, in contrast, is exponentially expensive

NISQ devices are particularly suited for working on QNLP due to the nature of variational quantum circuits. These are quantum circuits with a number of variables. The choice of variational circuits in practice is made solely based on performance rather than theory; thus, they are referred to as "ansatz"; acknowledging that they are empirically motivated not conceptually. The differentiating factor between the implementation on classical and quantum hardware is that the variational circuits represent the language structure inherently while the encoding of the grammar for the same language on classical hardware is expensive. This quantum native structure of language allows for effective NLP pipelines to be designed for NISQ devices[4].

B. Hybrid approaches for Quantum Natural Language Processing

Natural Language Processing is a very broad field which includes Linguistics, Computer Science and AI (Artificial Intelligence). As we progress more and more devices are equipped with applications that interact with day to day life of users using NLP. The adoption of NLP is happening at a rapid pace and so is the need for newer models that can process much more data and are computationally faster and efficient. Quantum Computing arises as a probable solution for tackling this problem. Some challenges posed by current approaches include grammatical rules, unstructured data and the semantics.

Classical Machine Learning uses various techniques like deep neural networks to find statistically significant patterns in Data and can recognize most of the patterns well. These deep neural networks become quite computationally complex for larger problems and also can't produce certain patterns. Quantum computers use quantum mechanics to produce such patterns at a much lower computational cost. Quantum speedups are categorized by two measures from complexity theory. The gate complexity determines the number of gates for elementary operations required for a specific result & query complexity determines the number of queries to the information source. The quantum subroutines like solving linear equations exhibit exponential quantum speedups over the best of classical algorithms & might prove useful in various machines, Fourier transforms, learning tasks like linear algebra, principal component analysis to name a few and some others[5].

Development of a Hybrid model like "A hybrid workflow for representing small and large scale corpus data sets to be encoded, processed, and decoded using a quantum circuit model"[6] was created to solve the problem and lays a foundation for the same. The model represents meaning in quantum states by pre-processing the tag tokens with appropriate grammatical type. This is stored on to classical memory.

The way to enable encoding/decoding is to do one to one mapping between pre-processed data and the Quantum circuit. Mapping is done in such a way that it preserves the inter-token relationships. The model has used hamming distance for encoding. These tokens are given a weight and encoded in an equal weight superposition state. This model gives us a pathway for data to be encoded, decoded and processed using quantum circuits. The results show a promising workflow for natural language processing models to use the hybrid approach as noted in [6]. This can be used to further lower the computational work performed by classical computers. Another model proposed a quantum algorithm to be used for Compositional Natural Language Processing. The model[7], incorporates scaling of Quantum systems such that it addresses the computational complexities in tensor product based compositional semantics & CSC (DisCo) based sentence similarity algorithm which gives quadratic speed up under certain conditions.

C. Libraries for QNLP implementation

To speed up the development of real-world quantum natural language processing applications such as text mining, automated dialogue, language translation, text-to-speech and bioinformatics is the goal of the quantum language processing toolkits and libraries. The concept behind the approach is that a sentence consisting of a group of words can be represented as a quantum network or tensor network and then it can be converted to a quantum circuit. Currently the only available QNLP library is Lambeq developed by Cambridge Quantum(CQ). But there are Quantum Machine Learning libraries available like TensorFlow Quantum (TFQ) which can be implemented to design QNLP specific quantum circuits.

1) *Lambeq:* Lambeq is the first software library for quantum natural language processing that is used for transforming phrases to quantum circuits. The community of quantum computing has made Lambeq open-sourced to benefit the global community and it is quickly developing an ecosystem of developers, academics, research and users. The QNLP library Lambeq is smoothly compatible with Cambridge Quantum's TKET which is also open-source and used extensively as a software development platform. Because of this benefit QNLP developers have access to as many quantum computers as feasible. To integrate natural language into quantum circuits is not an easy task.The initial process of Lambeq is to process and parse a sentence. With the help of a statistical Combinatory Categorial Grammar(CCG) parser, a syntax tree is produced for selected compositional models. A string diagram is formed from a parse tree as the next process[8]. A string diagram shows the grammatical structure of a sentence in a more detailed manner. For storing and manipulating this string diagram as a grammatical structure a python library called as DisCoPy is used by QNLP library lambeq as a backend database. This is based on the DisCoCat model discussed in section - A. A rewrite rule is used for these string diagrams for the transformation by the application. The transformation applied to these string diagram is then converted into actual quantum circuits. The output via Cambridge Quantum Computer's TKET can be guided towards a quantum simulator or a quantum computer, while in the classical case, the network(tensor network) for optimization can be passed to an ML library such as Jax or Pytorch.

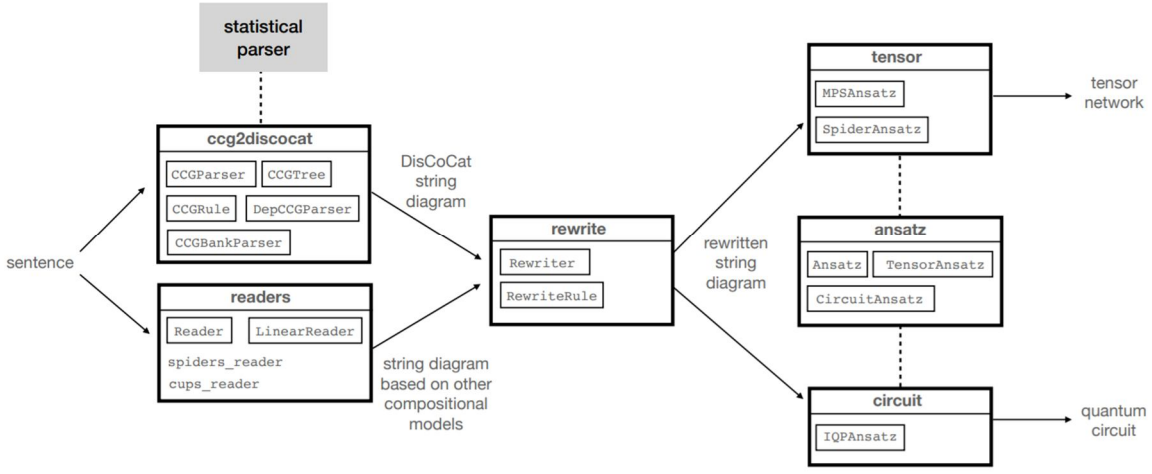


Fig. 1. lambeq's architecture and main modules from : Kartsaklis, Dimitri & Fan, Ian & Yeung, Richie & Pearson, Anna & Lorenz, Robin & Toumi, Alexis & Felice, Giovanni & Meichanetzidis, Konstantinos & Clark, Stephen & Coecke, Bob. (2021). *lambeq: An Efficient High-Level Python Library for Quantum NLP*.

2) *Tensorflow Quantum:* Another framework that uses the NISQ(Noisy Intermediate Scale Quantum) Processors is TensorFlow Quantum (TFQ). It is open source and used for development of hybrid quantum-classical ML models. It focuses on quantum data and integrates with quantum algorithms, tensorflow APIs and on quantum simulators as well. Newer algorithms are introduced like QNN(Quantum Neural Networks) & PQM(Parameterized Quantum Circuits). TFQ can unify classical and quantum ML infrastructure because fundamentally speaking circuits are tensors and using Cirq Constructs we can generate these tensors. So, we can seamlessly convert quantum circuits to tensors that are compatible with TensorFlow data structures. TFQ aims at bridging the two fields : Machine Learning & Quantum Computing. Cirq is a framework that enables us to work on Quantum Circuits for near term devices & also helps in making algorithms for NISQ machines. As noted in [9], TFQ coupled with Qsim and sometimes with GPU is a lot faster at computation than Cirq.

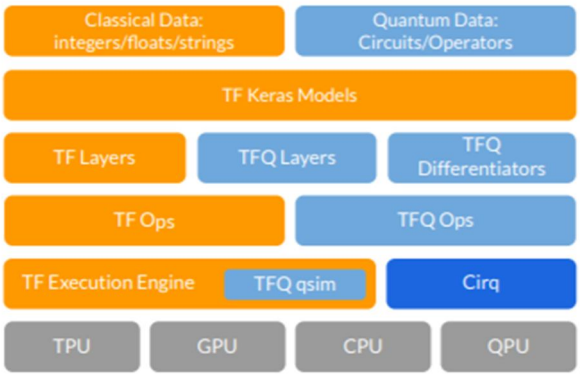


Fig. 2. The software stack of TFQ, showing its interactions with TensorFlow, Cirq, and computational hardware from: Broughton, Michael & Verdon, Guillaume & McCourt, Trevor & Martinez, Antonio & Yoo, Jae & Isakov, Sergei & Massey, Philip & Niu, Yuezhen & Halavati, Ramin & Peters, Evan & Leib, Martin & Skolik, Andrea & Streif, Michael & Von Dollen, David & Mcclean, Jarrod & Boixo, Sergio & Bacon, Dave & Ho, Alan & Neven, Hartmut & Mohseni, M.. (2020). *TensorFlow Quantum: A Software Framework for Quantum Machine Learning*.

III. A GENERAL PIPELINE FOR QNLP IMPLEMENTATION

Having looked at the approaches to QNLP and available libraries to implement the required algorithms; let us now establish a general pipeline for QNLP systems. This kind of pipeline will provide a flow from a sentence to a quantum circuit to a label in a model. The pipeline discussed here shall have 3 major steps - parsing, DisCoCat implementation and quantum circuit interpretation.

1) *Parsing:* This is where the sentence is processed to get a syntax tree as the output which can be used for the implementation of the DisCoCat model in the next step. This step is non-optional when you are dealing with large datasets with thousands of sentences; thus, requiring a pre-processing using a parser that can make groups and create a syntax tree based on the different structural aspects of the language i.e, parts of speech. If, however, the dataset is small enough, this step can be executed somewhat semi-automatically as there will be a small number of grammatical structures to account for. Referring[13, 14] we can cover the prerequisite concepts in detail but for the sake of covering everything, let us look at the basics of the concepts required to understand the working of this pipeline. Languages in general have a hierarchical structure containing the following components: sentences, phrases, clauses and words. Syntax refers to a set of rules that determine the structure of sentences in a language. You can think of it as a definition of the sequence of nouns, verbs, etc. (parts of speech) which must be followed to make up sentences. Referencing [14], a syntax tree or parse tree is nothing but the graphical representation of the renormalization of the (linguistic) information of a sentence. You can think of it as a representation of the syntactic categories of a sentence.
2) *DisCoCat Implementation:* After the generation of the syntax tree, on each sentence of the dataset Categorical Compositional Distributional model[2], also called DisCoCat model, is used. An important thing to note here is that the DisCoCat model is syntax-sensitive which is the reason that the first step of determining the syntactic structure of the dataset is important. The application of this model on each sentence can be briefly understood as working in three steps - creation of DisCoCat diagrams, rewriting and Ansatz. Let us understand each of these sub-steps.

a) Creation of DisCoCat Diagram: A simple way to visualize the generation of DisCoCat diagram is to represent each word as a state and then connect them with wires that represent each of the reduction rules (expresses a semantic equivalence between two expressions). For example, let us consider the statement "Bob enjoys good sentences". The DisCoCat diagram for this would be something like this:

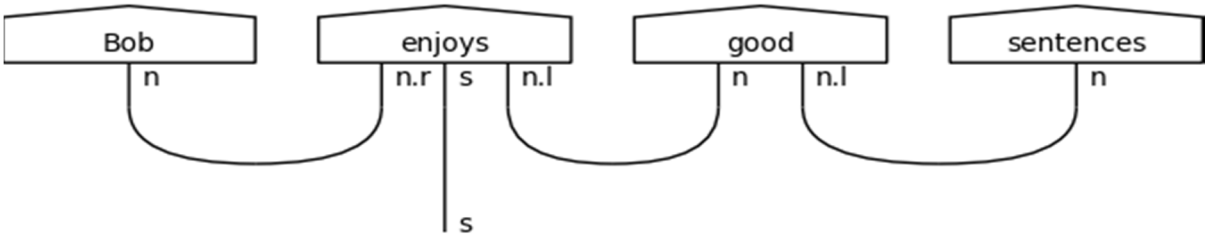


Fig.3: DisCoCat diagram of "Bob enjoys good sentences"

b) Rewriting: This is the sub-step where the DisCoCat diagram is used as a reference to get the reductions possible in the sentence. The transformation of a diagram like the one given above, makes the structure more compact and provides a computational advantage. The rewriting explained here is a simplified version of the bigraph method[4]. A common way to visualize this is to "bend down" the nouns of the sentences as can be seen below:

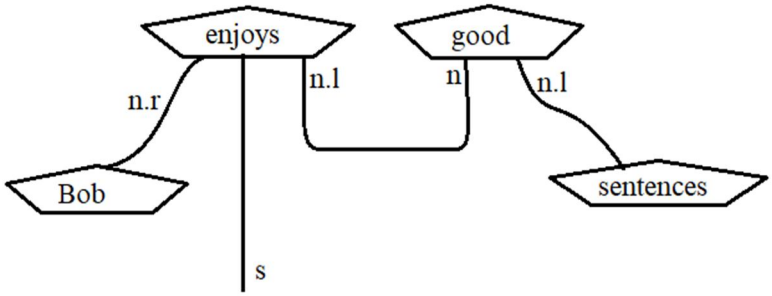


Fig. 4: Reduced diagram after rewriting(optimizing) DisCoCat diagram

c) Ansatz: This is the final step of the DisCoCat model where the DisCoCat representation of the sentences created by the previous steps is transformed into a quantum circuit[4]. This transformation is done by mapping the DisCoCat diagram in the following manner - (i) qn and qs are chosen as the number of qubits for every wire of type n and s and their dual types to get mapped to and (ii) selecting concrete parameterized quantum states for the word states to get replaced by. The choice of Ansatz thus determines the number of parameters for each word's representation and the connectivity of the circuit is fixed by the syntax's structure.

3) Quantum Circuit Interpretation: This is the final step where the quantum circuit is converted to a series of mathematical operations in accordance with the available gates in the system and these operations are then carried out on the quantum hardware. For this, the quantum circuit is first translated to machine instructions by a quantum compiler; this is done by taking into account the mathematical interpretation of the circuit as well as the available set of gates as hardware and the topology of the system. These machine-specific instructions are then passed on to a quantum computer to execute the quantum circuit a number of times. The final step of the process is to get estimations of the relative frequencies to determine the final result. A detailed explanation of this step is beyond the scope of this paper[16].

IV. FUTURE SCOPE

Considering the limitations of classical computers, quantum devices will soon have commercial purposes like developing complex conversational systems using QNLP techniques which will outperform the fastest classical computers by leaps and bounds. The quantum-native QNLP algorithms designed for inherent processing of language structures will provide exponential speedup in NLP model training when implemented on quantum hardware. Future work in QNLP will take into consideration the topology of qubit arrangement in the system. The current theoretical work has been done under the assumption of linear arrangement of qubits. The topology consideration will minimize the number of crossings in the bigraph[4]; optimizing the process. As the theoretical work progresses in the field and experiments are carried out, optimized task-specific word circuits will be designed leading to a new area for optimization research as general ansatz will become too much of a trade-off compared to efficient task-specific ones.

Another big consideration is the effect of the scaling up of QNLP. It is interesting to note how scaling up in different dimensions can result in the scaling up of resource cost which can be expensive for the current generation of quantum devices. For example, as sentences get longer, the quantum circuit you have to design to define its structure grows in width depending on the number of qubits available. Longer sentences will end up costing exponential increase in the time required for qubits to be post-selected. These costs are, however, not impassable hurdles as sentences in languages are usually upper-bounded in length. However, larger vocabulary also means higher dimensional parameter space. This is another scaling issue which can be handled by closely examining different optimization methods and the cost function itself.

As quantum computers continue improving, future work may be able to do comparative analyses with approaches that do not use a compositional model like DisCoCat, that hardwired grammar, and may be able to design experiments demonstrating a possible quantum advantage for NLP tasks over traditional systems.

V. CONCLUSIONS

In this paper, we studied QNLP, NISQ devices and the implementation of QNLP on near-term quantum computers. Then we looked at hybrid classical-quantum approaches followed by the libraries available for QNLP implementation. The quantum-native nature of QNLP leads us to conclude that the combination of meaning and language structure into a single model is a quantum model and thus working on it on quantum hardware is in fact a more natural choice than classical hardware. Another important note is that NISQ devices are particularly well-suited for QNLP due to which when classical data is encoded using variational quantum circuits, the linguistic structure is automatically accounted for unlike on classical hardware where accounting for linguistic structure is resource-costly. QNLP in quantum systems is also completely understood and exposed unlike modern classical machine learning techniques which are black boxes where the humans do not know the intricacies of the language structure inside the model. It is also important to note that QNLP, when scaled up in different dimensions, scales up in resource-cost and thus, to be able to utilize a quantum advantage on the current generation of modest quantum devices, we can turn to classical-quantum hybrid systems. These systems as we saw provide an exponential speed up in the areas which act as bottlenecks for classical systems by outsourcing the most computationally expensive parts of the processing to quantum hardware.

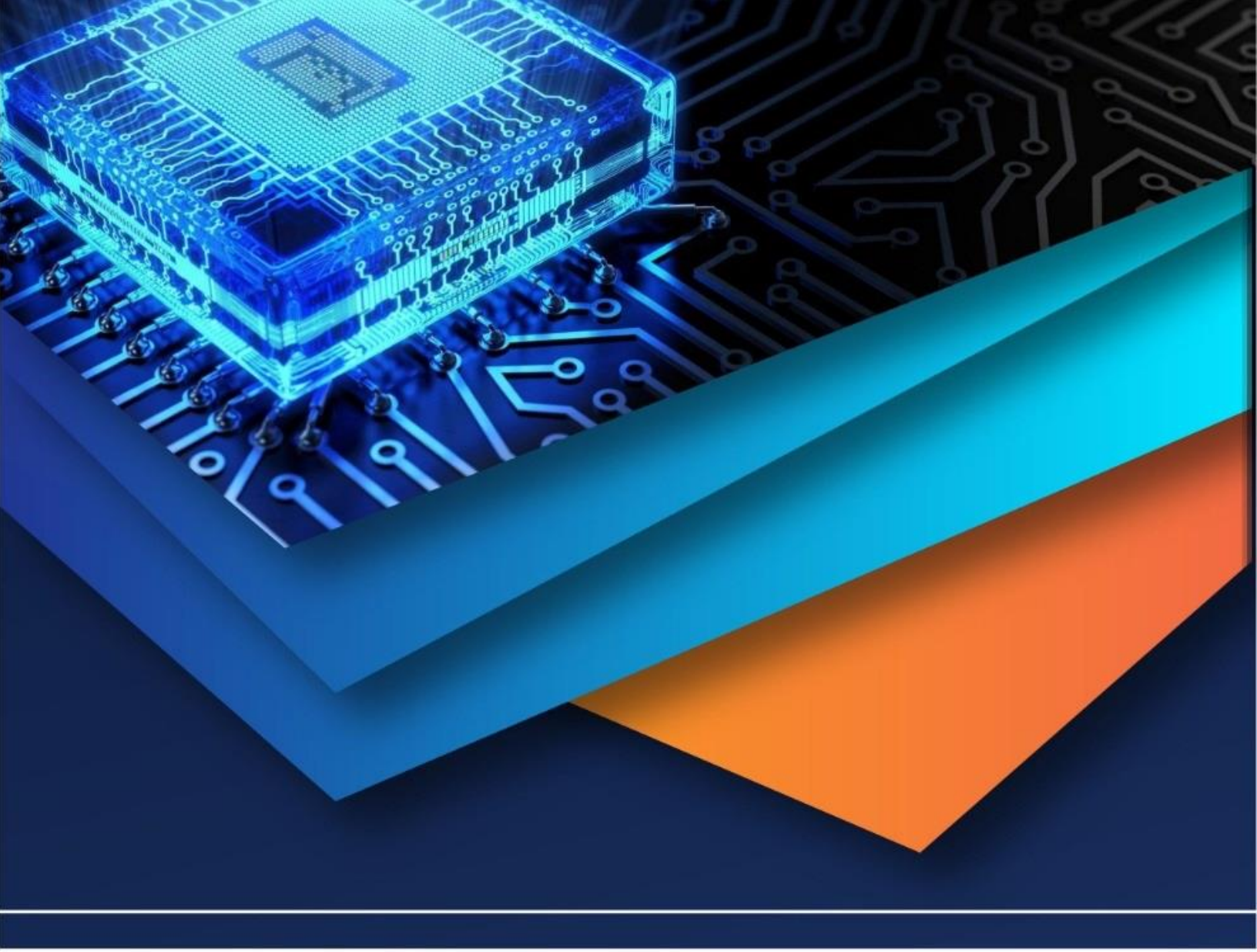
VI. ACKNOWLEDGMENT

We would like to express our gratitude towards everyone that provided us with any help with the research for this paper; without their help and support, it would not have been completed on time. We are also grateful to our project guide, Assistant Prof. Ahlam Ansari, who provided us with the much needed guidance.

REFERENCES

[1] Preskill, J. (2018). Quantum computing in the NISQ era and beyond. Quantum, 2, 79.
[2] Coecke, B., Sadrzadeh, M., & Clark, S. (2010). Mathematical foundations for a compositional distributional model of meaning. arXiv preprint arXiv:1003.4394.
[3] Coecke, B., & Duncan, R. (2011). Interacting quantum observables: categorical algebra and diagrammatics. New Journal of Physics, 13(4), 043016.
[4] Meichanetzidis, K., Gogioso, S., De Felice, G., Chiappori, N., Toumi, A., & Coecke, B. (2020). Quantum natural language processing on near-term quantum computers. arXiv preprint arXiv:2005.04147.
[5] Biamonte, J.D., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. Nature, 549, 195-202.
[6] O'Riordan, L.J., Doyle, M., Baruffa, F., & Kannan, V. (2021). A hybrid classical-quantum workflow for natural language processing. Mach. Learn. Sci. Technol., 2, 15011.
[7] Zeng, W.J., & Coecke, B. (2016). Quantum Algorithms for Compositional Natural Language Processing. SLPCS@QPL.
[8] Kartsaklis, D., Fan, I., Yeung, R., Pearson, A.N., Lorenz, R., Toumi, A., Felice, G.D., Meichanetzidis, K., Clark, S., & Coecke, B. (2021). lambeq: An Efficient High-Level Python Library for Quantum NLP. ArXiv, abs/2110.04236.

[9] Broughton, M., Verdon, G., McCourt, T., Martinez, A.J., Yoo, J.H., Isakov, S.V., Massey, P., Niu, M.Y., Halavati, R., Peters, E., Leib, M., Skolik, A., Streif, M., Dollen, D.V., McClean, J.R., Boixo, S., Bacon, D., Ho, A.K., Neven, H., & Mohseni, M. (2020). TensorFlow Quantum: A Software Framework for Quantum Machine Learning. ArXiv, abs/2003.02989.
[10] Lorenz, R., Pearson, A.N., Meichanetzidis, K., Kartsaklis, D., & Coecke, B. (2021). QNLP in Practice: Running Compositional Models of Meaning on a Quantum Computer. ArXiv, abs/2102.12846.
[11] Coecke, B., Felice, G.D., Meichanetzidis, K., & Toumi, A. (2020). Foundations for Near-Term Quantum Natural Language Processing. ArXiv, abs/2012.03755.
[12] Meichanetzidis, K., Toumi, A., Felice, G.D., & Coecke, B. (2020). Grammar-Aware Question-Answering on Quantum Computers. ArXiv, abs/2012.03756.
[13] Liddy, E.D. 2001. Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc.
[14] Khurana, Diksha & Koli, Aditya & Khatter, Kiran & Singh, Sukhdev. (2017). Natural Language Processing: State of The Art, Current Trends and Challenges.
[15] Gallego, Á.J., & Orús, R. (2017). The physical structure of grammatical correlations: equivalences, formalizations and consequences. ArXiv, abs/1708.01525.
[16] Michael A. Nielsen & Isaac L. Chuang. (2010). Quantum Computation and Quantum Information (10th ed.) (pp. 177 - 204). CAMBRIDGE UNIVERSITY PRESS.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY