IEEE Access
Multidisciplinary Rapid Review Open Access Journal

Received 9 June 2024, accepted 24 June 2024, date of publication 28 June 2024, date of current version 26 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3420707

RESEARCH ARTICLE

Quantum Natural Language Processing: A Comprehensive Survey

CHARLES M. VARMANTCHAONALA[1], JEAN LOUIS K. E. FENDJI[2,3], JULIUS SCHÖNING[4], (Senior Member, IEEE), AND MARCELLIN ATEMKENG[5]

[1]Department of Mathematics and Computer Science, Faculty of Science, University of Ngaoundéré, Ngaoundere 28359, Cameroon
[2]Department of Computer Engineering, University Institute of Technology, University of Ngaoundéré, Ngaoundere 28359, Cameroon
[3]Stellenbosch Institute for Advanced Study (STIAS), Wallenberg Research Centre at Stellenbosch University, Stellenbosch 7602, South Africa
[4]Faculty of Engineering and Computer Science, Osnabrück University of Applied Sciences, 49076 Osnabrück, Germany
[5]Department of Mathematics, Rhodes University, Grahamstown 6140, South Africa

Corresponding author: Jean Louis K. E. Fendji (lfendji@gmail.com)

ABSTRACT By using the quantum mechanics phenomenon, quantum computers provide a new dimension of computational power that drastically accelerates solving complex and resource-intensive problems. One of the most evolving but, due to the computation power of standard computers, reasonably limited application domain is natural language processing (NLP). NLP seeks to give interactive systems the ability to understand and manipulate human language. Making systems understand and manipulate human languages requires large amounts of data and computational power during learning as well as during the execution of NLP. For handling these amounts of data like text and audio recordings and the complexity of classical NLP algorithms, quantum computation has emerged as a promising solution. This work gives an extensive overview of this new field, known as quantum natural language processing (QNLP). Introducing the basics of quantum computing, we discuss its use in NLP by explaining the different proposed embedding models, quantum algorithms, and other methods of QNLP. As QNLP is still in its infancy, this comprehensive overview is the foundation that points to the upcoming research direction.

INDEX TERMS Natural language processing (NLP), quantum algorithms, quantum computing, quantum gates, qubits.

NOMENCLATURE

BQP	Bounded-error Quantum Polynomial.
cTPR	contextual Tensor Product Representation.
DisCoCat	Distributional Compositional Categorical.
DisCoPy	Distributional Compositional Python.
FRM	Fock Representation Model.
GSC	Gradient Symbolic Computation.
LLM	Large Language Model.
MLM	Multi Language Model.
NISQ	Nosy Intermediate-Scale Quantum computers.
NLP	Natural Language Processing.
pTPR	positional Tensor Product Representation.
QBW	Quauntum-Bag-of-Word.
QC	Quantum Computing.
QLLM	Quantum Large Language Modeling.
QML	Quantum Modeling Language.
QNLP	Quantum Natural Language Processing.
QRAM	Quantum RAM.
TPR	Tensor Product Representation.
W2K	Word2Ket model.
W2KXS	Word2KetXS model.

The associate editor coordinating the review of this manuscript and approving it for publication was Maria Chiara Caschera.

I. INTRODUCTION

Phenomena that highlight the limitations of the classical mechanics model have always been present in our surroundings, yet they have remained elusive due to the constraints of the human mind. Until now, classical computing based on logical gates built from relays and transistors has reigned as the predominant model for simulating the dynamics of the universe, simplifying everything into different number spaces: binary, octal, hexadecimal, and even continuous variation in analog computers. The computation in these number spaces shaped our fundamental understanding of modeling the world with algorithms. However, these number spaces cannot accurately model specific objects, states, and phenomena within the universe. These include, among others, light that exhibits both particle-like and wave-like behavior, the intricate dynamics of money in economics and finance [1], and even the complexity of language behavior [2]. These enigmatic phenomena prompted the emergence of another and more comprehensive model for understanding the workings of the universe: quantum mechanics. This model harnesses the inherent properties of quantum systems in the universe, such as superposition and entanglement of states, to accelerate computations. It grants the ability to perform a single operation on multiple interconnected states simultaneously, revolutionizing our approach to problem-solving.

Natural language processing (NLP) is one of the biggest challenges despite today's computing power, as it involves several tasks simultaneously. These tasks encompass calculating the grammatical meaning of a sentence checking syntax and semantics. Within NLP, interactive systems must be capable of acquiring natural language, understanding contexts, answering questions, managing dialogues, recognizing speech as well as speaker, generating natural language, segmenting speech, extracting terminology, transferring text-to-speech, creating word segmentation – tokenization, tagging parts-of-speech, performing named entity recognition (NER), extracting relationship, discoursing analysis, performing argument mining, translating context, and performing many other tasks.

Significant breakthroughs in human-like NLP can be traced back to 2018 [3], [4], [5], even though the theoretical foundation dates back to the 1900s with the works of Georgetown [6] in 1950 and Joseph Weizenbaum on the ELIZA project in the 1960s, among many others. However, these earlier works were limited as the systems in these frameworks were programmed with explicit knowledge or supervised learning [7]. Since 2018, there has been ongoing work on more real-world NLP systems.

Contrary to the programmed behaviors of earlier NLP approaches, recent works are termed general-purpose large language models (LLMs) [8]. General-purpose LLMs enable computers to learn natural languages using sources like audio, videos, articles, books, and the Internet. However, this approach demands intensive resources.

On 11 June 2018, OpenAI researchers and engineers designed an NLP model called ChatGPT, and since then, new versions have evolved to reach GPT-5 with better capabilities [9], [10]. Several other giants have followed the step in the design LLM-based NLP systems, such as Google with BARD, Deepmind with Gopher, or Baidu with Ernie 3.0, to name a few. However, the costs of training and learning NLP systems are complex to maintain today on a large scale with current methods. For example, GPT-3 has been trained on over 510 trillion word tokens. In 2020, Lambdalabs [11] estimated the cost to train GPT-3 at 4.6 million US dollars and 355 GPU-years.

Despite their advanced capabilities, NLP systems still face limitations in task performance and learning ability. They struggle to consider the broader context in which a word appears and its multiple meanings, leading to lower accuracy and reliability. For example, GPT-3-based systems exhibit limitations such as providing incorrect or nonsensical answers [12], 'hallucination' [13], limited knowledge [14], and issues related to data justice [15], [16]. There is substantial work ahead to develop efficient and reliable NLP systems. These challenges and limitations primarily stem from the current architectures and methods employed. They are constrained in effectively learning patterns in natural languages, and more critically, these methods struggle to process massive datasets on the order of trillions. The imperative for new and more efficient approaches, such as quantum computing, is crucial for the future of NLP [17].

The previous paragraphs highlight two primary challenges persisting in classical NLP, which employs non-quantum processors: extensive computation time and substantial storage space demands. A potential and promising remedy lies in the adoption of quantum methods. These methods hold the potential to address both issues. Quantum approaches leverage the phenomena of superposition and entanglement of states to store multiple overlapping pieces of information in a single register. Additionally, the inherent parallelism of quantum operators facilitates a reduction in computing time, as elaborated in the subsequent sections.

II. MOTIVATION AND POSITION WITH OTHER SURVEYS

Firstly, there are only a limited number of surveys on the topic, specifically [18], [19], [20], given that quantum natural language processing (QNLP) is still in its early stages, and its parameters have not been thoroughly addressed. Secondly, most current surveys concentrate on particular models, such as the distributional compositional categorical (DisCoCat) model. While this model proves effective in specific linguistic tasks, it also exhibits limitations in other aspects. Therefore, exploring other existing models, including those that have not been adequately investigated, is crucial to establishing a more comprehensive foundation for discussing the appropriateness of QNLP models and charting a path forward.

III. METHODOLOGY

Throughout this study, the methodology navigated three key stages: paper collection, filtering, and analysis of the selected literature.

A. COLLECTION

This survey utilized reputable sources for disseminating scientific information, including but not limited to Science Direct, IEEE Explore, SpringerLink, Scopus, Web of Science, and various web search engines. The exploration across these diverse information repositories employed search terminologies such as 'Quantum computing' AND 'Natural Language Processing', 'Quantum Natural Language Processing' and 'Quantum system learning for Natural Language Processing.'

B. FILTERING

At this stage, each paper was comprehensively reviewed, encompassing its title, abstract, and conclusion, to assess its relevance to the theme and objective of this survey. Given the nascent stage of QNLP, the current pool of available papers on the subject is limited to a few dozen. Following the meticulous selection of papers pertinent to the focus of this review, a total of 122 sources, including papers, articles, documents, books, reports, and theses, were identified for further investigation in the subsequent phases of this study.

C. ANALYZING

In this phase, each paper underwent a comprehensive reevaluation, from the introduction to the conclusion, to extract and categorize various elements of interest. Subsequently, works related to QNLP were organized based on distinct features, including categorization by model, publication year, and application. Throughout this process, the survey seizes the opportunity to document the toolkits and frameworks employed in QNLP research as reported in the experiments that were conducted. This analysis provided insights into the current landscape of QNLP and facilitated a forward-looking perspective, identifying potential avenues for future research that could enhance the practical application of this technology within the broader scientific community of NLP.

IV. CONTRIBUTIONS

The contributions of this paper below are:

- Description of the advantages of applying quantum methods in NLP;
- Description of various quantum modeling languages;
- Simple, practical case studies and examples helping to understand the concepts and the survey;
- Evaluation of the advantages and disadvantages of different QNLP-based models;
- The generalisation of models to certain cases not covered by the state of the art;
- Summarization of various quantum algorithms applied to these models and their practical implementation;
- A comparative study of the different models and a summary of the results based on the common metrics of the experiments;
- Examination of challenges associated with QNLP and presentation of future research directions.

The subsequent sections of the paper are structured as follows. The paper commences with an introduction to fundamental concepts explicitly tailored for comprehension in Section V. In Section VI, QNLP is introduced, followed by the presentation of different models in Section VII. To provide a brief overview of current trends in QNLP, related works are summarized in Section VIII. Section IX lists valuable tools for the QNLP community and Section X gives an overview of the steps involved in implementing the proposed quantum models. This section also gives a summary of the datasets, metrics and results of the various experiments carried out. Thus, Sections XI and XII delve into QNLP challenges and potential future research directions, respectively. Section XIII concludes this paper with a summary.

V. BACKGROUND AND FUNDAMENTALS

This section provides an overview of the fundamental concepts necessary for understanding this survey. Following the explanation of technical concepts, comparisons and analogies are drawn between classical concepts and the innovative methods of quantum computation.

A. NATURAL LANGUAGE PROCESSING

While lacking a universally agreed-upon definition, NLP remains an evolving field with numerous technological applications. Broadly, NLP can be defined as the process of endowing machines with the ability to understand and utilize human language. By "natural language", we refer to any language used by humans for communication, whether in textual or audio format. To imbue machines with this human-like language processing capability, various methods have been devised, including:

- Rule-based Methods: These approaches involve a thorough analysis of the structure of natural language, followed by the formulation of algorithms based on syntactic, semantic, and grammatical rules, as well as operations such as tokenization. However, these methods often struggle with the intricacies and variations present in natural languages [21], [22], [23].
- Machine Learning (ML) Methods: In contrast to rule-based techniques, ML methods entail training supervised and unsupervised learning algorithms [24], [25], [26], [27]. Nonetheless, the efficacy of ML methods heavily relies on the availability and quality of data, which can be limited, particularly for low-resource languages, and can be time-consuming during training.
- Cross-lingual Methods: These models leverage existing NLP resources and models from high-resource languages such as German, English, and French to address the challenges of training on new or low-resource languages through knowledge transfer mechanisms. However, the success of cross-lingual methods hinges on the availability of parallel data or similar languages [28], [29], [30], [31].
- Bootstrapping Methods: These models incrementally enhance NLP capabilities by gradually incorporating additional resources, thereby mitigating the initial scarcity of resources [32], [33], [34], [35]. However, in multilingual models, due to the diversity of languages, they necessitate fast algorithms like quantum algorithms to continually update the model's general knowledge.

Across these NLP methods, certain stages such as preprocessing and encoding/decoding are indispensable. Preprocessing involves refining the data to yield high-quality results, while encoding/decoding entails providing a digital representation of natural language corpora, comprehensible to machines. Techniques like tokenization; to segment the corpus into manageable units, stemming; to reduce words to their roots, lemmatization; to identify canonical forms of words, and stopwords removal; to eliminate irrelevant words, are commonly employed to enhance processing and model performance.

B. QUANTUM COMPUTATION

In quantum computing, the computational universe, or space, is the Hilbert space [36]. Within this space, fundamental entities, such as qubits used for calculations, are represented as vectors. Specifically, these vectors belong to the set of complex numbers $\mathbb{C}^n$, and quantum operations are functions or operators capable of executing linear computations on these entities. For clarity and simplicity, it is worth noting that n is commonly equal to 2 in quantum computing. In contrast to the conventional vector notation in linear algebra, quantum mechanics employs a unique vector notation, denoted as $|\psi\rangle$ rather than $\vec{\psi}$.

Definition 1: The smallest unit of information in quantum computation, like the bit in classical computation, is called a qubit. A qubit is a vector $|\psi\rangle \in \mathbb{C}^2$ of the form: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, under the condition $\alpha^2 + \beta^2 = 1$, with $|0\rangle$ and $|1\rangle$ the basis vectors of the vector space $\mathbb{C}^2$ [36].

In general, the orthonormal basis $\{|0\rangle, |1\rangle\}$ of the Hilbert space can be expressed in the column vectors form, as

$$\{|0\rangle, |1\rangle\} = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}. \tag{1}$$

Quantum mechanics, distinct from classical mechanics, introduces an intermediate state between the classical true (1) or false (0) states. Consequently, a qubit is expressed as a linear combination of vectors or ground states $|0\rangle$ and $|1\rangle$, utilizing complex numbers α and β to represent amplitudes or weights that convey the significance or degree of presence of the corresponding state.

Upon measurement of a qubit, it collapses, yielding either the state $|0\rangle$ or $|1\rangle$, with probabilities of α^2 and β^2, respectively. The problem of decoherence in quantum systems is a notable challenge in quantum computing [37].

In classical computing, information, such as words or sentences, is not represented solely by a single bit but by a bit string—a sequence of 0's and 1's. The classical operation for constructing a bit string involves the concatenation operator. Similarly, in quantum computing, information is conveyed not only by a qubit but by a sequence of qubits or qubit strings, employing a formalism of *togetherness* [38]. The operation forming a qubit string is realized through the tensor product, allowing multiple qubits to combine and form an extensive quantum system.

Definition 2: Let V and W be two Hilbert spaces of dimension m and n respectively. The tensor product $V \bigotimes W$, read V tensor W, is a new vector space of dimension mn where an element is a linear combination $|v\rangle \otimes |w\rangle$, of two vectors $|v\rangle$ and $|w\rangle$, belonging to V and W respectively [36].

The tensor product $\bigotimes_{j=1}^{n} V_j = V_1 \otimes V_2 \otimes \cdots \otimes V_n$ is said to be of order n and the rank r of a simple tensor $v = \bigotimes_{j=1}^{n} v_j$, $v_j \in V_j$ is the smallest number of simple tensors that sum up to v.

1) SUPERPOSITION OF QUANTUM SATES

It follows from the previous definition that a quantum system with two qubits contains $2 \times 2 = 4$ classical information, and, in general, a quantum system with n qubits contains 2^n classical information in the same quantum register. Thus, a system of n qubits can represent up to 2^n different contexts of a word in QNLP, for example. A multi-qubit system with n qubits is represented by Equation (2).

$$\begin{bmatrix} \alpha_1| & \alpha_2| & \ldots & |\alpha_m \\ \beta_1| & \beta_2| & \ldots & |\beta_m \end{bmatrix} \tag{2}$$

According to Definition 1, for each qubit $|\alpha_i|^2 + |\beta_i|^2 = 1$, $i = 1, \cdots, m$.

Thus, for example, if there are three qubits in the system as given by:

$$\begin{bmatrix} \frac{1}{\sqrt{2}}| & \frac{1}{\sqrt{2}|} & |\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}}| & \frac{1}{\sqrt{2}|} & |\frac{1}{\sqrt{2}} \end{bmatrix} \tag{3}$$

the final system can be represented as:

$$|\psi\rangle = \frac{1}{\sqrt{2^3}} \Big[|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle \Big]. \tag{4}$$

The probability of observing the final system, cf. Equation 4 resp. Equation 3, in states $|000\rangle$, $|001\rangle$, $|100\rangle$, $|101\rangle$, $|010\rangle$, $|011\rangle$, $|110\rangle$, and $|111\rangle$ is $1/8$, indicative of a uniform superposition. Consequently, a quantum state composed of n qubits concurrently represents 2^n values. Conversely, in traditional or classical data processing, a sequence of n bits can only represent one of the 2^n possibilities. Thus quantum computation has great ability to perform high parallelism and enables an exponential speed-up using quantum operators.

2) QUANTUM OPERATORS

Quantum operators, typically represented as unitary matrices, serve as the fundamental entities for expressing transformations on quantum states [36]. Among the most commonly used are quantum logic gates:

- **X gate:** This quantum logic gate, called bit-flip gate, is the equivalent of the classical logic gate NOT. Its action allows to flip the qubit on which it is applied,

$X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$. It is defined by

$$X = |1\rangle\langle 0| + |0\rangle\langle 1| = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (5)$$

- **H gate (Walsh-Hadamard gate):** It is a very used quantum gate which has no equivalent in classical computing. It allows to exploit this famous property of quantum physics, namely the superposition of states. It allows to make a uniform superposition of the basic states. $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Thus, applying the **H** gate three times to the $|0\rangle$ qubit, $\mathbf{H}^{\otimes 3}|0\rangle$, gives the quantum state of Equation (4). The matrix representation of the **H** gate is

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (6)$$

- **CNOT gate:** It's the quantum operator that allows operation executions with conditional tests, such as "if condition then operation". It is defined by the matrix

$$\mathbf{CNOT} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X = \begin{pmatrix} I & O \\ O & X \end{pmatrix}. \quad (7)$$

where I is the identity matrix. Therefore $\mathbf{CNOT}|x\rangle|y\rangle = |x\rangle|x \oplus y\rangle$. This quantum gate takes as input the control qubit(s) and the target qubit(s). If the control qubit(s) is/are at 1, the operator executes the **X** gate on the target qubit(s), otherwise nothing happens. For example, $\mathbf{CNOT}|00\rangle = |00\rangle$, $\mathbf{CNOT}|01\rangle = |01\rangle$, $\mathbf{CNOT}|10\rangle = |1\rangle X|0\rangle = |11\rangle$, $\mathbf{CNOT}|11\rangle = |1\rangle X|1\rangle = |10\rangle$. Intuitively, it is worth to note that the action of **X** gate of the **CNOT** operator can be replaced by any other quantum operator **U**, in order to obtain a **CU** gate.

Leveraging these quantum logic gates and various other quantum operators [36], models can be developed to tackle specific linguistic tasks. The quantum computation model has demonstrated universality [36], capable of performing all possible operations using a combination of quantum logic gates. The subsequent section provides an overview of the potential applications of quantum computation in NLP.

VI. QUANTUM NATURAL LANGUAGE PROCESSING

In this section, we delve into the integration of quantum properties into NLP and demonstrate how these properties can serve as assets and offer solutions to enhance the performance of NLP models.

The similarity between NLP and quantum logic was observed some decades ago [2], [39]. The idea of using quantum computation is linked to the fact that NLP is a complex and computationally expensive problem that cannot be efficiently solved on a classical computer unless P = BQP [40]. NLP-related tasks require an enormous amount of computation time. In general, they involve performing calculations in large data sets. The ability of humans to process natural language takes into account several factors that are difficult for systems to grasp. Among the factors taken into account by humans to process natural language, we can cite, e.g., the context in which we find ourselves, the syntax and semantics of sentences, and the language used. Each of these factors is subject to several other factors, such as time and speaker. Making interactive systems to process natural languages as humans is complex and challenging. It includes a modeling problem and a vital computing power requirement. Several approaches are possible. The first approach is to give a system an enormous amount of information to learn and understand the language and perform linguistic tasks [9], [10], [11]. In such an approach, the system would be like a child born without any prior knowledge of the language he will use. But he ends up learning and understanding over time. From the immense data and information that existed before he is born, he learns to take the patterns out of the language and interpret them intelligently. However, we also quickly understand what this means for the system, just as it did for the child regarding the required learning time [11]. The child needs years to do this. The child's first years are spent learning the basics, and then she/he has to go through an academic system to improve her/his knowledge. The simulation of this process when dealing with systems is known as system learning through neural networks or deep learning. The second approach, which aims to improve the first by combining the two, involves giving the system statistical computing capacity for specific tasks, such as syntax or grammar recognition, considering the word's frequencies, contexts, and positions. As pointed out earlier, NLP seems to be a challenging task. Practical NLP systems are still flawed and can only perform certain specific tasks [5], [6], [12]. The future of NLP on the scale of human capacity remains difficult unless new, more efficient methods are developed to reduce learning time. Applying the computational power of quantum mechanics seems to be a good candidate for this task. Therefore, quantum computing can be a game changer in how NLP has been tackled.

A. SUPERPOSITION OF QUANTUM STATES IN NLP

The superposition principle of quantum states can significantly benefit NLP. Once again, NLP involves simultaneous consideration of multiple pieces of information when interpreting languages, such as word identifiers, contexts, syntax, word frequency, distance between words, and dependencies. A quantum state represents a simultaneous superposition of multiple states or properties. Thus, encoding NLP model training data into a quantum state offers greater advantages. This quantum state has the potential to capture several useful pieces of information for NLP that classical methods cannot. For instance, the phrase *Hello the quantum world* can be represented as a superposition of its constituent words, as illustrated by Equation (8),

$$|\psi\rangle = \frac{1}{\sqrt{30}}|00\rangle + \frac{2}{\sqrt{30}}|01\rangle + \frac{3}{\sqrt{30}}|10\rangle + \frac{4}{\sqrt{30}}|11\rangle. \quad (8)$$

In this quantum state $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$ represent the words, *Hello*, *the*, *quantum*, and *world*, respectively. The amplitudes $\frac{1}{\sqrt{30}}$, $\frac{2}{\sqrt{30}}$, $\frac{3}{\sqrt{30}}$ and $\frac{4}{\sqrt{30}}$ can represent the distance between the different words and the probability of one word appearing after another. The amplitudes $\frac{2}{\sqrt{30}}$ and $\frac{3}{\sqrt{30}}$ can mean how much far the word *quantum* is from the word *Hello* than the word *the*, ($\frac{2}{\sqrt{30}} < \frac{3}{\sqrt{30}}$). It's important to note that this representation is both expressive and memory-efficient compared to conventional vector representations, which often require multiple large vectors for each word in a sentence.

B. QUANTUM OPERATORS IN NLP

Quantum operators serve as the fundamental building blocks for manipulating quantum states that reflect both training and inference data in NLP. Leveraging basic quantum operators like quantum logic gates, we can enhance NLP performance through parallelism, owing to the linearity inherent in quantum operators. Furthermore, these same quantum operators, such as the Hadamard gate (**H**), enable the encoding of data into quantum states. Moreover, these operators facilitate the transformation of quantum states into classical values through quantum measurement operators.

For instance, constructing the quantum state described in Equation (8) can be effectively achieved in practice by applying a sequence of Hadamard gates using the *Initialize*() module of the IBM Qiskit framework,[1] with parameters specifying the amplitudes and the required number of qubits. Numerous models that exploit these quantum principles and operators are currently being investigated, with some demonstrating greater efficiency compared to classical models.

VII. QUANTUM NATURAL LANGUAGE PROCESSING MODELS

Several quantum models have been proposed for NLP-related tasks. Their main differences lie in how the NLP data is embedded or encoded within the quantum formalism. These varied encoding methods are suitable only for specific tasks. Beyond task suitability, the efficiency of these models in terms of memory resources and computation time varies. While some are efficient, others are not. The commonality among these models lies in their utilization of tensor products to manifest the *togetherness* notion in NLP, albeit in different ways. Seven distinct models are identified after thoroughly analyzing the papers considered in this literature review.

These emerging models are still in their nascent stages and exhibit varying levels of development. Some are more developed than others for several reasons. Practicality, applicability to specific NLP tasks on Noisy Intermediate-Scale Quantum (NISQ) devices, theoretical exploration, and time-frames for implementation all contribute to the divergence in development stages [18], [19], [20]. Particular models with short-term applications are hybrids, combining classical and quantum methods to exploit today's computers. Conversely, other models requiring more effort or intended for long-term applications are still in their theoretical state. The availability of experimental frameworks and libraries has also influenced the development of some models. A visual representation of the development levels of each model is provided in Figure 1, organized into model families as presented in subsequent sections.

These models employ different methods and techniques, with some sharing common elements. Drawing upon elements such as tensor product usage, object representation architecture, matrix utilization, and the order of release, a chronological tree of the models is constructed in Figure 2. Relationships between models are based on shared elements, indicating commonalities rather than direct derivations. Subsequent sections will delve into the description of these various models, emphasizing their similarities and differences.

A. QUANTUM BAG-OF-WORDS MODEL

The inspiration for this model originates from the most straightforward attempt to address the challenges in NLP. It involves a straightforward transposition of classical mechanics into quantum mechanics, maintaining the same principles but leveraging quantum probability theory.

In the classical bag-of-words model, a document containing m words is represented by the unknown probability distribution $\vec{\theta}$ across various elements, as described by Equation (9).

$$D_m = \{w_i : i = 1, \cdots, m\}. \tag{9}$$

The quantum probability model represents a set of words in a sentence or document using a quantum probability distribution. The quantum bag-of-words model portrays a document consisting of m words as a superposition of quantum events with their dependencies as:

$$\begin{aligned} D_m(d) &= D_m(\{d_1, \cdots, d_m\}) \\ &= |d\rangle\langle d|, \ |d\rangle = \sum_{i=1}^{m} \delta_i |e_{w_i}\rangle \end{aligned} \tag{10}$$

where $d = \{d_1, \cdots, d_m\}$ denotes the dependencies between different words, $|e_{w_i}\rangle$ defines the projector corresponding to the word w_i, and the normalization coefficients δ_i satisfy the condition $\sum \delta_i^2 = 1$.

To illustrate the modeling process, let us consider the corpus $W = \{orange, fruit\}$. Utilizing one-hot encoding, we associate the vectors $|0\rangle$ and $|1\rangle$ with the words *orange* and *fruit*, respectively. The projectors for *orange* and *fruit* are respectively represented by:

$$\begin{aligned} \Pi_{orange} &= |0\rangle\langle 0| \\ &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} (1, 0) \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \end{aligned} \tag{11}$$

[1]https://www.ibm.com/quantum/qiskit

and

$$\begin{aligned}\Pi_{fruit} &= |1\rangle\langle 1| \\ &= \begin{pmatrix}0\\1\end{pmatrix}(0, 1) \\ &= \begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}\end{aligned} \tag{12}$$

A possible superposition of *orange* and *fruit*, materializing the dependency between these two words, can be represented by the state:

$$D_m(W) = \sqrt{\frac{1}{3}}|orange\rangle + \sqrt{\frac{2}{3}}|fruit\rangle \tag{13}$$

The density matrix of this superposition is:

$$\begin{aligned}D &= \begin{bmatrix}\sqrt{\frac{1}{3}}\\ \sqrt{\frac{2}{3}}\end{bmatrix}\left[\sqrt{\frac{1}{3}}, \sqrt{\frac{2}{3}}\right] \\ &= \begin{bmatrix}\frac{1}{3} & \frac{\sqrt{2}}{3}\\ \frac{\sqrt{2}}{3} & \frac{2}{3}\end{bmatrix}\end{aligned} \tag{14}$$

D_m is a superposition of words with amplitudes as the dependencies between words, and D is the density matrix containing all the correlations between the different words.

Similar to how a classical word or symbol is represented by a bit string, in this model, each word corresponds to a qubit or a qubit string, depending on the atomic components of the text corpora. This model requires many qubits to represent the words, excluding auxiliary qubits for calculations. Although this model does not explicitly consider natural language features like word position, order, context, combination, syntax, and semantics during the modeling process, it proves effective for various QLNP tasks, such as calculating word occurrences in the corpus or summing occurrences across different corpora. Quantum operations, e.g., maintaining the score of each word, can be implemented using a quantum rotation logic gate. For example, Figure 3 demonstrates how the sum of occurrences of a word in two different texts can be calculated through their angles. Various scoring and estimation functions are explored in [41]. This model is adaptable to system learning algorithms, including supervised learning, as demonstrated in its application to a classification problem [42], achieving 100% accuracy with a small dataset. While many NLP tasks can be accomplished with this basic model, its true advantage over classical computation remains to be fully explored, given the significant resources it requires, including the number of qubits and memory. In [42], efforts are made to enhance this model, particularly in corpus classification, addressing issues such as redundancy reduction and a more compact distribution of the model's encoding dimensions. The proposed models, namely 'One Qubit Per Embedding Dimension' and 'N Qubits for 2^N Embedding Dimensions', aim to address these improvements.

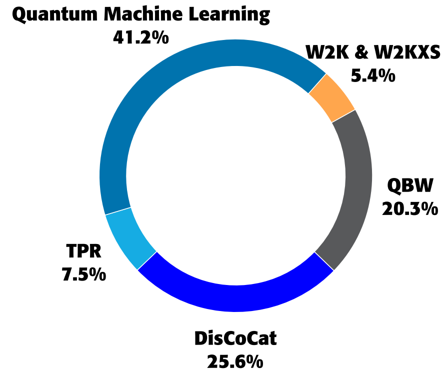

FIGURE 1. Percentage of adoption per QNLP models; based on the analyzed papers.

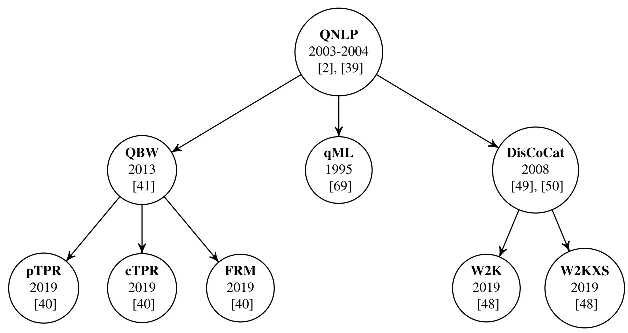

FIGURE 2. Chronological tree of QNLP models. qML stands for quantum machine learning.

B. TENSOR PRODUCT REPRESENTATION MODELS

This category of models, distinct from the previous ones, endeavors to explicitly consider natural language features such as word position and context during the word embedding process. Tensor product representation (TPR) models aim to portray a collection of intricate symbol structures as space vectors, drawing inspiration from gradient symbolic computation (GSC) principles [47], [48]. The objective is to encapsulate diverse features within a complex set of symbols and structure them within quantum systems. In the realm of QNLP, these features, which can be captured and encoded in quantum systems, include elements crucial for interpreting natural language, such as words, phrases, verbs, nouns, adverbs, and contexts. Within this category of QNLP models, three main types handle various NLP features: the positional tensor product representation (pTPR) model, the contextual tensor product representation (cTPR) model, and the fock-space representation model [40].

1) POSITIONAL TENSOR PRODUCT REPRESENTATION MODEL

The pTPR model organizes symbols based on their positions in a binary decision tree, with nodes being vectors representing language elements for processing. In pTPR, the left and right positional roles, materialized by the binary tree, represent different sentence elements. Like a sentence,

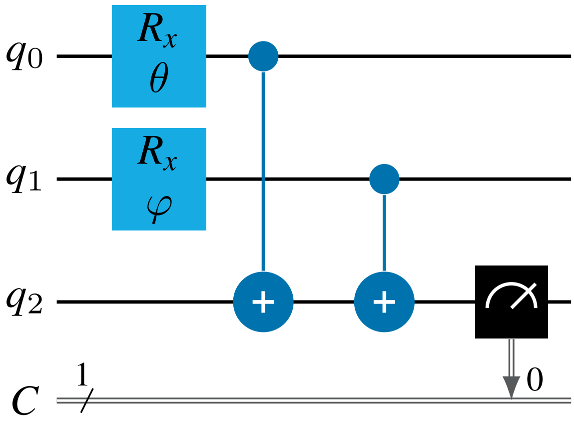

FIGURE 3. Quantum Adder Circuit that combines the angles θ and φ. This can be used to sum the scores of two quantum states.

the global vector representing a symbol structure is a superposition of vectors for the structure's basic constituents. Each constituent, located at a node in the phrase's binary parse tree, is associated with a vector derived from the tensor product of two basic vectors: one encoding the symbol s_i, and the other encoding its position n_i. Consequently, a complete sentence is represented by a high-dimensional vector S resulting from the tensor product of its various small-dimensional constituents, i.e., noun, verb, adverb, and others, as expressed in Equation (15).

$$S = \sum_i s_i \otimes n_i. \tag{15}$$

The symbol positions, n_i, can be encoded using qubits or alternative labels, with the left-hand position encoded as $|0\rangle$ and the right-hand position as $|1\rangle$. This allows the calculation for various elements of the binary tree, as illustrated in Figure 4, to be encoded (e.g., as $|011\rangle$) and interpreted as *the left ($|0\rangle$) child of the right ($|1\rangle$) child of the right child of the tree root*. A single child node can be randomly assigned a name of $|0\rangle$ or $|1\rangle$, and the position of the root of the sentence is considered as a standard symbol, such as $|\epsilon\rangle$. Therefore, the pTPR model for the sentence *Quantum Natural Language Processing* is represented by the Equation (16).

$$\begin{aligned}|\psi_S\rangle = |S\rangle|\epsilon\rangle + |AP\rangle|0\rangle + |Quantum\rangle|00\rangle + |AP\rangle|1\rangle \\ + |A\rangle|01\rangle + |Natural\rangle|001\rangle + |NP\rangle|11\rangle \\ + |N\rangle|011\rangle + |Language\rangle|0011\rangle + |N\rangle|111\rangle \\ + |Processing\rangle|0111\rangle\end{aligned} \tag{16}$$

and this can easily be visualized, the different positions, through its binary tree represented in Figure 4.

The vectors encoding symbols of basic constituents, denoted as $|N\rangle$, $|V\rangle$, $|A\rangle|NP\rangle$, $|AP\rangle$, belong to the dimension of the vector space of language symbol vectors. This dimension is relatively small compared to the dimension of the entire sentence space. The vector representing the whole sentence has a dimension corresponding to the tensor product of all basic vectors. If V_0 is the vector space of basic

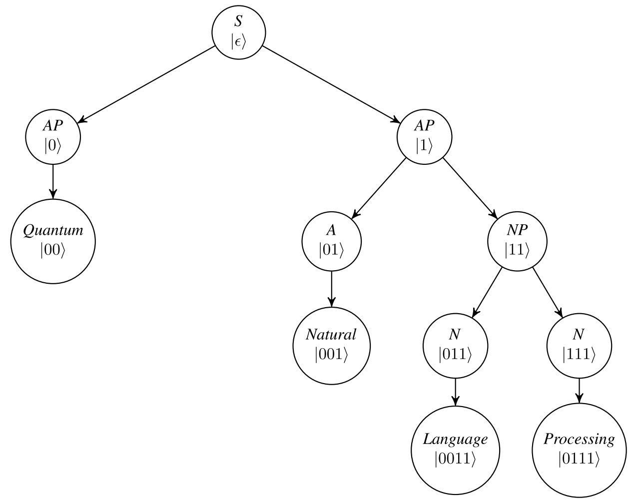

FIGURE 4. Parse tree for QNLP. S = sentence, N = noun, A = adjective, P = phrase. This representation captures fillers or symbols with their positions.

constituents, then the sentence space with n constituents is defined by the vector space given in Equation (17).

$$V_n = \bigoplus_{d=0}^{\infty} V_0^{\otimes d} \tag{17}$$

where d is the depth of a node in a tree. For example, $|0111\rangle \in V_0 \otimes V_0 \otimes V_0 \otimes V_0 = V_0^{\otimes 4}$ if V_0 is the vector space spanned by the basis of Equation (1).

The representation of a text corpus is a superposition of different sentence representations that compose it. It involves representing a text as a single tree comprising the superposition of various trees rather than considering them as distinct entities. In similar sentences with minor differences, the disparities are associated with weights only on the differing nodes, while the rest occupy identical nodes. For instance, if $|\psi_1\rangle =$ *Classical Natural Language Processing* and $|\psi_2\rangle =$ *Quantum Natural Language Processing* are two sentences, the corresponding pTPR model of the corpus formed by these two phrases, would be the superposition of the two sentences, expressed as:

$$|\psi\rangle = \frac{1}{2}(|\psi_1\rangle + |\psi_2\rangle). \tag{18}$$

This representation is derived from the binary tree of the symbolic structure, yielding $|\psi\rangle = \frac{1}{2}$*(Classical + Quantum) Natural Language Processing*.

2) CONTEXTUAL TENSOR PRODUCT REPRESENTATION MODEL

In this model, unlike pTPR model, language components or symbols are interpreted not by their positions in the corpus but through their context or environment. Instead of stating, 'the symbol s_i of sentence S is at position i', we express it as 'the symbol s_i of sentence S is preceded by the symbol s_{i-1} and followed by the symbol s_{i+1}'. This contextual tensor product representation (cTPR) model seems more effective in reducing ambiguity or redundancy in corpus representation. For example, in the cTPR model for the QNLP corpus, the word *natural* is presented as preceded by *quantum* and followed by *language*, represented by:

$$\begin{aligned}|\psi_S\rangle &= |Quantum\ Natural\ Language\rangle \\ &\quad + |Natural\ language\ Processing\rangle \\ &= |Quantum\rangle|natural\rangle|language\rangle \\ &\quad + |Natural\rangle|Language\rangle|Processing\rangle. \end{aligned} \tag{19}$$

In this example, spaces are not considered, and the context window size γ defining the word *natural*'s context is set to 2, although this window size can be adjusted.

We can observe that in this model, the dimension of the vector space encoding the sentence is very high compared to that of the pTPR model. To represent the word *natural* in the *Quantum natural language processing* corpus, we need a vector space of dimension v_o^4, at least, if v_o is the base space that extends or encodes base words such as $|natural\rangle$, $|language\rangle$, $|processing\rangle$. The cTPR, however, is free of ambiguity since the superposition of corpora preserves the unambiguous property or identity of the corpora. For example, if we have two corpora $|\psi_1\rangle = Classical\ Natural\ Language\ Processing$, and $|\psi_2\rangle = Quantum\ Information\ Processing\ Example$, the superposition of the two corpora using pTPR models is given by Equation (20).

In this model, the dimension of the vector space encoding the sentence is significantly higher than that of the pTPR model. To represent the word *natural* in the *Quantum Natural Language Processing* corpus, a vector space of dimension v_0^4 is required, at least, if v_0 represents the space encoding base words such as $|natural\rangle$, $|language\rangle$, $|processing\rangle$. The cTPR, however, remains free of ambiguity, as the superposition of corpora maintains their unambiguous properties or identities. For instance, if we have two corpora $|\psi_1\rangle = Classical\ Natural\ Language\ Processing$ and $|\psi_2\rangle = Quantum\ Information\ Processing\ Example$, the superposition of these corpora using pTPR models is given byEquation (20).

$$\begin{aligned}|\psi_S\rangle &= |Classical\ Natural\ Language\ Processing\rangle \\ &\quad + |Quantum\ Information\ Processing\ Example\rangle \\ &= |Classical\rangle|Natural\rangle|Language\rangle|Processing\rangle \\ &\quad + |Quantum\rangle|Information\rangle|Processing\rangle|Example\rangle \\ &= (|Classical\rangle + |Quantum\rangle)|1\rangle + (|Natural\rangle + \\ &\quad + |Information\rangle)|2 + (|Language\rangle + |Processing\rangle)|3\rangle \\ &\quad + (|Processing\rangle + |Example\rangle)|4\rangle \\ &= |Classical\ Information\ Processing\ Example\rangle \\ &\quad + |Quantum\ Natural\ Language\ Processing\rangle \\ &= |Classical\ Information\ Language\ Processing\rangle \\ &\quad + |Quantum\ Natural\ Processing\ Example\rangle \\ &= \cdots \end{aligned} \tag{20}$$

This raises an ambiguity or does not preserve the identity of the phrases, as shown in Equation (21)

$$\begin{aligned}&|Classical\ Natural\ Language\ Processing\rangle \\ &\quad = |Classical\ Information\ Processing\ Example\rangle \\ &\quad = |Classical\ Information\ Language\ Processing\rangle \\ &\quad = \cdots \end{aligned} \tag{21}$$

Conversely, in the cTPR model, we obtain Equation (22)

$$\begin{aligned}|\psi_S\rangle &= |Classical\ Natural\ Language\ Processing\rangle \\ &\quad + |Quantum\ Information\ Processing\ Example\rangle \\ &= |Classical\rangle|Natural\rangle|Language\rangle|Processing\rangle \\ &\quad + |Quantum\rangle|Information\rangle|Processing\rangle|Example\rangle \end{aligned} \tag{22}$$

This latter representation is unambiguous but belongs to a higher-dimensional vector space, with the context size γ encompassing the entire structure.

3) FOCK SPACE REPRESENTATION MODEL

The Fock space representation model [40] draws inspiration from both the pTPR and cTPR models, aiming to leverage their advantages and address their shortcomings. The pTPR model represents symbolic structures using quantum state vectors, resulting in linear combinations that necessitate non-linear operations. However, quantum computers struggle with efficiently applying non-linear transformations to optimize the model's harmony. Harmony, in this context, is an objective function that maximizes the number of well-formed sentences [40]. Conventionally, a negative harmony value indicates an ungrammatical sentence, while zero signifies grammatical correctness. The goal is to modify the linear combination principle of the pTPR model by incorporating the contextual representation of the cTPR model. This adjustment avoids the non-deterministic linear combinations of pTPR generated by positional roles of symbols. Positional roles are transformed into factors of the tensor product of all symbols using maximal contextual roles. For instance, if we have N possible roles $r_1, r_2, \ldots, r_n$ and a set of symbols or fillers f_i to be encoded, instead of having a linear combination like:

$$|\psi_S\rangle = |f_1\rangle|r_1\rangle + |f_2\rangle|r_2\rangle + \ldots, \tag{23}$$

that is, a positional representation, we instead have a representation from cTPR as follows

$$|\psi_S\rangle = |f_{1,r_1}\rangle|f_{2,r_2}\rangle \cdots |f_{n,r_n}\rangle, \tag{24}$$

where each $|f_{1,r_1}\rangle$ is the filler (symbol) bound to the positional role r_i, having a new basis different from that of $|f_i\rangle$. For instance, if $|0\rangle$ signifies the positional role with no symbol or the empty character, and $a^\dagger_{f,r}|0\rangle$ is the new vector containing symbol f with positional role r, a basis for the vector space

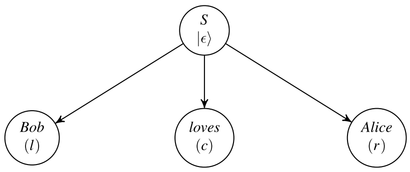

FIGURE 5. Ternary tree for transitive verbs. Example of ***loves*** in ***Bob loves Alice***. The transitive verb can be modeled by adding a start symbol and defining the positions of subject and object in relation to the verb itself.

of symbols with their roles can be defined as:

$$\begin{aligned} |v_s\rangle &= a^{\dagger}_{f_{1,r_1}}|0\rangle \otimes \cdots \otimes a^{\dagger}_{f_{n,r_n}}|0\rangle \\ &= a^{\dagger}_{f_{1,r_1}} \cdots a^{\dagger}_{f_{n,r_n}}|0\rangle, \end{aligned} \tag{25}$$

where $a^{\dagger}_{f_{i,r_j}}$ and $a_{f_{i,r_j}}$ are the quantum binding and unbinding operators, respectively, mapping the symbol and its positional role into a new state. Further details on this operator and its properties can be found in [40] and more investigations can be found in [49].

For example, let us consider the modeling of transitive verbs in the corpus *Bob loves Alice*. This corpus can be represented in a Fock space representation model by following these steps:

We begin by recognizing that a transitive verb in a sentence is grammatically correct if it is preceded by a subject and followed by an object. Thus, using a ternary tree and assigning roles $\{\epsilon, l, c, r\}$ to fillers (with l, c, s denoting 'left, center, right' daughter nodes), the corpus is depicted in the following form, as derived from Figure 5.

$$|\phi\rangle = |S\rangle \otimes |\epsilon\rangle + |Bob\rangle \otimes |l\rangle + |loves\rangle \otimes |c\rangle + |Alice\rangle \otimes |r\rangle, \tag{26}$$

where S is a start symbol.

A harmony operator $\mathcal{H}$ for recognizing a sentence containing a transitive verb is defined as follows:

- It assigns the value $n_{\epsilon_0} = -3$ to the symbol S in the role ϵ_0
- It assigns the value $n_{|Bob\rangle,l} = -1$ in role l
- It assigns the value $n_{|loves\rangle,c} = -1$ in role c
- It assigns the value $n_{|Alice\rangle,r} = -1$ in role r
- Finally, it assigns a $+2$ reward to the model for all links with $|S\rangle$.

Clearly, for our example, the harmony is $\mathcal{H} = (-3-1-1-1)+(+2+2+2) = 0$. This means that the sentence is grammatically correct.

In general, if we consider n_{ϵ_i}, n_s, n_{vt}, and n_o as the harmony of start symbols, subject, verb, and object, respectively, the harmony operator for an entire text corpus with m transitive verbs is:

$$\begin{aligned} \mathcal{H} = n_{\epsilon_0} - \sum_{i=0}^{m} \sum_{r\in\{l_i,c_i,r_i\}} (4n_{\epsilon_i} + n_{o,r} + n_{tv,r} + n_{s,r}) \\ + 2\sum_{i=0}^{m} \sum_{r\in\{l_i,c_i,r_i\}} n_{\epsilon_{i-1}}(n_{\epsilon_i} + n_{s,r} + n_{tv,r} + n_{o,r}) \end{aligned} \tag{27}$$

by adding the following rule:

- The harmony operator assigns harmony $n_{\epsilon_i} = -4$ to the symbol S in the role ϵ_i, $i \geq 1$.

Note that the harmony operator in this case is a classical function instead of a Hermitian operator, as it is a sum of the harmony operators $n_{r\prime}$, $r\prime \in \epsilon_i, (s, r), (tv, r), (o, r)$, and $n_{r\prime}\mathcal{H} = \mathcal{H}n_{r\prime}$ [40]. For this operator to be non-classical, all we need to do is define the local harmony operators $n_{r\prime}$ in such a way that $n_{r\prime}\mathcal{H} \neq \mathcal{H}n_{r\prime}$, i.e., the order in which we consider and apply the local harmony operators $n_{r\prime}$ counts.

In cases where the harmony operator $\mathcal{H}$ is non-classical, the harmony of a sentence is evaluated by the formula:

$$\mathcal{H}(|\phi\rangle) = \langle\phi|\mathcal{H}|\phi\rangle, \tag{28}$$

for any Fock space representation $|\phi\rangle$ of the sentence [40].

C. WORD2KET AND word2ketXS MODELS

Using vectors for modeling NLP tasks in classical computation has proven effective. Examples such as the word2vec model [50] or GloVe [51], representing words as vectors in NLP, demonstrate this effectiveness. Mapping words from a text corpus into vectors results in a more robust model that is well-suited for system learning. Specifically, this modeling transforms the discrete space of natural language into a continuous space, better accommodating neural network models. word2vec or GloVe can transform word corpora of dimension d into vectors of dimension $p < d$.

Consequently, rather than employing d-layer neural networks, we use p layers. In practical implementation, this requires a $d \times p$ matrix in CPU memory space, with d and p reaching values of 10^6 and 1024, respectively, in practice. Therefore, modeling and manipulating a large corpus, such as the English dictionary, with these vector models is cumbersome. To address this, word2ket and word2ketXS [46] are proposed, aiming to enhance vector models. word2ket and word2ketXS employ quantum entanglement of states to reduce vector dimension and memory footprint and facilitate manipulation through quantum operators.

1) WORD2KET MODEL

The word2vec function, denoted as $f : [d] \rightarrow \mathbb{R}^p$, transforms a d-token word vector into a p-dimensional vector. The concept of word2ket is to further transform the reduced vectors $v \in \mathbb{R}^p$ into very small q-dimensional basic elements of the tensor product, of rank r and order n, with $p = q^n$. Therefore, the representation of a d-token word vector $|\psi\rangle$ will be the superposition and entanglement of the token vectors $v' \in \mathbb{R}^q$ as given by Equation (29).

$$\begin{aligned} |\psi\rangle &= |v\rangle \\ &= \sum_{j=1}^{r} \bigotimes_{i}^{n} v'_{ij}. \end{aligned} \tag{29}$$

Embedding high d-dimensional vector words into basic vectors through the tensor product is even more efficient when visualized through the principle of parallel computing.

Construction of the state representing the word can be carried out in parallel, as shown by the cutout with parentheses in Equation (30), instead of performing a more costly sequential calculation as in Equation (31).

$$|\psi\rangle = \sum_j (v'_{1j} \otimes v'_{2j}) \otimes (v'_{3j} \otimes v'_{4j}). \tag{30}$$

$$|\psi\rangle = \sum_j \left((v'_{1j} \otimes v'_{2j}) \otimes v'_{3j}\right) \otimes v'_{4j}. \tag{31}$$

Thus, instead of performing n multiplications sequentially, we perform $\mathcal{O}(\log(n))$ multiplications. It also follows that the space required for a d-token word is $rnq = \mathcal{O}(rq\log(p/q))$, and for a corpus of m words, it requires a space of $\mathcal{O}(mp + rq\log(p/q))$ instead of $\mathcal{O}(dp)$ in classical computation [46].

2) word2ketXS MODEL

The word2ketXS model extends the word2ket model to word sets. word2ket transforms a vector of d token words into a tensor product of basic elements, each of size q. On the other hand, the word2ketXS model transforms a matrix M of size $d \times p$ from a d-word set vocabulary vector into a tensor product of a series of n linear operators or matrices M' of size $q \times t$. This transformation abides by the conditions $q^n = p$ and $t^n = d$, expressed as:

$$M = \sum_{j=1}^{r} \bigotimes_{i}^{n} M'_{ij}. \tag{32}$$

The word2ketXS model treats the d-token of word2ket, resulting in a simple vector, as a d-word vocabulary through linear operators. This, in turn, produces a $d \times p$ matrix. This approach is more memory-efficient, representing a d-word vocabulary in a single $d * p$ matrix instead of d individual vectors. The space required is $rnqt = \mathcal{O}(rqt\max(\log(p/q), \log(d/t)))$. The parallel calculation used by the word2ket model can also be exploited in the word2ketXS model to avoid building an embedding matrix.

The primary distinction between word2Ket and word2ketXS lies in the algorithm's input. While the word2Ket model takes a vector of dimension d as input, the word2ketXS model takes a matrix of dimension $d * p$. The objective of QNLP models is to reduce computation time and memory space, and this is where the two models differ. word2ketXS extends word2ket by grouping the independent word vectors of word2ket into a matrix. The advantage of word2ketXS over word2ket is that it performs a single operation on a large matrix containing several word vectors, saving computation time. Additionally, word2ketXS enables the storage of multiple words in a reduced-dimension matrix, saving memory space.

Figure 6 illustrates an example of word modeling and how these optimization objectives are achieved using the word2ket and word2ketXS models.

D. DISTRIBUTIONAL COMPOSITIONAL CATEGORICAL MODEL

DisCoCat, introduced by [44] and [45], stands out as the QNLP model warmly embraced by the QNLP community. In contrast to distributional models like the quantum bag-of-word model and other vector space models, DisCoCat goes beyond surface-level semantics. It incorporates both the syntactic and semantic structures of sentences by relying on the distributional nature of natural language and its compositional principles. This model harnesses quantum phenomena such as superposition and entanglement of states, grounded in two pivotal theories: compositionality theory of natural language and pregrouped grammar theory.

Compositionality theory posits that the meaning of a sentence or text is derived from the meanings of its constituent words. Natural language exhibits compositionality, allowing one to deduce the meaning of a sentence even when encountered for the first time. For instance, the phrase *orange juice* can be easily understood by grasping the meanings of *orange* and *juice*. It is easy to understand that it is probably a juice extracted from an *orange*. Thus, sentence meaning is a composition of the meanings of its individual words.

Pregroup grammar theory asserts that the grammatical correctness of a sentence can be assessed through calculations. The theory, developed by [52] and [53], associates a type p with each word, where each type p has left and right adjoints, denoted as p^l and p^r respectively. Grammatical reduction is computed based on these word types, following the rules outlined in Equation (33).

$$p \cdot p^r \rightarrow 1 \; ; \; p^l \cdot p \rightarrow 1 \tag{33}$$

Considering the typology principle, let n represent the noun type, s the sentence type, and $n^r s n^l$ the transitive verb type. The transitive verb type is deduced from the other typology, given that a transitive verb like "loves" in "Bob loves Alice" is surrounded by nouns of type n. Using these atomic types, the grammatical structure of a sentence, such as "Bob loves Alice," can be validated, as shown by the reduction inEquation (34), indicating grammatical correctness.

$$\overbrace{Bob}^{n}\ \overbrace{loves}^{n^r \cdot s \cdot n^l}\ \overbrace{Alice}^{n}$$
$$n \cdot (n^r \cdot s \cdot n^l) \cdot n \rightarrow (n \cdot n^r) \cdot s \cdot (n^l \cdot n) \rightarrow 1 \cdot s \cdot 1 \rightarrow s \tag{34}$$

This means that it is effectively a grammatically correct sentence. This observation naturally leads to the existence of a diagrammatic relationship between the different types, as in Figure 7.

Therefore, by employing a function that maps atomic types into vector spaces, such as n to N and s to S, and naturally $n^r s n^l$ to $N \otimes S \otimes N$, we can convert the NLP problem into a quantum model. This model harnesses the compositional and distributive structure of language using quantum operators.

Here, we elucidate how the model transforms sentences into a quantum system and how the model calculates

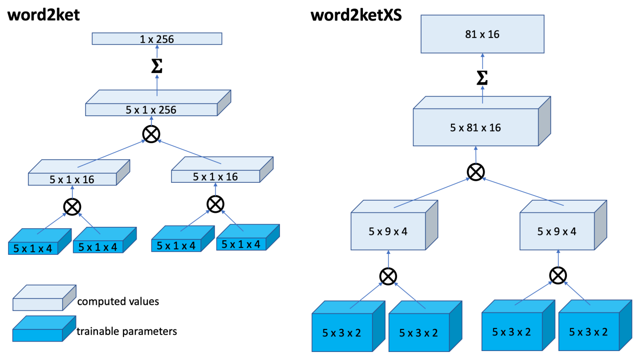

FIGURE 6. Word embedding with word2ket and word2ketXS models [46]. The representation of the word2ket (Left) model is as follows: The initial vector is of size 256. The second decomposition phase breaks down the initial vector of size 256 into 5 vectors, each with a size of 16. Each vector of size 16 is then decomposed into vectors of size 4, resulting in four blocks, each containing five vectors of size four. Regarding the representation of the word2ketXS (Right) model, instead of a vector as in word2ket, the initial input is a matrix of size 81 × 16. This initial matrix is then broken down into five smaller matrices of size 9 × 4. Finally, the matrices of size 9 × 4 are broken down into sub-matrices of size 3 × 2.

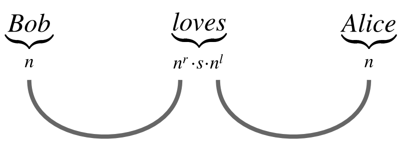

FIGURE 7. Diagrammatic relationship between the different constituents. This configuration computes the grammatical syntax of the sentence.

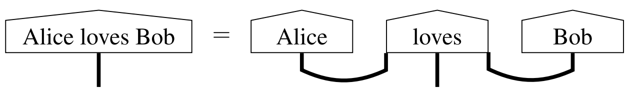

FIGURE 8. Diagrammatical representation to compute the meaning of the sentence.

meaning. To comprehend the workings of the model, let us consider a simple example: $|\psi\rangle$ = 'Alice loves Bob'. Before proceeding, it is essential to note a few aspects of this example. The sentence comprises three atoms or basic elements: two nouns, 'Alice' (the subject) and 'Bob' (the object), and another type, the transitive verb 'loves'. In the following, we designate the label n_s as the subject type ('Alice'), n_o as the object type ('Bob'), and tv as the type of the transitive verb. Additionally, we assume that $|\psi_{n_s}\rangle$ and $|\psi_{n_o}\rangle \in \mathbf{C}^2$. Since transitive verbs necessitate two types (a subject and an object), they can be regarded as a two-variable function f_{tv}, taking two nouns, $|\psi_{n_s}\rangle, |\psi_{n_o}\rangle \in \mathbf{C}^2$, to produce the state $|\psi_{n_s.tv.n_o}\rangle \in \mathbf{C}^{2k}$, where k is the size of the space of $|\psi\rangle$. Thus, f_{tv} applied to $|\psi\rangle$ gives:

$$|\psi_{n_s.tv.n_o}\rangle = f_{tv}\big(|\psi_{n_s}\rangle \otimes |\psi_{n_s}\rangle\big) \tag{35}$$

We also note that the transitive verb $|\psi_{tv}\rangle \in \mathbf{C}^2 \otimes \mathbf{C}^{2k} \otimes \mathbf{C}^2$ because of the interactions with the subject and the object.

Using the diagrammatic structure of natural language among the different atoms, the sentence's meaning can be calculated, as depicted in Figure 8.

The thick wire represents $\mathbf{C}^{2k}$, i.e., the spatial vector of the entire sentence, and the links between the atoms manifest the entanglement between their different meanings, forming the sentence's overall meaning. Employing quantum operators to transfer different meanings between atoms, the meaning of this sentence can be calculated using Equation (36).

$$\begin{aligned} |\psi_{n_s.tv.n_o}\rangle = \big(&\langle Bell| \otimes I \otimes \langle Bell|\big) \\ &\times \circ\big(|\psi_{n_s}\rangle \otimes |\psi_{tv}\rangle \otimes |\psi_{n_o}\rangle\big) \end{aligned} \tag{36}$$

where the Bell states stand for the cups and caps of the diagrammatic structure, respectively, $\bigcup = |00\rangle + |11\rangle$ and $\bigcap = \langle 00| + \langle 11|$. The operator $\langle Bell| \otimes I \otimes \langle Bell|$ represents the grammar applied to the equivalent quantum sentence $|\psi_{Alice}\rangle \otimes |\psi_{loves}\rangle \otimes |\psi_{Bob}\rangle$ and I is the identity matrix.

For an in-depth understanding of the model and how language types, adjectives applied to nouns, subjects and objects applied to a verb, and words such as relative pronouns and other cases are modeled, we direct the reader to the foundational document of the DisCoCat model [54].

E. HYBRID AND QUANTUM MACHINE LEARNING MODELS

A burgeoning trend in NLP models optimization involves combining classical machine learning techniques with quantum machine learning methods. Classical machine learning techniques [55], [56], [57] empower machines to learn unknown functions from known inputs or data to predict the outputs of unknown inputs. Let $\{x_1, x_2, \cdots, x_m\}$ represent the known inputs and $\{y_1, y_2, \cdots, y_m\}$ denote the corresponding outputs, arranged in index order. Classical machine learning entails determining the weights $w = \{w_1, w_2, \cdots, w_n\}$ such that $y_i = x_i w$, where each x_i is a vector of size n, $x_i = \{x_{i1}, x_{i2}, \cdots, x_{in}\}$. Generally, classical methods consist of a succession of layers of artificial neurons, known as neural networks (NN), which have evolved over the years, leading to innovations like convolutional neural networks [58], [59], [60] (used in computer vision), recurrent neural networks [61], [62], [63], [64] (applied in time series and NLP), ResNets [65], and Transformers [66], [67], [68], [69].

In recent years, numerous research projects have proposed hybrid or quantum versions of machine learning, continuously enhancing them. Due to their applicability in NLP and various other domains, quantum neural networks (QNN) have garnered particular attention [43], [70], [71], [72], [73]. The QNN can be viewed as the transformation $|\psi\rangle_{1,\cdots,n}|0\rangle \rightarrow |\psi_y\rangle_{1,\cdots,n}|y\rangle$, where the auxiliary qubit $|0\rangle$ records the response $|y\rangle$ at the process end, and $|\psi_y\rangle_{1,\cdots,n}$ represents the transformed input quantum state, which also, in some sense, represents the weights to be learned. This transformation highlights the reversibility of quantum computation. Unlike classical computation, implying the inverse transformation can recover the input state $|\psi\rangle_{1,\cdots,n}|0\rangle$ from the output quantum state $|\psi_y\rangle_{1,\cdots,n}|y\rangle$.

The input state $|\psi\rangle_{1,\cdots,n}$ represents purely quantum data or classical data transformed into quantum data. Classical data from natural language is transformed into quantum states $|\psi\rangle_{1,\cdots,n} = \sum_{i=1}^{2^n} a_i|x_i\rangle$, where a_i, $i \in \{1, \cdots, 2^n\}$ represent the amplitudes (the dependencies between words, for example).

Various methods have been developed for transforming classical natural language data into quantum data. One popular method used in QNN is the binarization [74] of classical data. The main idea is to binarize each component of the corpus, with each binary value corresponding to an eigenstate in the calculation basis of the quantum compute.

As an example, let's consider modeling NLP using a quantum self-attention QNN [73], [75] to represent the corpus *Hello the quantum world*. Among several other quantum machine learning models, this deliberate choice aims to demonstrate the advantages that quantum methods can offer in NLP modeling through this example. This choice is due to the simple mathematical formulation of quantum self-attention.

Firstly, classical data is converted into quantum states using the binarization method. The binarization would correspond *Hello* to 00, *the* to 01, *quantum* to 10, and *world* to 11. Each binary value is then associated with a eigenstate in the computation basis. In this example, two qubits may suffice to represent the corpus, as in Equation (37). Remember, a quantum system of n qubits can represent 2^n classical values in superposition (see section V-B1).

$$|Q_r\rangle = \frac{1}{\sqrt{30}}|00\rangle + \frac{2}{\sqrt{30}}|01\rangle + \frac{3}{\sqrt{30}}|10\rangle + \frac{4}{\sqrt{30}}|11\rangle. \tag{37}$$

Once the data has been encoded, the resulting quantum states are fit to the model for training and inference purposes. The training of an NLP model using the quantum self-attention mechanism, in steps, would proceed as follows.

An encoder/decoder architecture model using quantum Self Attention takes a set of keys and values as inputs and generates attention scores attention to predict the correct response. The keys and their values are learned during the model training. During the model inference, the query represents, for example, a text to be translated, prompts for text generation, or any other NLP task [75].

For simplicity, let's consider the quantum state $|Q_r\rangle$ as a query passed to the model for machine translation purpose. Additionally, let's assume that this same state is also used as the key of the correct value or translation, denoted as V_r, in the attention mechanism of the model,

$$|K_r\rangle = \frac{1}{\sqrt{30}}|00\rangle + \frac{2}{\sqrt{30}}|01\rangle + \frac{3}{\sqrt{30}}|10\rangle + \frac{4}{\sqrt{30}}|11\rangle, \tag{38}$$

The quantum self-attention mechanism will be calculated by the model as follows:

$$\begin{aligned} &Attention(Q, K, V) \\ &= \odot\langle Q|K\rangle \otimes |V\rangle \\ &= \begin{bmatrix} \langle Q_0|K_0\rangle \otimes |V_0\rangle \odot \cdots \odot \langle Q_0|K_{n-1}\rangle \otimes |V_{n-1}\rangle \\ \langle Q_1|K_0\rangle \otimes |V_0\rangle \odot \cdots \odot \langle Q_1|K_{n-1}\rangle \otimes |V_{n-1}\rangle \\ \vdots \\ \langle Q_{n-1}|K_0\rangle \otimes |V_0\rangle \odot \cdots \odot \langle Q_{n-1}|K_{n-1}\rangle \otimes |V_{n-1}\rangle \end{bmatrix}, \end{aligned} \tag{39}$$

where $\langle Q|K\rangle$ represents the operation that computes the similarity between the queries Q and the keys K, while the operator $\odot$ is utilized to select the correct answer V_r, which corresponds to the query Q_r and the key K_r, from the set of values, V. This selection process using the $\odot$ operator can be executed through a sequence of quantum gates, such as **CNOT** gates. The accurate value or translation, V_r, is determined based on the highest similarity value $\langle Q_r|K_r\rangle$, here equal to 1, because $|Q_r\rangle$ and $|K_r\rangle$ represent the same quantum state.

The keys K are acquired and stored by the model during the training phase. For instance, the key $|K_r\rangle = \frac{1}{\sqrt{30}}|00\rangle + \frac{2}{\sqrt{30}}|01\rangle + \frac{3}{\sqrt{30}}|10\rangle + \frac{4}{\sqrt{30}}|11\rangle$ is associated with the response V_r= *Bonjour le monde quantique*, such as in the case of French translation.

In this example, from Equation (39), the learnable matrix

$$\begin{bmatrix} \langle Q_0|K_0\rangle \cdots \langle Q_0|K_{n-1}\rangle \\ \langle Q_1|K_0\rangle \cdots \langle Q_1|K_{n-1}\rangle \\ \vdots \\ \langle Q_{n-1}|K_0\rangle \cdots \langle Q_{n-1}|K_{n-1}\rangle \end{bmatrix}, \tag{40}$$

reflects the distribution of attention scores to be learned during the model training.

One notable advantage of quantum NLP models is illustrated in this example. It's worth noting that each element of $\langle Q|K\rangle$ is a superposition of states, thus offering greater expressiveness in terms of information and features. Consequently, the quantum self-attention mechanism holds an advantage in terms of storage space and computation time efficiency [75].

Binarization mapping may suffer from the drawback, especially with continuous variables, of potential data loss. However, a solution to this issue is fine-grained representation [76], which involves increasing the degree of binarization to capture more continuous values. Another effective method for representing continuous variables without compromising data integrity is what can be termed electromagnetic representation [77], which entails encoding classical data into quantum states, such as the amplitudes of electromagnetic fields.

Various hybrid models of quantum machine learning have been proposed for immediate practical application in industry. These models typically involve an architecture that divides computation into an easy part, running on a classical processor, and a challenging part, running on a quantum processor [78], [79], [80]. Approaches like gradient-based optimization are used to learn parameters during training, as seen in [81], [82], and [83]. Hybrid models like [79] incorporate classical non-linear activation functions. Additionally, [84], [85] proposed a quantum auto-encoder for encoding data in compressed quantum states. Several practical applications of hybrid quantum machine models have been successfully explored in the realm of NLP [86], [87], [88], [89], [90], [91].

VIII. SUMMARY OF RELATED WORK ON QUANTUM NATURAL LANGUAGE PROCESSING

This section provides an overview of the work in the field of QNLP. Despite being in its infancy, numerous enhancements and applications have been proposed for QNLP models. For each model, we present an overview of the developed algorithms, distinguishing between improvements on the model itself and applications to NLP-related tasks. Table 1 summarizes the work conducted in the QNLP domain.

The interest in DisCoCat primarily stems from the model's capability to compute corpus meanings and validate grammatical syntax. Upon its proposal, efforts were initiated to optimize it for practical implementation on NISQs. The work introduced by [54] generalizes the modeling of various sentence types in text corpora. One of the principal challenges with this model is associated with the resources required for its implementation. The interaction among different sentence components is highly resource-intensive. The representation of entire sentences leads to states or vectors with very high dimensions, making constructing and handling such vectors challenging.

Consequently, using an easily accessible QRAM [92] for the practical implementation of DisCoCat holds significant promise. However, the practical realization of such QRAMs remains unattainable in the current quantum technology industry. Therefore, there is an ongoing need to develop alternatives to QRAMs.

In the work presented by [93], the authors proposed a full-stack pipeline to create the DisCoPy library for implementation on near-term quantum computers. This library is employed in numerous experiments testing the DisCoCat model. Using DisCoPy, [94] conducted sentence classification tasks on quantum hardware for datasets comprising 100 or more sentences. In a similar vein, [95] and [96] provided alternative implementations for sentence classification, while [54] tackled the question-answering task. The results obtained in these models demonstrate the superiority of quantum models for NLP, even though the size of the datasets remains insufficient to support such a claim definitively. Additionally, implementing the model on other tasks, such as translation, has proven effective. For example, [97] and [98] proposed a modeling of language translation from English to Persian.

TABLE 1. Summary of related work on QNLP.

Ref.	Model	Description/Application	Year
[95]	DisCoCat	Sentence classification	2023
[96]	DisCoCat	Sentence classification	2023
[94]	DisCoCat	Implementation on quantum hardware and results for datasets of size $\geq$ 100 sentences	2023
[105]	DisCoCat	A model for musical grammars	2022
[106]	DisCoCat	An approach to Pronoun esolution	2022
[107]	DisCoCat	Quantum sentence generation	2022
[108]	DisCoCat	Sentiment classification	2022
[97]	DisCoCat	Language Translation from English to Persian	2021
[98]	DisCoCat	Language Translation from English to Persian	2021
[109]	DisCoCat	The generic architecture and description of the most important modules of lambeq library implementation of the model and extend the algorithm into a quantum algorithm to categorize sentences	2021
[93]	DisCoCat	A full-stack pipeline for natural language processing on near-term quantum computers	2020
[54]	DisCoCat	DisCoCat model generalization for sentences	2020
[54]	DisCoCat	Question answering	2020
[95]	DisCoCat	Question answering	2020
[92]	DisCoCat	Uses a quantum RAM for	2016
[110]	DisCoCat	Quantum teleportation in NLP	2013
[42]	QBW	Text classification on NISQ	2022
[104]	QBW	Sentiment analysis	2021
[89]	QBW	Question answering	2020
[101]	QBW	Information retrieval	2020
[103]	QBW	Question answering	2018
[99]	QBW	Information retrieval	2018
[102]	QBW	Speech recognition	2017
[100]	QBW	Information retrieval	2015
[41]	QBW	Information retrieval	2013
[46]	word2ket +word2ketXS	text summarization, language translation, and question answering	2019

In comparison to DisCoCat, quantum bag-of-words models are tailored to specific tasks, such as information retrieval [41], [99], [100], [101], speech recognition [102], question answering [89], [103], and sentiment analysis [104]. These models leverage statistical calculations and dependencies between different corpus terms, utilizing elements like word and n-gram scores, density matrices of corpora to model the probability distribution over the corpus, and dependencies.

IX. QUANTUM NATURAL LANGUAGE PROCESSING FRAMEWORKS OR TOOLKITS

In Table 3, a selection of tools, references, and brief descriptions for QNLP are provided. Among these tools are compilation platforms, QNLP libraries, and parsers. The ones mentioned here are those most frequently encountered in various works or reports during this review.

X. PRACTICAL IMPLEMENTATION

A. GENERAL IMPLEMENTATION PIPELINE

This section outlines the general and common procedure for implementing these different models. The implementation

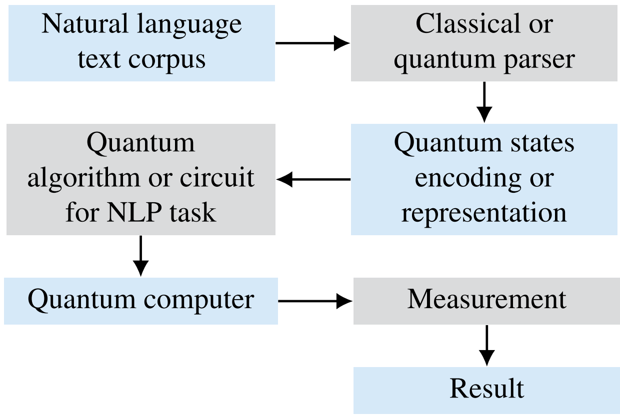

FIGURE 9. General pipeline for QNLP implementation. The process begins with the mapping of classical data into quantum states, through to measurements of the quantum values calculated after passing through the quantum algorithms. The measured values will be classical values from the quantum states and will be used to optimize NLP tasks or functions.

of quantum models for NLP follows a global and adaptable architecture, as outlined below:

- Data Collection: The first step involves collecting data to construct a natural language corpus.
- Preprocessing: This step includes cleaning classical data and performing preprocessing operations such as tokenization. The goal is to transform classical data into a representation of quantum states. The choice of the quantum state format depends on the selected model for implementation and the task at hand, as described in the first step. Tools used for this step may include classical parsers, such as the CCG parser, and an encoder like DisCoPy to transform symbols or words into quantum states.
- Problem formulation: This step clearly defines the NLP problem or task to be accomplished or performed, such as sentiment analysis, translation, question answering, etc.
- Quantum Algorithm or Quantum Circuit Design: This stage involves defining the quantum procedure that performs the required NLP task. This procedure is then translated into a quantum circuit using quantum operators, such as quantum gates, for execution on the quantum processor.
- Execution on Quantum Computer: The algorithm is executed on the quantum computer with the encoded quantum states as inputs.
- Measurement of the quantum states produced by running the algorithm to generate classical results is done.

A general QNLP implementation pipeline is presented in Figure 9.

B. DATASETS, METRICS AND EVALUATION

This section presents the details of the data employed to test various models and their corresponding results. Each experiment outlines the evaluation metrics used alongside

TABLE 2. Advantages and disadvantages of different models.

Models	Advantages	Disadvantages
QBW	- Quantum-inspired computation - It provides a simple representation - Parallel processing	- Struggle with handling large corpus - A lot of computing time and memory space - Lacks the ability to capture the semantic relationships
pTPR	- Dependencies parse trees encoded in a quantum computer - Incorporates word position - NLP tasks and applications based on the position of symbols are easy to perform	- The vectors used are of high dimension - Can become ambiguous in the representation
cTPR	- Unambiguous, superposition can preserve identity - Incorporates word context	- High vector dimension space - Challenging to scale up to larger texts
FSR	- NLP transformed into optimization problem	- Limited to NLP tasks that can be transformed into optimization problems
word2ket	- Reduction of memory space - Captures semantic nuances and context dependencies in language	- Requires a quantum hardware
word2ketXS	- Reduction of memory space - Handles larger vocabularies	- Requires a quantum hardware
DisCoCat	- Supports compositionality and distributionality - Rich and expressive representation of language	- Contextual nature of meaning is unhandled - Requires a quantum hardware

tests. Even though, due to the constraints of current quantum infrastructures, the results of quantum models reported in these experiments are conducted on very limited datasets to assess superiority over classical models, these results are promising. Table 4 provides insights into the data, metrics, and model evaluations for specific NLP tasks.

XI. QUANTUM NATURAL LANGUAGE PROCESSING CHALLENGES

One of the primary challenges in QNLP models revolves around the issue of reducing the dimension of the computational space, a problem shared by nearly all models. For QNLP to be effective, the model must facilitate statistical computations on lengthy texts and enable interactions among various components, including words and sentences. Specifically, to ascertain the meaning of a sentence, the model must accommodate interactions among its different components. Tasks like text extraction and question-answering necessitate the model's capability to establish similarity and dependency scores and indexes for efficiency and reliability. However, as the constituents of a sentence interact to convey their dependencies, the computational space expands significantly.

TABLE 3. Some QNLP frameworks and toolkits.

References	Description
DisCopy [2] [94]	Python Library for DisCoCat model
$t\|ket\rangle$ (pytket)[3]	Quantum Compiler
CCG parser	Used for the pregroup parsing for providing the syntax trees for DisCoCat
Qtransformer [4]	To promote the Transformer from the classical to quantum real
Qulacs library [111]	Performs simulations of quantum processes
Quanthoven [5]	DisCoCat music model
Lambeq [109]	An efficient high-level python library for quantum NLP.
Word2ket [6]	word2ket implementation
PyTorch [7]	for encoding of sentences as tensor network for training
JAX [8]	For encoding of sentences as quantum circuit
QuantumNLP [9]	Quantum sentence generation

Particularly, when considering tasks performed on entire books, the undertaking becomes unfeasible for NISQ. This issue is particularly relevant to models such as DisCoCat and TPR, which involve the use of tensor products of order n, with dimensions reaching up to a thousand or even a million [54], making them impractical for large texts.

Another challenge is the models' reliability, with none proving reliable for all NLP-related tasks. In this context, model reliability refers to the ability to perform various NLP tasks concurrently. Some models are well-suited for specific categories of NLP tasks, while others are not. Quantum bag-of-words models, for instance, are tailored to specific tasks like information retrieval, speech recognition, question-answering, and sentiment analysis. However, this model appears intricate for tasks such as sentence meaning computation, where the DisCoCat model excels by being suitable for both sentence meaning computation and grammatical syntax checking.

An ideal QNLP model should effectively combine natural languages' distributional and compositional features. DisCoCat addresses this challenge to some extent, but further work is required. For instance, integrating reliable mechanisms to enhance statistical calculations, such as word scores and pregroup grammar calculations, would contribute to its robustness.

XII. FUTURE DIRECTION

Developing more inclusive and efficient models for QNLP is still a pressing need. Models capable of considering natural languages' compositional and distributive nature are essential for efficient processing. A promising research direction involves the development of models with a multi-layer or modular architecture that integrates the compositional and distributional structure of natural languages, making them more inclusive and effective for QNLP. Combining architectures from different models could result in a more optimized solution. Reducing vector dimension resulting from the interaction of corpus elements will be beneficial for processing large texts effectively. Unfortunately, the interaction between language elements represented in vectors poses a challenge to the efficient implementation of QNLP, involving a high number of vectors, up to the number of words in a book, resulting in very high-dimensional vector spaces. Parallelism, as seen in the word2ket model, could serve as a potential solution in the calculation process. The extensive integration of parallel vector computation into model architectures will help reduce high-dimensional vectors' construction and manipulation costs. This way, the compositional nature of natural languages could be reduced to sub-compositions, avoiding full composition leading to high-dimensional spaces. Additionally, the reduction of statistical computation time on the distributional layer in a compositional-distributional model is of great importance and can be explored through quantum operators capable of performing simultaneous computations on sentence matrices or whole corpus matrices. Purely compositional and purely distributive models have limitations, and as we understand today, NLP exhibits both distributional and compositional properties, considering elements such as context, semantics, and syntactic structure when analyzing and understanding languages.

The quantum large language model (QLLM) has not received sufficient attention in the literature, yet it holds promise as a solution for the challenges posed by the LLM. Classical LLM presents challenges such as the extensive computational resources required for training and deployment, biases and fairness issues, and uncontextual and incoherent responses. LLM training and deployment demand substantial computational resources, with training times estimated in years and necessitating supercomputers. QNLP appears to offer a solution for exponentially reducing computing time. LLM-generated responses often exhibit bias or discrimination, and QNLP can enhance rapid response selection through its ability to superimpose and analyze all possible response contexts.

The multilanguage model (MLM), focusing on understanding and generating text in multiple languages, presents challenges related to computation time and language complexity due to variations in vocabulary, grammatical syntax, and context nuances. Constructing MLMs with these characteristics is highly complex, and integrating quantum methods for parsing, code-switching, response generation, and model training promises higher-performance models.

Purely compositional and purely distributive models have limitations, and as we understand today, NLP exhibits both distributional and compositional properties, considering elements such as context, semantics, and syntactic structure when analyzing and understanding languages.

The quantum large language model (QLLM) has not received sufficient attention in the literature, yet it holds promise as a solution for the challenges posed by the LLM. Classical LLM presents challenges such as the

TABLE 4. Datasets, metrics and evaluation of experiments per model.

QUANTUM BAG-OF-WORDS MODELS			
Models/Tasks	**Datasets**	**Metrics**	**Evaluation**
Classification task [42]	Lambeq datasets [105]	accuracy	62%
Information Retrieval [41]	TREC collections [41]	Mean Average Precision (MAP)	relative improvements in MAP going up to 12.1% in the case of WT10g collection and 19.2% for the ClueWeb-B collection
Information Retrieval [100] with a Quantum Entanglement based Model	five collections in the documents re-ranking task [100]	MAP	achieves better performances than QLM (a quantum IR model) and MRF T UPD model (a traditional IR model with UPD patterns)
Information Retrieval [99]	TREC 2013 and 2014 session track datasets [10, 11]	MAP@10 and nDCG@10 metrics	TREC 2013 : 10.37(71.40) vs 8.94(89.81) for nDCG@10, MAP@10 respectively. REC 2014: 15.19(6.82) vs 14.79(1.86) for Improvements in parentheses.
Automatic speech recognition [102]	TIMIT corpus [106]	Perplexity (PPL), number of parameters	Results are better than the other systems tested.

TPR MODEL			
Models/Tasks	**Datasets**	**Metrics**	**Evaluation**
HARMONY MAXIMIZATION (number of well formed sentences) [40]	Tree rules for grammatical expressions of the form $A^n.B^n$	Average complexity	Empirical average complexity of harmonizing to the $A^n.B^n$ expression is logarithmic $(log(n))$ on classical computer and the total number of iterations needed is $O(D^2/\delta^2)$ on quantum computer, D is the operator dimension and δ the markov chain gap

word2wet AND word2ketXS			
Models/Tasks	**Datasets**	**Metrics**	**Evaluation**
Summarization language translation and question answering [48],	Stanford Question Answering Dataset (SQuAD) datase	F1 score, Accuracy,	The experiments show substantial decreases in the memory footprint. In text summarization, word2ket achieve 16-fold reduction in trainable parameters and 34,000 fold reduction in trainable parameters for word2ketXS

DisCoCat MODELS			
Models/Tasks	**Datasets**	**Metrics**	**Evaluation**
Classification tasks for sentences [94]	MC (‘meaning classification’) [94], RELPRON [107]	train and test errors	Train and test errors read 16.9% and 20.2%, respectively(MC); 9.4% and 27.7%, respectively(RP)
Machine Translation [97]	English-Persian [97]	Complexity, accuracy	Quadratic speedup, better accuracy over classical methods
Audio processing (music classification) [108]	musical snippets [108]	Accuracy	76%
Classification of sentences [95]	Corpora [95]		proof-of-concept work, QNLP is possible on available quantum devices
Natural Language Generation [109]	Generation of 30 sentences about food	Avg No. of guesses	8.5 (hybrid quantum) vs 8.4(Random Generation and Testing)

HYBRID AND QUANTUM MACHINE LEARNING MODELS			
Models/Tasks	**Datasets**	**Metrics**	**Evaluation**
Question Answering [89]	TREC-QA [110] and WikiQA datasets [111]	Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR)	11.87% MAP and 13.61% MRR on TREC-QA, and by 27.15% MAP and 28.09% on WIKIQA
Text Classification [90]	Snips [112] and ATIS [113] Dataset	Average Prediction Accuracy	Boosts the performance in two text classification datasets by 1.57% and 1.52% relative improvements.
Transfer learning for spoken command recognition [91]	Google speech command dataset [114]	Accuracy, cross-entropy (CE), parameters	Better accuracy (94.58% vs. 94.42%), lowest CE value (0.248 vs. 0.251) and fewer parameters (0.00096 vs. 0.216)
Quantum Self Attention Neural Network for Text Classification [73]	MC and RP [94] Yelp, IMDb and Amazon [115],	Accuracy, parameters	100% test accuracy, only 25 parameters vs 80%, 40 parameters (DisCoCat). QSANN has potential advantage for text classification
Sentiment Analysis	MELD[12] [116] and IEMOCAP[13] [117]	F1, Accuracy	Achieves the best classification results on all metrics, increased by 9.1% and 6.7%(MELD), 7.4% and 4.2% (IEMOCAP)

[2] https://github.com/oxford-quantum-group/discopy
[3] https://github.com/CQCL/pytket
[4] https://github.com/rdisipio/qtransformer
[5] https://github.com/CQCL/Quanthoven
[6] https://github.com/panaali/word2ket
[7] https://pytorch.org
[8] https://github.com/google/jax
[9] https://bit.ly/QuantumNLG
[10] https://trec.nist.gov/data/session2013.html
[11] https://trec.nist.gov/data/session2014.html
[12] https://affective-meld.github.io/
[13] http://sail.usc.edu/iemocap/

extensive computational resources required for training and deployment, biases and fairness issues, and uncontextual and incoherent responses. LLM training and deployment demand substantial computational resources, with training times estimated in years and necessitating supercomputers. QNLP appears to offer a solution for exponentially reducing computing time. LLM-generated responses often exhibit bias or discrimination, and QNLP can enhance rapid response selection through its ability to superimpose and analyze all possible response contexts.

The multilanguage model (MLM), focusing on understanding and generating text in multiple languages, presents challenges related to computation time and language complexity due to variations in vocabulary, grammatical syntax, and context nuances. Constructing MLMs with these characteristics is highly complex, and integrating quantum methods for parsing, code-switching, response generation, and model training promises higher-performance models.

Integrating quantum system learning into QNLP models is also a highly advantageous research direction. The well-established importance and advantages of quantum system learning over conventional system learning highlight its vital role in QNLP. While compositional and distributional methods mentioned earlier can be effective, the added benefit of system learning lies in its ability to detect patterns in data that may elude human perception. With the ever-growing size of data, simple statistical methods become insufficient for NLP tasks. Quantum machine learning, mainly through quantum neural networks, detects learning patterns in QNLP more effectively. The efficient implementation of compositional and distributional models with quantum system learning will be pivotal for the future of QNLP. Integrating quantum system learning into QNLP models is also a highly advantageous research direction. The well-established importance and advantages of quantum system learning over conventional system learning highlight its vital role in QNLP. While compositional and distributional methods mentioned earlier can be effective, the added benefit of system learning lies in its ability to detect patterns in data that may elude human perception. With the ever-growing size of data, simple statistical methods become insufficient for NLP tasks. Quantum machine learning, mainly through quantum neural networks, detects learning patterns in QNLP more effectively. The efficient implementation of compositional and distributional models with quantum system learning will be pivotal for the future of QNLP.

XIII. CONCLUSION

Despite being in its infancy, QNLP deserves investigation. This paper reviews various NLP embedding models to explore opportunities and challenges. The results presented in this work show that current models exhibit promise for specific NLP tasks. Different models are identified, including the quantum bag-of-word model, the TPR model, the DisCOCat model, and the word2ket and word2ketXS models. Some models consider the distributional character, others focus on the compositional character, and some attempt to incorporate both. However, the most efficient model would account for both aspects of natural language properties. Developing more efficient and reliable models for various NLP tasks is imperative. The application of quantum models to specific NLP-related tasks, such as text-to-image generation, text-to-scene generation, and text-to-video, has not been fully realized, but quantum computing holds great potential for these applications. Current experimental work on quantum hardware underscores the applicability of quantum methods to NLP, warranting further investigation for more efficient models. While the proposed models show efficiency, implementing them on a large scale, such as the English language dictionary, remains challenging due to the necessary resources. Further enhancements are crucial for the future of QNLP.

REFERENCES

[1] D. Orrell, *Quantum Economics and Finance*. New York, NY, USA: Panda Ohana Publishing, 2022.
[2] D. Widdows, "A mathematical model for context and word-meaning," in *Proc. 4th Int. Interdiscipl. Conf.*, 2003, pp. 369–382.
[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North*, 2019, pp. 1–16.
[4] L. De Angelis, F. Baglivo, G. Arzilli, G. P. Privitera, P. Ferragina, A. E. Tozzi, and C. Rizzo, "ChatGPT and the rise of large language models: The new AI-driven infodemic threat in public health," *Frontiers Public Health*, vol. 11, pp. 1–18, Apr. 2023.
[5] C. D. Manning, "Human language understanding & reasoning," *Daedalus*, vol. 151, no. 2, pp. 127–138, May 2022.
[6] J. Hutchins. (2014). *The History of Machine Translation in a Nutshell*. [Online]. Available: https://aclanthology.org/www.mt-archive.info/10/Hutchins-2014.pdf
[7] R. Shams, "Semi-supervised classification for natural language processing," 2014, *arXiv:1409.7612*.
[8] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie, "A survey on evaluation of large language models," *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 3, pp. 1–45, Mar. 2024.
[9] K. Keshamoni, *ChatGPT: An Advanceds Natural Language Processing System for Conversational AI Applications—A Comprehensive Review and Comparative Analysis With Other Chatbots and NLP Models*. Cham, Switzerland: Springer, 2023, pp. 447–455.
[10] K. I. Roumeliotis and N. D. Tselikas, "ChatGPT and open-AI models: A preliminary review," *Future Internet*, vol. 15, no. 6, p. 192, May 2023.
[11] C. Li. (2020). *OpenAI's GPT-3 Language Model: A Technical Overview*. [Online]. Available: https://lambdalabs.com/blog/demystifying-gpt-3
[12] OpenAI. (2022). *Chatgpt: Optimizing Language Models for Dialogue*. [Online]. Available: https://openai.com/blog/chatgpt
[13] L. Lakshmanan. (2022). *Why Large Language Models (like ChatGPT) are Bullshit Artists*. [Online]. Available: https://becominghuman.ai/why-large-language-models-like-chatgpt-are-bullshit-artists-c4d5bb850852
[14] A. News. (2023). *What Can ChatGPT Maker's New Ai Model GPT-4 Do?*. [Online]. Available: https://abcnews.go.com/International/
[15] B. Perrigo. (2022). *AI Chatbots are Getting Better. But an Interview With ChatGPT Reveals Their Limits*. [Online]. Available: https://time.com/6238781/chatbot-chatgpt-ai-interview/
[16] S. Biddle. (2022). *The Internet's New Favorite AI Proposes Torturing Iranians Surveilling Mosques*. [Online]. Available: https://theintercept.com/2022/12/08/openai-chatgpt-ai-bias-ethics/
[17] B. Mocialov, G. Turner, and H. Hastie, "Transfer learning for British sign language modelling," in *Proc. 5th Workshop NLP Similar Language Varieties Dialects*, 2018, pp. 1–20.
[18] S. Wu, J. Li, P. Zhang, and Y. Zhang, "Natural language processing meets quantum physics: A survey and categorization," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2021, pp. 3172–3182.
[19] R. Guarasci, G. De Pietro, and M. Esposito, "Quantum natural language processing: Challenges and opportunities," *Appl. Sci.*, vol. 12, no. 11, p. 5651, Jun. 2022.
[20] A. Rai, "A review article on quantum natural language processing," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 1, pp. 1588–1594, Jan. 2022.
[21] S. A. Thorat and V. Jadhav, "A review on implementation issues of rule-based chatbot systems," in *Proc. Int. Conf. Innov. Comput. Commun. (ICICC)*, 2020, pp. 1–54.

[22] D. Sykes, A. Grivas, C. Grover, R. Tobin, C. Sudlow, W. Whiteley, A. Mcintosh, H. Whalley, and B. Alex, "Comparison of rule-based and neural network models for negation detection in radiology reports," *Natural Lang. Eng.*, vol. 27, no. 2, pp. 203–224, Mar. 2021.

[23] H. Li, H. Mao, and J. Wang, "Part-of-speech tagging with rule-based data preprocessing and transformer," *Electronics*, vol. 11, no. 1, p. 56, Dec. 2021.

[24] Y. Goldberg, *Neural Network Methods for Natural Language Processing*. Cham, Switzerland: Springer, 2022.

[25] A. Torfi, R. A. Shirvani, Y. Keneshloo, N. Tavaf, and E. A. Fox, "Natural language processing advancements by deep learning: A survey," 2020, *arXiv:2003.01200*.

[26] I. Lauriola, A. Lavelli, and F. Aiolli, "An introduction to deep learning in natural language processing: Models, techniques, and tools," *Neurocomputing*, vol. 470, pp. 443–456, Jan. 2022.

[27] S. Wu, K. Roberts, S. Datta, J. Du, Z. Ji, Y. Si, S. Soni, Q. Wang, Q. Wei, Y. Xiang, B. Zhao, and H. Xu, "Deep learning in clinical natural language processing: A methodical review," *J. Amer. Med. Inform. Assoc.*, vol. 27, no. 3, pp. 457–470, Mar. 2020.

[28] M. Pikuliak, M. Šimko, and M. Bieliková, "Cross-lingual learning for text processing: A survey," *Expert Syst. Appl.*, vol. 165, 2021, Art. no. 113765.

[29] Z. Chi, L. Dong, F. Wei, W. Wang, X. L. Mao, and H. Huang, "Cross-lingual natural language generation via pre-training," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 5, pp. 7570–7577.

[30] K. Qi, H. Wan, J. Du, and H. Chen, "Enhancing cross-lingual natural language inference by prompt-learning from cross-lingual templates," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, 2022, pp. 1910–1923.

[31] J. Wang, "Cross-lingual transfer learning for low-resource natural language processing tasks," M.S. thesis, Inst. Anthropomatics Robot., Karlsruhe, Germany, 2021. [Online]. Available: https://isl.anthropomatik.kit.edu/pdf/Wang2021.pdf

[32] C. Maria Keet, "Bootstrapping NLP tools across low-resourced African languages: An overview and prospects," 2022, *arXiv:2210.12027*.

[33] Y. Zhang, R. He, Z. Liu, L. Bing, and H. Li, "Bootstrapped unsupervised sentence representation learning," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Language Process.*, 2021, pp. 5168–5180.

[34] F. Chen, M. Han, H. Zhao, Q. Zhang, J. Shi, S. Xu, and B. Xu, "X-LLM: Bootstrapping advanced large language models by treating multi-modalities as foreign languages," 2023, *arXiv:2305.04160*.

[35] O. Ikechukwu, O. Ikechukwu-Onyenwe, and O. Ebele, "A review on part-of-speech technologies," 2021, *arXiv:2110.04977*.

[36] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[37] T. Pellizzari, S. A. Gardiner, J. I. Cirac, and P. Zoller, "Decoherence, continuous observation, and quantum computing: A cavity QED model," *Phys. Rev. Lett.*, vol. 75, no. 21, pp. 3788–3791, Nov. 1995.

[38] B. Coecke, *From Quantum Foundations via Natural Language Meaning to a Theory of Everything*. Springer, 2017, pp. 63–80.

[39] C. J. van Rijsbergen, *The Geometry of Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[40] N. Wiebe, A. Bocharov, P. Smolensky, M. Troyer, and K. M. Svore, "Quantum language processing," 2019, *arXiv:1902.05162*.

[41] A. Sordoni, J.-Y. Nie, and Y. Bengio, "Modeling term dependencies with quantum language models for IR," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2013, pp. 1–36.

[42] D. Widdows, A. Alexander, D. Zhu, C. Zimmerman, and A. Majumder, "Near-term advances in quantum natural language processing," *Ann. Math. Artif. Intell.*, Apr. 2024, doi: 10.1007/s10472-024-09940-y.

[43] J. Tian, X. Sun, Y. Du, S. Zhao, Q. Liu, K. Zhang, W. Yi, W. Huang, C. Wang, X. Wu, M.-H. Hsieh, T. Liu, W. Yang, and D. Tao, "Recent advances for quantum neural networks in generative learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 10, pp. 1–20, Nov. 2023, doi: 10.1109/TPAMI.2023.3272029.

[44] S. Clark, B. Coecke, and M. Sadrzadeh, "A compositional distributional model of meaning," in *Proc. 2nd Symp. Quantum Interact.*, 2008, pp. 133–140.

[45] B. Coecke, M. Sadrzadeh, and S. Clark, "Mathematical foundations for a compositional distributional model of meaning," 2010, *arXiv:1003.4394*.

[46] A. Panahi, S. Saeedi, and T. Arodz, "Word2ket: Space-efficient word embeddings inspired by quantum entanglement," 2019, *arXiv:1911.04975*.

[47] J. Pater, *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar*, vol. 1. Cambridge, MA, USA: MIT Press, May 2009, pp. 217–226.

[48] P. Smolensky, M. Goldrick, and D. Mathis, "Optimization and quantization in gradient symbol systems: A framework for integrating the continuous and the discrete in cognition," *Cognit. Sci.*, vol. 38, no. 6, pp. 1102–1138, Aug. 2014.

[49] J. Qi, C.-H.-H. Yang, P.-Y. Chen, and M.-H. Hsieh, "Theoretical error performance analysis for variational quantum circuit based functional regression," *Npj Quantum Inf.*, vol. 9, no. 1, p. 4, Jan. 2023.

[50] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 1–9.

[51] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 56–65.

[52] J. Lambek, *From Word to Sentence*. Montreal, QC, Canada: McGill Univ., 2008. [Online]. Available: https://www.math.mcgill.ca/barr/lambek/pdffiles/2008lambek.pdf

[53] K. M. Hermann and P. Blunsom, "The role of syntax in vector space models of compositional semantics," in *Proc. 51st Annu. Meeting Assoc. for Comput. Linguistics*, Aug. 2013, pp. 894–904.

[54] B. Coecke, G. de Felice, K. Meichanetzidis, and A. Toumi, "Foundations for near-term quantum natural language processing," 2020, *arXiv:2012.03755*.

[55] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *Social Netw. Comput. Sci.*, vol. 2, no. 3, p. 160, May 2021.

[56] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020.

[57] T. Zhang, *Mathematical Analysis of Machine Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2023.

[58] K. N. Satone, C. A. Dhawale, and P. B. Ulhe, "Convolution neural network: Architecture, applications, and recent trends," in *Scalable and Distributed Machine Learning and Deep Learning Patterns*. Hershey, PA, USA: IGI Global, 2023, pp. 123–146.

[59] M. Krichen, "Convolutional neural networks: A survey," *Computers*, vol. 12, no. 8, p. 151, Jul. 2023.

[60] M. M. Taye, "Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions," *Computation*, vol. 11, no. 3, p. 52, Mar. 2023.

[61] A. Orvieto, S. L. Smith, A. Gu, A. Fernando, C. Gulcehre, R. Pascanu, and S. De, "Resurrecting recurrent neural networks for long sequences," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 26670–26698.

[62] S. Cheng, "Recurrent neural networks," School ECE, Univ. Oklahoma, Norman, OK, USA, Tech. Rep., 2023.

[63] J. P. Bharadiya, "Exploring the use of recurrent neural networks for time series forecasting," *Int. J. Innov. Sci. Res. Technol.*, vol. 8, no. 5, pp. 2023–2027, 2023.

[64] F. Hu, "Survey on neural networks in natural language processing," in *Proc. IEEE Int. Conf. Control, Electron. Comput. Technol. (ICCECT)*, Apr. 2023, pp. 591–594.

[65] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2023, *arXiv.1512.03385*.

[66] A. B. Yadav, "Generative AI in the era of transformers: Revolutionizing natural language processing with LLMs," *J. Image Process. Intell. Remote Sens.*, vol. 4, no. 2, pp. 54–61, Feb./Mar. 2024.

[67] N. Patwardhan, S. Marrone, and C. Sansone, "Transformers in the real world: A survey on NLP applications," *Information*, vol. 14, no. 4, p. 242, Apr. 2023.

[68] P. Taneja, K. Chandra, D. Batra, A. Gupta, R. Kumar, and B. Tyagi. (2024). *Enhancing Neural Language Models: A Comprehensive Approach With Tensorized Transformer and Over-Parameterization*. [Online]. Available: https://vixra.org/abs/2402.0060

[69] A. Rahali and M. A. Akhloufi, "End-to-end transformer-based models in textual-based NLP," *AI*, vol. 4, no. 1, pp. 54–110, Jan. 2023.

[70] M.-G. Zhou, Z.-P. Liu, H.-L. Yin, C.-L. Li, T.-K. Xu, and Z.-B. Chen, "Quantum neural network for quantum neural computing," *Research*, vol. 6, p. 0134, Jan. 2023.

[71] A. Sagingalieva, M. Kordzanganeh, N. Kenbayev, D. Kosichkina, T. Tomashuk, and A. Melnikov, "Hybrid quantum neural network for drug response prediction," *Cancers*, vol. 15, no. 10, p. 2705, May 2023.

[72] J. S. Neto, L. Q. Ardila, T. N. Nogueira, F. Albuquerque, J. P. Papa, R. C. Guido, and F. Fernandes Fanchini, "Quantum neural networks successfully calibrate language models," *Quantum Mach. Intell.*, vol. 6, no. 1, p. 8, Jun. 2024.

[73] G. Li, X. Zhao, and X. Wang, "Quantum self-attention neural networks for text classification," *Sci. China Inf. Sci.*, vol. 67, no. 4, pp. 1–13, 2024.

[74] E. Farhi and H. Neven, "Classification with quantum neural networks on near term processors," 2018, *arXiv:1802.06002*.

[75] J. Shi, R.-X. Zhao, W. Wang, S. Zhang, and X. Li, "QSAN: A near-term achievable quantum self-attention network," 2022, *arXiv:2207.07563*.

[76] J. Allcock, C.-Y. Hsieh, I. Kerenidis, and S. Zhang, "Quantum algorithms for feedforward neural networks," *ACM Trans. Quantum Comput.*, vol. 1, no. 1, pp. 1–24, Dec. 2020.

[77] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, "Continuous-variable quantum neural networks," *Phys. Rev. Res.*, vol. 1, no. 3, Oct. 2019, Art. no. 033063.

[78] L. J. O'Riordan, M. Doyle, F. Baruffa, and V. Kannan, "A hybrid classical-quantum workflow for natural language processing," *Mach. Learning, Sci. Technol.*, vol. 2, no. 1, Dec. 2020, Art. no. 015011.

[79] C. Zhao and X.-S. Gao, "QDNN: DNN with quantum neural network layers," 2019, *arXiv:1912.12660*.

[80] A. K. M. Masum, A. Maurya, D. S. Murthy, and N. Mahmud, "Hybrid quantum-classical machine learning for sentiment analysis," 2023, *arXiv:2310.10672*.

[81] J. Yao, M. Bukov, and L. Lin, "Policy gradient based quantum approximate optimization algorithm," *Math. Sci. Mach. Learn.*, vol. 1, pp. 605–634, May 2020.

[82] M. Watabe, K. Shiba, M. Sogabe, K. Sakamoto, and T. Sogabe, "Quantum circuit parameters learning with gradient descent using backpropagation," 2019, *arXiv:1910.14266*.

[83] F. Barratt, J. Dborin, and L. Wright, "Improvements to gradient descent methods for quantum tensor network machine learning," 2022, *arXiv:2203.03366*.

[84] J. Romero, J. P. Olson, and A. Aspuru-Guzik, "Quantum autoencoders for efficient compression of quantum data," *Quantum Sci. Technol.*, vol. 2, no. 4, p. 045001, 2017.

[85] C. Bravo-Prieto, "Quantum autoencoders with enhanced data encoding," *Mach. Learning: Sci. Technol.*, vol. 2, no. 3, Sep. 2021, Art. no. 035028.

[86] S. Markidis, "Programming quantum neural networks on NISQ systems: An overview of technologies and methodologies," *Entropy*, vol. 25, no. 4, p. 694, Apr. 2023.

[87] E. Ardeshir-Larijani and M. M. Nasiri Fatmehsari, "Hybrid classical-quantum transfer learning for text classification," *Quantum Mach. Intell.*, vol. 6, no. 1, p. 19, Jun. 2024.

[88] D. Baronia, "Hybrid quantum-classical neural networks for text classification," in *Quantum Machine Intelligence*. Cham, Switzerland: Springer, 2023.

[89] P. Zhang, J. Niu, Z. Su, B. Wang, L. Ma, and D. Song, "End-to-end quantum-like language models with application to question answering," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2018, vol. 32, no. 1, pp. 1–26.

[90] C. H. Yang, J. Qi, S. Y. Chen, Y. Tsao, and P.-Y. Chen, "When BERT meets quantum temporal convolution learning for text classification in heterogeneous computing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 8602–8606.

[91] J. Qi and J. Tejedor, "Classical-To-Quantum transfer learning for spoken command recognition based on quantum neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 8627–8631.

[92] W. Zeng and B. Coecke, "Quantum algorithms for compositional natural language processing," *Electron. Proc. Theor. Comput. Sci.*, vol. 221, pp. 67–75, Aug. 2016.

[93] K. Meichanetzidis, S. Gogioso, G. de Felice, N. Chiappori, A. Toumi, and B. Coecke, "Quantum natural language processing on near-term quantum computers," *Electron. Proc. Theor. Comput. Sci.*, vol. 340, pp. 213–229, Sep. 2021.

[94] R. Lorenz, A. Pearson, K. Meichanetzidis, D. Kartsaklis, and B. Coecke, "QNLP in practice: Running compositional models of meaning on a quantum computer," *J. Artif. Intell. Res.*, vol. 76, pp. 1305–1342, Apr. 2023.

[95] K. Meichanetzidis, A. Toumi, G. de Felice, and B. Coecke, "Grammar-aware sentence classification on quantum computers," *Quantum Mach. Intell.*, vol. 5, no. 1, pp. 1–36, Jun. 2023.

[96] D. T. Chang, "Variational quantum classifiers for natural-language text," 2023, *arXiv:2303.02469*.

[97] M. Abbaszade, V. Salari, S. S. Mousavi, M. Zomorodi, and X. Zhou, "Application of quantum natural language processing for language translation," *IEEE Access*, vol. 9, pp. 130434–130448, 2021.

[98] V. Salari, M. Abbaszadeh, and S. S. Mousavi, "Parametrized quantum circuits of synonymous sentences in quantum natural language processing," 2021, *arXiv:2102.02204*.

[99] Q. Li, M. Melucci, and P. Tiwari, "Quantum language model-based query expansion," in *Proc. ACM SIGIR Int. Conf. Theory Inf. Retr.*, Sep. 2018, pp. 1–18.

[100] M. Xie, Y. Hou, P. Zhang, J. Li, W. Li, and D. Song, *Modeling Quantum Entanglements in Quantum Language Models*. Washington, DC, USa: AAAI Press, 2015, pp. 1362–1368.

[101] Y. Jiang, P. Zhang, H. Gao, and D. Song, "A quantum interference inspired neural matching model for ad-hoc retrieval," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 56–78.

[102] I. Basile and F. Tamburini, "Towards quantum language models," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2017, pp. 45–53.

[103] P. Zhang, Z. Su, L. Zhang, B. Wang, and D. Song, "A quantum many-body wave function inspired language modeling approach," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 1303–1312.

[104] Y. Zhang, D. Song, X. Li, P. Zhang, P. Wang, L. Rong, G. Yu, and B. Wang, "A quantum-like multimodal network framework for modeling interaction dynamics in multiparty conversational sentiment analysis," *Inf. Fusion*, vol. 62, pp. 14–31, Oct. 2020.

[105] E. R. Miranda, R. Yeung, A. Pearson, K. Meichanetzidis, and B. Coecke, *A Quantum Natural Language Processing Approach to Musical Intelligence*. Cham, Switzerland: Springer, 2022, pp. 313–356.

[106] H. Wazni, K. I. Lo, L. McPheat, and M. Sadrzadeh, "A quantum natural language processing approach to pronoun resolution," 2022, *arXiv:2208.05393*.

[107] A. Karamlou, J. Wootton, and M. Pfaffhauser, "Quantum natural language generation on near-term devices," in *Proc. 15th Int. Conf. Natural Lang. Gener.*, 2022, pp. 267–277.

[108] S. Ganguly, S. N. Morapakula, and L. M. P. Coronado, "Quantum natural language processing based sentiment analysis using lambeq toolkit," in *Proc. 2nd Int. Conf. Power, Control Comput. Technol. (ICPC2T)*, Mar. 2022, pp. 1–6.

[109] D. Kartsaklis, I. Fan, R. Yeung, A. Pearson, R. Lorenz, A. Toumi, G. de Felice, K. Meichanetzidis, S. Clark, and B. Coecke, "Lambeq: An efficient high-level Python library for quantum NLP," 2021, *arXiv:2110.04236*.

[110] S. Clark, B. Coecke, E. Grefenstette, S. Pulman, and M. Sadrzadeh, "A quantum teleportation inspired algorithm produces sentence meaning from word meaning and grammatical structure," 2013, *arXiv:1305.0556*.

[111] Y. Suzuki, Y. Kawase, Y. Masumura, Y. Hiraga, M. Nakadai, J. Chen, K. M. Nakanishi, K. Mitarai, R. Imai, S. Tamiya, T. Yamamoto, T. Yan, T. Kawakubo, Y. O. Nakagawa, Y. Ibe, Y. Zhang, H. Yamashita, H. Yoshimura, A. Hayashi, and K. Fujii, "Qulacs: A fast and versatile quantum circuit simulator for research purpose," *Quantum*, vol. 5, p. 559, Oct. 2021.

[112] J. S. Garofolo, "Timit acoustic phonetic continuous speech corpus," Linguistic Data Consortium, 1993, doi: 10.35111/17gk-bn40.

[113] L. Rimell, J. Maillard, T. Polajnar, and S. Clark, "RELPRON: A relative clause evaluation data set for compositional distributional semantics," *Comput. Linguistics*, vol. 42, no. 4, pp. 661–701, Dec. 2016.

[114] M. Wang, N. A. Smith, and T. Mitamura, "What is the jeopardy model? A quasi-synchronous grammar for QA," in *Proc. Joint Conf. Empirical Methods Natural Language Process. Comput. Natural Lang. Learn.*, 2007, pp. 22–32.

[115] Y. Yang, W.-T. Yih, and C. Meek, "WikiQA: A challenge dataset for open-domain question answering," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2015, pp. 2013–2018.

[116] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, M. Primet, and J. Dureau, "Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces," 2018, *arXiv:1805.10190*.

[117] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The atis spoken language systems pilot corpus," in *Speech and Natural Languag*. Springer, 1990.

[118] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," 2018, *arXiv:1804.03209*.

[119] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml
[120] S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea, "MELD: A multimodal multi-party dataset for emotion recognition in conversations," 2018, *arXiv:1810.02508*.
[121] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Lang. Resour. Eval.*, vol. 42, no. 4, pp. 335–359, Dec. 2008.

CHARLES M. VARMANTCHAONALA received the B.S. degree in mathematics and computer science and the M.S. degree in systems and software in distributed environments from the University of Ngaoundéré, Cameroon, in 2019 and 2021, respectively, where he is currently pursuing the Ph.D. degree in computer sciences, specialized in quantum computing. Since 2019, he has been a Mathematics and Computer Science Teacher at the Secondary School and a Teaching Assistant at the University of Ngaoundéré. His research interests include quantum computing, natural language processing, systems learning, data science, autonomous systems, and algorithms optimization.

JEAN LOUIS K. E. FENDJI received the B.S. and M.S. degrees in computer science from the University of Ngaoundéré, Cameroon, in 2010, and the Ph.D. degree from the University of Bremen, Germany, in 2015. Since 2022, he has been an Associate Professor with the Department of Computer Engineering, University Institute of Technology, University of Ngaoundéré. From 2024 to 2026, he is an Iso Lomso Fellow with the Stellenbosch Institute for Advanced Study, working on digital inclusion through oral languages. He has been working for more than ten years on network modeling and optimization. His current research interests include systems learning, deep learning, NLP, speech recognition, optimization, and AI in agriculture and education.

JULIUS SCHÖNING (Senior Member, IEEE) received the Ph.D. degree in computer vision from the Institute of Cognitive Science, Osnabrück University. He has studied mechatronics as well as intelligent embedded microsystems. He is currently a Full Professor with the Media Laboratory, Osnabrück University of Applied Sciences, focusing on AI-in-the-loop, which combines AI and HMI to create safe and trustworthy AI applications. Previously, he was a Systems Architect and a Researcher with ZF Friedrichshafen AG, focused on self-driving vehicles. Besides, he has working experience as a Software Requirement Engineer for HMIs.

MARCELLIN ATEMKENG received the Ph.D. degree in big data and statistical signal processing from Rhodes University, South Africa, in 2016. He was a Signal Analyst with VASTech, a telecommunication industry in Stellenbosch, South Africa. From 2017 to 2018, he was a Postdoctoral Research Fellow with the Square Kilometre Array-related projects. Since 2019, he has been a Senior Lecturer with the Department of Mathematics, Rhodes University, and a Coordinator of the Rhodes AI Research Group (RAIRG). He has published several articles in peer-reviewed journals. His research interests include big data, statistical signal processing, artificial intelligence, deep learning, and modern astronomical data processing. He was a recipient of the Kambule Doctoral Award, in 2019, for recognizing and encouraging excellence in research and writing by doctoral candidates at African universities in any area of computational and statistical sciences.