A Quantum Many-body Wave Function Inspired Language Modeling Approach

Peng Zhang[1], Zhan Su[1], Lipeng Zhang[2], Benyou Wang[3], Dawei Song[4] *
[1] School of Computer Science and Technology, Tianjin University, Tianjin, China
[2] School of Computer Software, Tianjin University, Tianjin, China
[3] Department of Information Engineering, University of Padova, Padova, Italy
[4] School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
{pzhang,suzhan,lpzhang}@tju.edu.cn
wang@dei.unipd.it;dawei.song2010@gmail.com

ABSTRACT

The recently proposed quantum language model (QLM) aimed at a principled approach to modeling term dependency by applying the quantum probability theory. The latest development for a more effective QLM has adopted word embeddings as a kind of global dependency information and integrated the quantum-inspired idea in a neural network architecture. While these quantum-inspired LMs are theoretically more general and also practically effective, they have two major limitations. First, they have not taken into account the interaction among words with multiple meanings, which is common and important in understanding natural language text. Second, the integration of the quantum-inspired LM with the neural network was mainly for effective training of parameters, yet lacking a theoretical foundation accounting for such integration. To address these two issues, in this paper, we propose a Quantum Many-body Wave Function (QMWF) inspired language modeling approach. The QMWF inspired LM can adopt the tensor product to model the aforesaid interaction among words. It also enables us to reveal the inherent necessity of using Convolutional Neural Network (CNN) in QMWF language modeling. Furthermore, our approach delivers a simple algorithm to represent and match text/sentence pairs. Systematic evaluation shows the effectiveness of the proposed QMWF-LM algorithm, in comparison with the state of the art quantum-inspired LMs and a couple of CNN-based methods, on three typical Question Answering (QA) datasets.

KEYWORDS

Language modeling, quantum many-body wave function, convolutional neural network

ACM Reference Format:
Peng Zhang[1], Zhan Su[1], Lipeng Zhang[2], Benyou Wang[3], Dawei Song[4] . 2018. A Quantum Many-body Wave Function Inspired Language Modeling Approach. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3269206.3271723

*Corresponding authors: P. Zhang and D. Song.

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3271723

1 INTRODUCTION

It is essential to model and represent a sequence of words for many Information Retrieval (IR) or Natural Language Processing (NLP) tasks. In general, Language Modeling (LM) approaches utilize probabilistic models to measure the *uncertainty* of a text (e.g., a document, a sentence, or some keywords). Based on different probability measures, there are roughly two different categories of LM approaches, namely traditional LMs [39] based on the classical probability theory, and quantum-inspired LMs [31, 40] motivated by the quantum probability theory, which can be considered as a generalization of the classical one [20, 30].

Recently, Sordoni, Nie and Bengio proposed a Quantum Language Modeling (QLM) approach, which aims to model the term dependency in a more principled manner [31]. In traditional LMs, modeling word dependency will increase the number of parameters to be estimated for compound dependencies (e.g., *n*-gram LM for IR) [28]), or involve computing additional scores from matching compound dependencies in the final ranking function (e.g., Markov Random Field based LM [21]). To solve these problems, QLM estimates a density matrix, which has a fixed dimensionality and encodes the probability measurement for both single words and compound words. In addition to its theoretical benefits, QLM has been applied to ad-hoc information retrieval task and achieved effective performance.

In order to further improve the practicality of the quantum language models, a Neural Network based Quantum-like Language Model (NNQLM) was proposed [40]. NNQLM utilizes word embedding vectors [22] as the state vectors, based on which a density matrix can be directly derived and integrated into an end-to-end Neural Network (NN) structure. NNQLM has been effectively applied in a Question Answering (QA) task. In NNQLM, a joint representation based on the density matrices can encode the similarity information of each question-answer pair. A Convolutional Neural Network (CNN) architecture is adopted to extract useful similarity features from such a joint representation and shows a significant improvement over the original QLM on the QA task.

Despite the progress in the quantum-inspired LMs from both theoretical and practical perspectives, there are still two major limitations, in terms of the representation capacity and seamless integration with neural networks. First, both QLM and NNQLM have

not modeled the *complex interaction* among words with multiple meanings. For example, suppose we have two polysemous words A and B, in the sense that A has two meanings A_1 and A_2, while B has two meanings B_1 and B_2. If we put them together and form a compound word, this compound word will have four possible states (A_1B_1, A_1B_2, A_2B_1, A_2B_2), each corresponding to a combination of specific meanings of different words. If we have more words, such an interaction will become more complex. However, in QLM and NNQLM, a compound word is modeled as a direct addition of the representation vectors or subspaces of the single words involved. Therefore, it is challenging to build a language modeling mechanism which has the representation capacity towards the complex interactions among words as described above.

Second, although in NNQLM the neural network structure can help quantum-inspired LM with effective training, the fundamental connection between the quantum-inspired LM and the neural network remains unclear. In other words, the integration of NN and QLM so far has not been in a principled manner. Hence, we need to investigate and explain the intrinsic rationality of neural network in quantum-inspired LM. It is challenging, yet important to bridge quantum-inspired idea, language modeling, and neural network structure together, and develop a novel LM approach with both theoretical soundness and practical effectiveness.

In order to address the above two challenges, we propose a new language modeling framework inspired by Quantum Many-body Wave Function (QMWF). In quantum mechanics, the wave function can model the interaction among many spinful particles (or electrons), where each particle is laying on multiple states simultaneously, and each state corresponds to a basis vector [6, 23]. Therefore, by considering a word as a particle, different meanings (or latent/embedded concepts) as different basis vectors, the interaction among words can be modeled by the tensor product of different basis vectors for different words. It is then natural to use such a QMWF formalism to represent the complex interaction system for a sequence of natural language words.

In addition, we show that the convolutional neural network architecture can be mathematically derived in our quantum-inspired language modeling approach. Since the tensor product is performed in QMWF based LM, the dimensionality of the tensor will be exponentially increased, yielding a quantum many-body problem. To solve this problem, the tensor decomposition can be used to solve a high-dimensional tensor [18]. With the help of tensor decomposition, the projection of the global representation to the local representation of a word sequence can result in a convolutional neural network architecture. In turn, for the convolutional neural network, it can be interpreted as a mapping form a global semantic space to the local semantic space.

Hence, our QMWF inspired Language Modeling (QMWF-LM) approach also delivers a feasible and simple algorithm to represent a text or a sentence and match the text/sentence pairs, in both word-lever and character-level. We implement our approach in the Question Answering task. The experiments have shown that the proposed QMWF-LM can not only outperform its quantum LM counterparts (i.e., QLM and NNQLM), but also reaches comparable or even better performance with the CNN-based approaches on three typical QA datasets.

Our main contributions can be summarized as follows:

(1) We propose a Quantum Many-body Wave Function based Language Modeling (QMWF-LM) approach, which is able to represent complex interaction among words, each with multiple semantic basis vectors (for multiple meanings/concepts).
(2) We show a fundamental connection between QMWF based language modeling approach and the convolutional neural network architecture, in terms of the projection between the global representation to the local one of a word sequence.
(3) The proposed QMWF-LM delivers an efficient algorithm to represent and match the text/sentence pairs, in both word-lever and character-level, as well as achieves effective performance on a number of QA datasets.

2 QUANTUM PRELIMINARIES

In this section, we first describe some basis notations and concepts of the quantum probability theory. Then, we briefly explain the quantum many-body wave function.

2.1 Basic Notations and Concepts

The formalism of quantum theory is actually based on vector spaces using Dirac notations. In line with previous studies on the quantum-inspired language models [29, 31, 40], we restrict our problem to vectors spaces over real numbers in $\mathbb{R}$.

A wave function in quantum theory is a mathematical description of the quantum state of a system. A state vector is denoted by a unit vector $|\psi\rangle$ (called as a *ket*), which can be considered as a column vector $\vec{\psi} \in \mathbb{R}^n$ (for better understanding). The transpose of $|\psi\rangle$ is denoted as $\langle\psi|$ (called as *bra*), which is a row vector.

The state vector can be considered as a *ray* in a Hilbert space (i.e., $|\psi\rangle \in \mathcal{H}$), which has a set of orthonormal basis vectors $|e_i\rangle$ ($i = 1, \ldots, n$). A state vector $|\psi\rangle$ can be a superposition of the basis vectors:

$$|\psi\rangle = \sum_{i=1}^{n} a_i |e_i\rangle \tag{1}$$

where a_i is a probability amplitude and $\sum_i a_i^2 = 1$, since a_i^2 represents a probability of the sum to 1.

For example, suppose we have a two basis vectors $|0\rangle$ and $|1\rangle$, which can be considered as $(1, 0)^T$ and $(0, 1)^T$, respectively. Then, we have a state vector

$$|\psi\rangle = a_1 |0\rangle + a_2 |1\rangle$$

It means that the corresponding quantum system $|\psi\rangle$ is a *superposed state*, i.e., it is in the two states $|0\rangle$ and $|1\rangle$ simultaneously. In the natural language processing tasks, such a superposed state can be used to model the multiple semantic meanings of a word [3].

In Eq. 1, the probability amplitude a_i can be calculated by the *inner product* $\langle e_i|\psi\rangle$.

$$a_i = \langle e_i|\psi\rangle$$

The inner product $\langle e_i|\psi\rangle$ is a projection of $|\psi\rangle$ onto $|e_i\rangle$. As illustrated in Fig.1, the projection measurement can be formulated as

$$p(e_i|\psi) = a_i^2 = \langle e_i|\psi\rangle^2 \tag{2}$$

where $p(e_i|\psi)$ denotes the probability of the quantum elementary event $|e_i\rangle$ [1] given the system $|\psi\rangle$.

[1] More strictly, the outer product of $|e_i\rangle$ is called as the quantum elementary event.

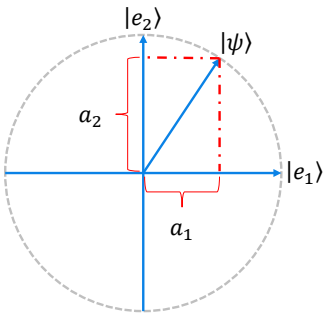

Figure 1: Projection of $|\psi\rangle$ on its basis

It turns out that the projection measurement based on inner product plays an essential role in the probability measurement, We will further illustrate such a concept in our quantum many-body wave function inspired LM approach. Note that, in a broad sense, the *wave function* is a state vector $|\psi\rangle$. In a narrow sense, the wave function is a projection on a basis, e.g., $\psi(x) = \langle x|\psi\rangle$, where x can be a basis e_i, and $\psi(x)$ is the probability amplitude. In this paper, we will use the description of wave function in the broad sense.

2.2 Quantum Many-Body Wave Functions

What we mentioned above is a single system which corresponds to a single particle in a Hilbert space.

A quantum many-body system consists of N particles, each one with a wave function residing in a finite dimensional Hilbert space $\mathcal{H}_i$ for $i \in [N] := \{1 \dots N\}$. We set the dimensions of each Hilbert space $\mathcal{H}_i$ for all i, i.e., $\forall i : dim(\mathcal{H}_i) = M$ and the orthonormal basis of the Hilbert space as $\{|e_h\rangle\}_{h=1}^{M}$. The Hilbert space of a many-body system is a *tensor product* of the spaces: $\mathcal{H} := \otimes_{i=1}^{N} \mathcal{H}_i$, and the corresponding state vector $|\psi\rangle \in \mathcal{H}$ is

$$|\psi\rangle = \sum_{h_1,\dots,h_N=1}^{M} \mathcal{A}_{h_1 \dots h_N} |e_{h_1}\rangle \otimes \cdots \otimes |e_{h_N}\rangle \tag{3}$$

where $|e_{h_1}\rangle \otimes \cdots \otimes |e_{h_N}\rangle$ is a basis vector of the M^N dimensional Hilbert space $\mathcal{H}$, and $\mathcal{A}_{h_1 \dots h_N}$ is a specific entry in a tensor $\mathcal{A}$ holding all the probability amplitude. A tensor $\mathcal{A}$ can be considered as N-dimensional array $\mathcal{A} \in \mathbb{R}^{M \times \cdots \times M}$.

For example, a system includes two spinful particles, which are qubit states superposed as two basis vectors $|e_1\rangle = |0\rangle = (1,0)^T$ and $|e_2\rangle = |1\rangle = (0,1)^T$. Therefore, we can get four basis vectors $|e_1\rangle \otimes |e_1\rangle = |00\rangle$ (abbreviation of $|0\rangle \otimes |0\rangle = (1,0,0,0)^T$), $|01\rangle = (0,1,0,0)^T$, $|10\rangle = (0,0,1,0)^T$ and $|11\rangle = (0,0,0,1)^T$, and the state ψ of this system can be represented as

$$\begin{aligned} |\psi\rangle &= \sum_{i,j=1}^{2} a_{ij} |e_i\rangle \otimes |e_j\rangle \\ &= a_{11}|00\rangle + a_{12}|01\rangle + a_{21}|10\rangle + a_{22}|11\rangle \end{aligned} \tag{4}$$

where a_{ij} is the probability amplitude and $\sum_{ij} a_{ij}^2 = 1$. Each a_{ij} can be considered as a specific entry in a tensor $\mathcal{A} \in \mathbb{R}^{2\times 2}$.

3 QUANTUM MANY-BODY WAVE FUNCTION INSPIRED LANGUAGE MODELING

3.1 Basic Intuitions and Architecture

In Physics, Quantum Many-body Wave Function (QMWF) can model the interaction among many particles and the associated basis vectors. In the language scenario, by considering a word as a particle, different meanings (or latent/embedded concepts) as different basis vectors, the **interaction** among words (or word meanings) can be modeled by the **tensor product** of basis vectors, via the many-body wave function. The tensor product of different basis vectors generate *compound meanings* for compound words.

Based on such an analogy, **QMWF representation** can model the *probability distribution* of *compound meanings* in natural language. The representation of **compound meanings** depends on **basis vectors**. The choices of basis vectors can be one-hot vectors (representing single words), or embedding vectors (representing latent concepts). The **probabilities** are encodes in a **tensor**, as we can see from Eq.3. Each entry in a tensor is the probability amplitude of the compound meaning, or can be considered as a coefficient/weight.

As shown in Fig. 2, given a word sequence and the basis vectors, there are **local** and **global representations** (see details in Section 3.2). Intuitively, the local representation is constructed by the current word sequence (e.g., a sentence), and the global representation corresponds to the information of a large corpora (e.g., a collection of data). In classical language modeling approaches, there are also local and global information, as well as the interplay between them. For example, in n-grams, the probability/statistics of each term can be estimated from the current piece of text (as local information), and also be smoothed with the statistics from a large corpora (as global information).

Based on QMWF representation, in Section 3.2.3, we describe the **projection** from the global representation to the local one. Such projection can model the interplay between the local information and the global one, and enable us to focus on the high-dimensional tensors, which encode the probability distribution of the compound meanings. In Fig. 2, we can observe that, the high-dimensional tensor $\mathcal{T}$ can be reduced by the tensor decomposition, the tensors $\mathcal{A}$ (for probabilities in local representation) and $\mathcal{T}$ (for probabilities in global representation) are kept. Therefore, the projection can also be considered as an interplay between global and local tensors (see Section 3.2.3 for details).

The high-dimensional tensor $\mathcal{T}$ can be reduced by the **tensor decomposition**. With the help of the tensor decomposition, the above projection from the global representation (as a global semantic space) to the local one can be realized by a **convolutional neural network** architecture(see Section 3.3). Intuitively, each decomposed subspace of the high-dimensional tensor corresponds to a convolutional channel. Together with a product pooling technique, a CNN architecture can be constructed. Then, an algorithm based on a CNN architecture is revealed based on the above intuitions.

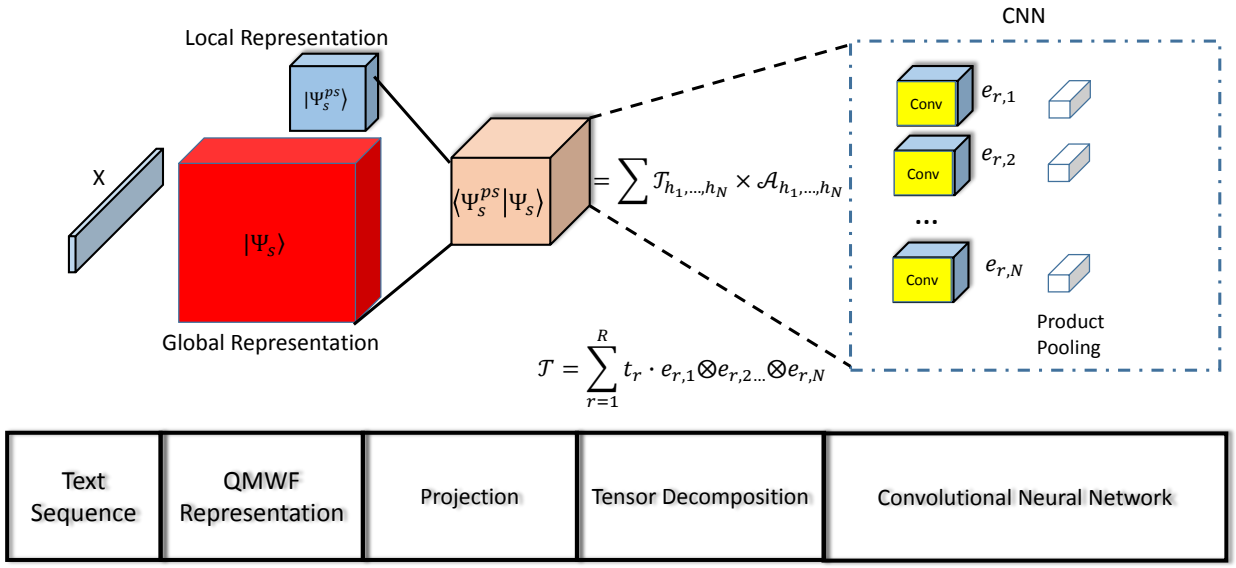

Figure 2: Outline of quantum many-body wave function inspired language modeling approach

3.2 Language Representation and Projection via Many-body Wave Function

3.2.1 Local Representation by Product State. Suppose we have a word sequence S (e.g., a sentence) with the length N: $S = [x_1, x_2, ..., x_N]$. For each word x_i in S, based on Eq. 1, we define its state vector as:

$$|x_i\rangle = \sum_{h_i=1}^{M} \alpha_{i,h_i} |\phi_{h_i}\rangle \tag{5}$$

where the basis vectors are $|\phi_{h_i}\rangle$ ($h_i = 1, \ldots, M$). α_{i,h_i} is the corresponding amplitudes. Different from the notation α_i in Eq. 1, the notation α_{i,h_i} in Eq.5 is for the convenience to be represented in a tensor depicted latter. For a better understanding, as an example, the state vectors for words x_1 and x_2 can be represented by

$$\begin{cases} |x_1\rangle = \alpha_{1,1} |\phi_1\rangle + \alpha_{1,2} |\phi_2\rangle + \alpha_{1,3} |\phi_3\rangle \\ |x_2\rangle = \alpha_{2,1} |\phi_1\rangle + \alpha_{2,2} |\phi_2\rangle + \alpha_{2,3} |\phi_3\rangle \end{cases} \tag{6}$$

where $N = 2$, $M = 3$ and h_i is from 1 to 3, i.e., three basis vectors $|\phi_{h_i}\rangle$ are involved, each corresponding to a word meaning.

For the basis vectors $|\phi_{h_i}\rangle$, there can be different choices, e.g., one-hot vectors or embedded vectors. Different basis vectors yield different interpretations for the semantic meanings. We will adopt the embedding space when we instantiate this framework in the question answering task (see Section 4). If we use such a space, the probability amplitude α_{i,h_i} is the feature value (after normalization) of the word x_i on the $h_i - th$ dimension of the embedding space.

Next, we show how to use the tensor product to model the interaction among word meanings. For a sentence $S = [x_1, x_2, ..., x_N]$, its wave function can be represented as:

$$|\psi_S^{ps}\rangle = |x_1\rangle \otimes \ldots \otimes |x_N\rangle \tag{7}$$

where $|\psi_S^{ps}\rangle$ is the *product state* of the QMWF representation of a sentence. We can expand the product state $|\psi_S^{ps}\rangle$ as follows:

$$|\psi_S^{ps}\rangle = \sum_{h_1,\ldots,h_N=1}^{M} \mathcal{A}_{h_1 \ldots h_N} |\phi_{h_1}\rangle \otimes \ldots \otimes |\phi_{h_N}\rangle \tag{8}$$

where $|\phi_{h_1}\rangle \otimes \ldots \otimes |\phi_{h_N}\rangle$ is the new basis vectors with M^N dimension, and each new basis vector corresponds a *compound meanings* by the tensor product of the word meanings $|\phi_{h_i}\rangle$. $\mathcal{A}$ is a M^N dimensional tensor and each entry $\mathcal{A}_{h_1 \ldots h_N}$ ($= \prod_{i=1}^{N} \alpha_{i,h_i}$) encodes the probability of the corresponding compound meaning.

Eq. 8 can represent the interaction among words as we discussed in Introduction. For example, for two words x_1 and x_2 in Eq. 6, suppose x_1 only has two meanings corresponding to the basis vectors $|\phi_1\rangle$ and $|\phi_2\rangle$, while x_2 has two meanings corresponding to $|\phi_2\rangle$ and $|\phi_3\rangle$. Then, $\mathcal{A}_{1,3} (= \alpha_{1,1}\alpha_{2,3})$ represents the probability with the basis vector $|\phi_1\rangle \otimes |\phi_3\rangle$. Intuitively, this implies that the underlying meaning ($|\phi_1\rangle$) of word x_1 and the meaning ($|\phi_3\rangle$) of x_2 is interacted and form a *compound meaning* $|\phi_1\rangle \otimes |\phi_3\rangle$.

Now, we can see that this product state representation is actually a local representation for a word sequence. In other words, given the basis vectors, the probability amplitudes can be estimated from the current word sequence. In fact, $\mathcal{A}$ is a *rank-1* N-order tensor, which includes only $M \times N$ free parameters α_{i,h_i}, rather than M^N parameters to be estimated. In addition, given only a word sequence, the valid compound meanings are not too many. In summary, this rank-1 tensor actually encodes the local distributions of these compound meanings for the given sentence.

3.2.2 Global Representation for All Possible Compound Meanings. As aforementioned in Section 3.1 , we need a global distribution of all the possible compound meanings, given a set of basis vectors. Intuitively, a global distribution is useful in both classical

LM and quantum-inspired LM, since we often have *unseen* words, word meanings or the compound meanings, in a text.

To represent such a global distribution of state vectors, we define a quantum many-body wave function as follows:

$$|\psi_S\rangle = \sum_{h_1,\ldots,h_N=1}^{M} \mathcal{T}_{h_1\ldots h_N} |\phi_{h_1}\rangle \otimes \ldots \otimes |\phi_{h_N}\rangle \tag{9}$$

where $|\phi_{h_1}\rangle \otimes \ldots \otimes |\phi_{h_N}\rangle$ is the basis state (corresponding to a compound meaning) with M^N dimension, and $\mathcal{T}_{h_1\ldots h_N}$ is the corresponding probability amplitude. This wave function represents a semantic meaning space with a sequence of N uncertain words, which does not rely on a specific sentence showed in Eq. 8. The probability amplitudes in tensor $\mathcal{T}$ can be trained in a large corpora.

The difference between $|\psi_S^{ps}\rangle$ in Eq. 8 and $|\psi_S\rangle$ in Eq. 9 is the different tensors $\mathcal{A}$ and $\mathcal{T}$. $\mathcal{A}$ encodes the *local* distribution of compound meanings (for the current sentence) and $\mathcal{T}$ encodes the *global* distribution (for a large corpora). Moreover, $\mathcal{A}$ is essentially rank-1, while $\mathcal{T}$ has a higher rank. In fact, solving $\mathcal{T}$ is an intractable problem which is referred as a quantum many-body problem.

3.2.3 Projection from Global to Local Representation. Section 2.1 has emphasized the role of projection in the probability measurement. Now, we show the projection of the global semantic representation $|\psi_S\rangle$ on its product state $|\psi_S^{ps}\rangle$ as a local representation for the given sentence, to calculate the probability amplitudes in the tensor.

Such a projection can be modeled by the inner product $\langle\psi_S^{ps}|\psi_S\rangle$. Inspired by a recent work [18], this projection will eliminate the high-dimensional basis vectors of the wave function:

$$\begin{aligned}
\langle\psi_S^{ps}|\psi_S\rangle &= \langle x_1 \ldots x_N| \sum_{h_1,\ldots,h_N=1}^{M} \mathcal{T}_{h_1\ldots h_N} |\phi_{h_1}\ldots\phi_{h_N}\rangle\rangle \\
&= \sum_{h_1,\ldots,h_N=1}^{M} \mathcal{T}_{h_1\ldots h_N} \prod_{i=1}^{N} \langle x_i|\phi_{h_i}\rangle_i \\
&= \sum_{h_1,\ldots,h_N=1}^{M} \mathcal{T}_{h_1\ldots h_N} \prod_{i=1}^{N} \alpha_{i,h_i} \\
&= \sum_{h_1,\ldots,h_N=1}^{M} \mathcal{T}_{h_1\ldots h_N} \times \mathcal{A}_{h_1\ldots h_N}
\end{aligned} \tag{10}$$

which reveals the interplay between the global tensor $\mathcal{T}$ and local tensor $\mathcal{A}$. This is similar to the idea in the classical LM, where the local statistics in a text will be smoothed by collection statistics.

3.3 Projection Realized by Convolutional Neural Network

As shown in Eq. 10, the high-dimensional tensor $\mathcal{T}$ is still an unsolved issue. Now, we first describe the tensor decomposition to solve this high-dimensional tensor. Then, with the decomposed vectors, we will show that the convolutional neural network can be considered as a projection or a mapping process from the global semantics to local ones.

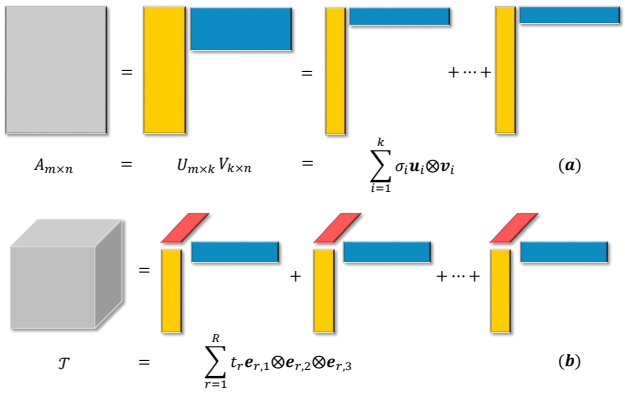

Figure 3: An illustration of the singular value decomposition of a matrix with dimension $M \times N$ in (a) and rank K and the CP decomposition of a three order tensor in (b).

3.3.1 Tensor Decomposition. In general, *Tensor decomposition* can be regarded as a generalization of Singular Value Decomposition (SVD) from matrices to tensors and can help to solve high-dimensional problems (see Fig. 3). There are many methods to decompose a high-dimensional tensor, such as Canonical Polyadic Decomposition (CP decomposition [13]), Tucker Decomposition, etc. The CP decomposition with weights is:

$$\mathcal{T} = \sum_{r=1}^{R} t_r \cdot \boldsymbol{e}_{r,1} \otimes \boldsymbol{e}_{r,2} \otimes \ldots \otimes \boldsymbol{e}_{r,N} \tag{11}$$

where t_r is the weight coefficient for each rank-1 tensor and $\boldsymbol{e}_{r,i} = (e_{r,i,1},\ldots,e_{r,i,M})^T$ $(i = 1,\ldots,N)$ is a unit vector with M-dimension. R is the rank of $\mathcal{T}$, which is defined as the smallest number of rank-one tensors that generate $\mathcal{T}$ as their sum.

The decomposed vector $\boldsymbol{e}_{r,i}$ with a low dimension will play a key role in the later derivation. A set of vectors $\boldsymbol{e}_{r,i}(i = 1,\ldots,N)$ can be a subspace of the high-dimensional tensor $\mathcal{T}$.

3.3.2 Towards Convolutional Neural Network. We will show that the projection from the global representation $|\psi_S\rangle$ to the local one $|\psi_S^{ps}\rangle$ can be realized by a Convolutional Neural Network (CNN) with product pooling [7]. To see this, we can put the CP decomposition Eq. 11 of $\mathcal{T}$ in Eq. 10, and obtain:

$$\langle\psi_S^{ps}|\psi_S\rangle = \sum_{r=1}^{R} t_r \prod_{i=1}^{N} \Big(\sum_{h_i=1}^{M} e_{r,i,h_i} \cdot \alpha_{i,h_i}\Big) \tag{12}$$

The above equation provides a connection between the quantum-inspired LM and the CNN design. The CNN interpretations of Eq. 12 are summarized in Table 1 and also illustrated in Fig. 4.

Given a sequence with N words, each is represented by an vector $\boldsymbol{x}_i = (\alpha_{i,h_1},\ldots,\alpha_{i,h_M})^T$. The convolution function is $\sum_{h_i=1}^{M} e_{r,i,h_i} \cdot \alpha_{i,h_i}$, which is the inner product $< \boldsymbol{x}_i, \boldsymbol{e}_{r,i} >$ between $\boldsymbol{x}_i$ and $\boldsymbol{e}_{r,i}$. The input vector $\boldsymbol{x}_i$ is a kind of local information and its entries α_{i,h_i} actually are the values in the local tensor $\mathcal{A}$. The entries in the vector $\boldsymbol{e}_{r,i}$ decomposed from the global tensor $\mathcal{T}$, are now parameters to be trained in the convolutional layer. Such an inner product, can be considered as a mapping from the global information $\boldsymbol{e}_{r,i}$ to

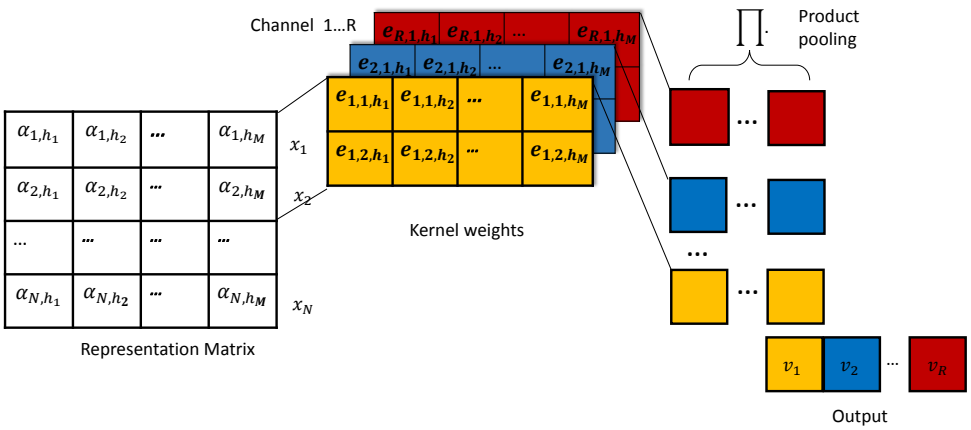

Figure 4: Realization of QMWF-LM via convolution neural network with product pooling

Table 1: CNN Interpretation of Projection

Input	$\boldsymbol{x}_i = (\alpha_{i,h_1}, \ldots, \alpha_{i,h_M})^T$
Convolution	$\Sigma_{r,i} = \sum_{h_i=1}^{M} e_{r,i,h_i} \cdot \alpha_{i,h_i}$
Product pooling	$\Pi_r = \prod_{i=1}^{N} \Sigma_{r,i}$
Output	$\sum_{r=1}^{R} t_r \cdot \Pi_r$

the local representation $\boldsymbol{x}_i$. After that, the product pooling layer (i.e., $\prod_r$, see Table 1) multiplies all the mapping results $\Sigma_{r,i} = <\boldsymbol{x}_i, \boldsymbol{e}_{r,i}>$ for all the N words.

As mentioned above, a set of N decomposed vectors corresponds to a subspace of the high-dimensional tensor. The rank R is the number of decomposed subspace, and this number corresponds to the number of convolution channels. In Fig. 4, different color means different channels of convolution. Following the input layer, a convolution layer with R channels calculates weighted sums of the representation vectors $\boldsymbol{x}_i$ and the vectors $\boldsymbol{e}_{r,i}$ (as kernel weights). In Eq. 12, one can sum R products Π_r with weights t_r to obtain the projection $\langle \psi_S^{ps} | \psi_S \rangle$. Then, a vector $\boldsymbol{v} = (v_1, \ldots, v_R)^T$ can be used to represent a sequence of words, where $v_r = t_r \cdot \Pi_r$ $(r = 1, \ldots, R)$.

It's worth noting that the decomposition of a *symmetric* tensor can make the unit vectors $\boldsymbol{e}_r$ as same as each order, which means for a convolutional kernel, $\boldsymbol{e}_{r,1} = \boldsymbol{e}_{r,2} = \ldots = \boldsymbol{e}_{r,N}$. In this way, we will get a property about convolutional neural networks, i.e., the weight sharing.

3.3.3 Algorithm. Based on the above ideas, a practical algorithm can be obtained with four parts as follows, also shown in Fig. 4:

- **Input layer**
  The input to our model, a sequence of words S, is composed of N words or patches $[x_1, ..., x_N]$. Each word x_i will be represented by a vector $(\alpha_{i,h_1}, ..., \alpha_{i,h_M})^T$. Then, we will get a representation matrix $\mathbf{S} \in \mathbb{R}^{N \times M}$.
- **Convolution layer** For each word or patch, the convolution is computed as follows: $\Sigma_{r,i} = \sum_{h_i=1}^{M} e_{r,i,h_i} \cdot \alpha_{i,h_i}$ $(r = 1, \ldots, R)$, where R is the number of convolution channels.
- **Product pooling layer** We apply the product pooling on the results of the convolution. It multiplies a number (N) of $\Sigma_{r,i}$ to get the $\Pi_r = \prod_{i=1}^{N} \Sigma_{r,i}$, where $\Pi_r \in \mathbb{R}$.
- **Output** Finally, we represent a sequence S using a vector $\boldsymbol{v} = (v_1, \ldots, v_R)^T \in \mathbb{R}^R$.

We can utilize the above algorithm to model the sentence representation and apply it to natural language processing tasks such as classification or matching between sentence pairs.

4 APPLICATIONS

Question Answering (QA) tasks aim to rank the answers from a candidate answer pool, given a question. The ranking is based on the matching scores between question and answer sentences. The key points are to build effective representations of the question and answer sentences and measure their matching score over such representations. In this paper, we model the question and answer sentences with quantum many-body wave functions and apply the algorithm in Section 3.3.3 to obtain question and answer sentences and match pairs of them.

Compared with the ad-hoc retrieval task, the question in the QA task is usually a piece of fluent natural language instead of a phrase or multiple keywords. The candidate answers are also shorter than the documents in ad-hoc retrieval. There is often less number of overlapping words between the question and answer sentences in the QA task, where semantic matching via neural network is widely used. In this paper, we apply the proposed QMWF based LM with neural network implementation in QA tasks. It should be noted that our model is also applicable to other LM based ranking tasks.

4.1 Quantum Many-Body Representation

As introduced in Section 3.2, each word vector $|x\rangle$ locates at the M dimensional Hilbert space. The product state representation of a specific sentence is represented by the wave function in Eq. 7, and the global representation of an arbitrary sentence with the same length is using the wave function in Eq. 9. As introduced in Section 3.2.3, the projection onto the product state of a sentence is formulated in Eq.10. The process of projection can be implemented

by a convolution and a product pooling which has been introduced in Section 3.3.2. As a text has different granularity, we can utilize two kinds of methods to obtain our input matrix S.

Word-level Based Input. We can utilize the expressive ability of pre-trained embedding. We think each dimension of word embedding is corresponding to a latent concept and a basis vector. Let α_{i,h_i} be the coefficient with respect to the corresponding basis vector, reflecting that the words can reside in specific concept with a certain probability. Given a sentence $S = [x_1, \ldots, x_N]$, where x_i is a single word represented by a vector $(\alpha_{i,1}, \ldots, \alpha_{i,M})^T$. Then, this sentence can be represented as a matrix in $\mathbf{S} \in \mathbb{R}^{N \times M}$.

Character-level Based Input. Inspired by the work [16] that using character-level convolution. A sentence is represented by a sequence of characters: $S^c = [x_1, ..., x_{\overline{N}}]$. The first part is a look-up table. We utilize the CNN with max pooling to obtain the representation in word level. We define the matrix $\mathcal{Z} = [z_1, ..., z_{\overline{N}}]$ as a input matrix where each column contains a vector $z_m \in R^{dk}$ that is the concatenation of a sequence of k char embeddings. $N = \overline{N} - k + 1$, d is the dimension of char embedding and k is the window size of convolution. Each output of convolution is $(\alpha_{i,h_1}, \ldots, \alpha_{i,h_M})^T$. Then, we can obtain the input matrix $\mathbf{S} \in \mathbb{R}^{N \times M}$.

Based on word-level and character-level, a representation matrix S can be obtained. Then, a sentence is represented by $\boldsymbol{v} = (v_1, \ldots, v_R)^T \in \mathbb{R}^R$ based on the algorithm in Section 3.3.3.

4.2 Matching in Many-body Wave Function Representation on QA

Let $Q = [q_1, q_2 \ldots q_{N_Q}]$ and $A = [a_1, a_2 \ldots a_{N_A}]$ be the sentence of the question and answer, respectively. As introduced above, we have:

$$v_i^q = t_r \cdot \prod_{i=1}^{N} \left(\sum_{h_i=1}^{M} e_{r,i,h_i} \cdot \alpha_{i,h_i}^q \right) \tag{13}$$

$$v_i^a = t_r \cdot \prod_{i=1}^{N} \left(\sum_{h_i=1}^{M} e_{r,i,h_i} \cdot \alpha_{i,h_i}^a \right) \tag{14}$$

Then, we have $\boldsymbol{v}^q = (v_1^q, \ldots, v_R^q)^T$ and $\boldsymbol{v}^a = (v_1^a, \ldots, v_R^a)^T$ for the vector representation of the question and answer, respectively. The matching score is defined as the *projection* from the answer state to the question state, which is an inner product $\langle \boldsymbol{v}^q, \boldsymbol{v}^a \rangle$.

5 LITERATURE REVIEW

Quantum theory is one of the most remarkable discoveries in Science. It not only can explain the behavior and movement of microscopic particles or electrons but also have been widely applied in the macro-world problems. For instance, the quantum theory has been applied in social science and economics [12], cognition and decision making [4, 5], language model [1], natural language processing [1, 3], and information retrieval [31, 33, 40].These research directions are making use of the mathematical formulation and non-classical probability measurement of quantum theory, rather than for quantum computation.

In Information Retrieval (IR), van Rijsbergen for the first time proposed to adopt the mathematical formalism of quantum theory to unify various IR formal models and provide a theoretical foundation for developing new models [33]. Later, a number of research efforts have been made to build quantum-like IR models [25, 31, 41, 42]. The main inspiration is rooted on the quantum theory as a principled framework for manipulating vector spaces and measuring probability in *Hilbert space* [20]. Piwowarski et al. [25] proposed a quantum IR framework, which represents the queries and documents as density matrices and subspaces, respectively. The information need space can be constructed by the tensor product of term vectors, based on a so-called multi-part system, which corresponds to a *product state* (see Section 3) of the quantum many-body wave function. However, the issue of the probability amplitudes (forming a high dimensional tensor) have not been addressed systematically. In addition, this framework has not shown the effect of tensor decomposition and its connection with the neural network design.

Some quantum-inspired retrieval models are based on the *analogies* between IR problems and quantum phenomena. For instance, by considering the inter-document dependency as a kind of quantum interference phenomena, a Quantum Probability Ranking Principle was proposed [42]. In addition, a quantum-inspired re-ranking method was developed by considering the ranking problem as a filtering process in the photon polarization [41]. These models are novel in term of their quantum-inspired intuitions. In practice, they adopted the relevance scores of classical retrieval models as the input probabilities, without actually performing a quantum probability measurement (e.g., the projection measurement).

Recently, Sordoni et al. [31] proposed a principled Quantum Language Model (QLM), which generalizes the traditional statistical LM with the quantum probability theory. In QLM, the probability uncertainties of both single and compound words are measured by the *projection measurement* in Hilbert space. For a text (a query or a document), a density matrix is then estimated based on a Maximal Likelihood Estimation (MLE) solution. Practically, QLM shows an effective performance on the ad-hoc retrieval task. Extending QLM with the idea of quantum entropy minimization, Sordoni et al. proposed to learn latent concept embeddings for query expansion in a supervised way [29]. In addition, to capture the dynamic information need in search sessions, an adaptive QLM was built with an evolution process of the density matrix [19]. More recently, an End-to-End Quantum-like Language Model (named as NNQLM) [40] has been proposed, which built a quantum-like LM into a neural network architecture and showed a good performance on QA tasks.

In this paper, we aim to tackle these challenging problems (as identified in the Introduction) of the existing quantum-inspired LMs. Our work is inspired by the recent multidisciplinary research findings across *quantum mechanics* and *machine learning* [2] (especially neural network [6, 18]). The two different disciplines, while seemingly to have huge gaps at the first glance, can benefit each other based on rigorous mathematical analysis and proofs. For instance, the neural network can help yield a more efficient solution for the quantum many-body problem [6]. The quantum many-body system, on the other hand, can help better explain the mechanism behind the neural network [2, 18]. The neural network based approaches have been shown effective in both the neural IR [8, 9, 11] and QA fields [14, 15, 37]. In this paper, we propose a novel quantum-inspired language modeling approach and apply it in ranking-based QA tasks. We expect that our attempt would potentially open a

door for the consequent research across the fields of information retrieval, neural network, and quantum mechanics.

6 EXPERIMENTS

6.1 Datasets

We conduct our evaluation on three widely used datasets (summarized in Table 2) for the question answering task.

- **TRECQA** is a benchmark dataset used in the Text Retrieval Conference (TREC)'s QA track(8-13) [35]. It contains two training sets, namely TRAIN and TRAIN-ALL. We use TRAIN-ALL, which is a larger and contains more noise, in our experiment, in order to verify the robustness of the proposed model.
- **WIKIQA** is an open domain question-answering dataset released by Microsoft Research [36]. We remove all questions with no correct candidate answers.
- **YahooQA**, collected from Yahoo Answers, is a benchmark dataset for community-based question answering. It contains 142627 questions and answers. The answers are generally longer than those in TRECQA and WIKIQA. As introduced in [32], we select the QA pairs containing questions and the best answers of length 5-50 tokens after removing non-alphanumeric characters. For each question, we construct negative examples by randomly sampling 4 answers from the set of answer sentences.

6.2 Algorithms for Comparison

QMWF-LM is a quantum inspired language model. The closest approaches to our QMWF-LM are QLM [31] and NNQLM [40]. We treat them as our baselines.

- **QLM**. The question and answer sentences are represented by the density matrices $\boldsymbol{\rho}_q$ and $\boldsymbol{\rho}_a$, respectively. Then the score function is based on the negative Von-Neumann (VN) Divergence between $\boldsymbol{\rho}_q$ and $\boldsymbol{\rho}_a$.
- **NNQLM-II**. This model is an end-to-end quantum language model. We actually compare our model with NNQLM-II, which performs much better than NNQLM-I [40]. The question and answer sentences are also encoded in the density matrix, but with the embedding vector as the input. The density matrix $\boldsymbol{\rho}$ is trained by a neural network. The matching score is computed by the convolutional neural network over the joint representation of two density matrices $\boldsymbol{\rho}_q$ and $\boldsymbol{\rho}_a$.
- **QMWF-LM**. QMWF-LM is the model introduced in this paper. It is inspired by the quantum many-body wave function. QMWF-LM-word is the model whose input matrix is the word embedding, QMWF-LM-char is the model whose input matrix is based on char embedding.

Since we utilize the CNN with product pooling to implement the QMWF based LM, we compare our model with a range of CNN-based QA models [10, 26, 27]. Additional CNN-based models include QA-CNN [10], and AP-CNN which is the attentive pooling network. Our focus is to show the potential of the QMWF-inspired LM and its connection with CNN, rather than a comprehensive comparison with all the recent CNN based QA models. Therefore, we just pick up a couple of basic and typical CNN-based QA models for comparison.

6.3 Evaluation Metrics

For experiments on TRECQA and WIKIQA, we use the same matrix as in [26], namely the MAP (mean average precision) and MRR (mean reciprocal rank). For experiments on YahooQA dataset, we use the same metrics as in [34], namely Precision@1 (P@1) and MRR. P@1 is defined by $\frac{1}{N}\sum_1^N[rank(A^*)=1]$ where $[\cdot]$ is the indicator function and A^* is the ground truth.

According to Wilcoxon signed-rank test, the symbols α and β denote the statistical significance (with $p < 0.05$) over QLM and NNQLM-II, respectively, in experimental table.

6.4 Implementation Details and Hyperparameters

For QLM and NNQLM, we use the same parameters introduced in [40]. The model is implemented by Tensorflow and the experimental machine is based on TITAN X GPU. We train our model for 50 epochs and use the best model obtained in the dev dataset for evaluation in the test set. We utilize the Adam [17] optimizer with learning rate [0.001,0.0001,0.00001]. The batch size is tuned among [80,100,120,140]. The L2 regularization is tuned among [0.0001,0.00001,0.000001]. For QMWF-LM-word, we initialize the input layer with 300-dimensional Glove vectors [24]. For QMWF-LM-char, the initial char embedding is a one-hot vector. In QMWF algorithm, we use the logarithm value for the product pooling, and use two or three words in a patch to capture the phrase information.

6.5 Experimental Results

Table 3 reports the results on the TRECQA dataset. The first group shows a comparison of the three quantum inspired language models. QMWF-LM-word significantly outperforms QLM by 7% on MAP and 9% on MRR. The result of QMWF-LM-word is comparable to NNQLM-II, In the second group, we compare our model with a range of CNN-based models against their results reported in the corresponding original papers. We can see that the QMWF-LM-word achieves a better performance over the CNN [38] by 4% on MAP and 3% on MRR and comparable to other CNN models.

Table 4 reports the results on WIKIQA . QMWF-LM-word significantly outperforms QLM (by 18% on MAP and 20% on MRR), as well as NNQLM-II by about 4% on MAP and 5% on MRR. In comparison with CNN models, QMWF-LM-word outperforms the QA-CNN and AP-CNN by (1%∼2%) on MAP and MRR, against their results reported in the corresponding original papers.

The experimental results on YahooQA are shown in Table 5. Our QMWF-LM-word achieves a significant improvement over QLM by about 18% on P@1 and 14% on MRR. It also outperforms NNQLM-II on P@1 by 11% and on MRR by 7%. Compared with the results of other CNN model on YahooQA dataset as reported in [32], QMWF-LM-word shows some marginal improvements over QA-CNN and AP-CNN by about 1% on P@1 and 2% on MRR. Note that the data preprocessing of YahooQA dataset in our experiments is a little different, as we randomly sample 4 negative examples from the answers sentence set.

Table 2: Statistics of Datasets

	TREC-QA			WIKIQA			YahooQA		
	TRAIN	DEV	TEST	TRAIN	DEV	TEST	TRAIN	DEV	TEST
#Question	1229	82	100	873	126	243	56432	7082	7092
#Pairs	53417	1148	1517	8672	1130	2351	287015	35880	35880
%Correct	12.0	19.3	18.7	12.0	12.4	12.5	20	20	20

Table 3: Experimental Result on TRECQA (raw). α denotes significant improvement over QLM.

Model	MAP	MRR
QLM	0.678	0.726
NNQLM-II	0.759	0.825
CNN (Yu et al.) [38]	0.711	0.785
CNN (Severyn) [27]	0.746	0.808
aNMM (Yang et al.) [37]	0.749	0.811
QMWF-LM-char	0.715$^{\alpha}$	0.758 $^{\alpha}$
QMWF-LM-word	**0.752** $^{\alpha}$	**0.814** $^{\alpha}$

Table 4: Experimental Result on WIKIQA. α and β denote significant improvement over QLM and NNQLM-II, respectively.

Model	MAP	MRR
QLM	0.512	0.515
NNQLM-II	0.650	0.659
QA-CNN (Santos et al.) [10]	0.670	0.682
AP-CNN (Santos et al.) [10]	0.688	0.696
QMWF-LM-char	0.657 $^{\alpha}$	0.679 $^{\alpha}$
QMWF-LM-word	**0.695** $^{\alpha\beta}$	**0.710** $^{\alpha\beta}$

Table 5: Experimental Result on YahooQA. α and β denote significant improvement over QLM and NNQLM-II, respectively.

Model	P@1	MRR
Random guess	0.200	0.457
QLM	0.395	0.604
NNQLM-II	0.466	0.673
QA-CNN (Santos et al.)[10]	0.564	0.727
AP-CNN (Santos et al.)[10]	0.560	0.726
QMWF-LM-char	0.513 $^{\alpha}$	0.696 $^{\alpha}$
QMWF-LM-word	**0.575** $^{\alpha\beta}$	**0.745** $^{\alpha\beta}$

As we can see from the results, the performance of the QMWF-LM-char is not as good as QMWF-LM-word. However, QMWF-LM-char also has better results on WIKIQA and YahooQA datasets compared with NNQLM-II. We will give a further analysis of this phenomenon in Section 6.6.2.

6.6 Discussion and Analysis

6.6.1 The Result Analysis. The experimental results show that our proposed model, namely QMWF-LM, has achieved a significant improvement over QLM on three QA datasets, and outperforms NNQLM-II on both WIKIQA and YahooQA datasets. Especially on YahooQA, which is the largest among the three datasets, QMWF-LM significantly outperforms the other two quantum-inspired LM approaches. Note that the original QLM is trained in an unsupervised manner. Therefore, unsurprisingly it under-performs the other two supervised models (i.e., NNQLM and QMWF-LM). NNQLM adopts the embedding vector as its input and uses the convolutional neural network to train the density matrix. However, the interaction among words is not taken into account in NNQLM. The experiment also shows that the proposed model can achieve a comparable and even better performance over a couple of CNN-based QA models. In summary, our proposed model reveals the analogy between the quantum many-body system and the language modeling, and further effectively bridge the quantum many-body wave function inspired LM with the neural network design.

6.6.2 The Comparison between Word embedding and Char embedding. In our experiment, the input layer is based on word embedding and char embedding. For char embedding, we treat the text as a kind of raw signal which has been proved effective in modeling sentence [16]. As char embedding is initialized by one-hot vector, the semantic information is only based on training dataset. In QA tasks, the semantic matching is needed for a better performance. Compared with char embedding, pre-trained embedding trained by an external large corpus (rather than training data only) is more effective.

6.6.3 Influence of Channels in Convolution. As introduced in Section 3. As we explained, the number of convolution channels is corresponding to R which is the rank of a tensor. In general, there is no straightforward algorithm to determine the rank of a given tensor. We select the optimal R in a range $[20, 200]$ with increment 5. For different datasets, we need set a suitable number of channels to obtain the best performance. The number of channels is set to 150 for TRECQA and WIKIQA dataset, and 200 For YahooQA dataset.

6.6.4 Efficiency Analysis. As we utilize convolution neural network to implement the operation of tensor decomposition. The efficiency relies on the convolution neural network. In our experiment, for QMWF-LM-word, after training 20 epochs, we will obtain the results. For QMWF-LM-char, the training epoch is set to be 200 to obtain a good result. All the training time is less than 5 hour to obtain a comparable result.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a Quantum Many-body Wave Function (QMWF) inspired Language Modeling (QMWF-LM) framework. We have shown that the incorporation of QMWF has enhanced the representation space of quantum-inspired LM approaches, in the sense that QMWF-LM can model the complex interactions among

words with multiples meanings. In addition, inspired by the recent progress on solving the quantum many-body problem and its connection to the neural network, we analyze the common characteristics between the quantum-inspired language modeling and the neural network. A series of derivations (based on projection and tensor decomposition) show that the quantum many-body language modeling representation and matching process can be implemented by the convolutional neural network (CNN) with product pooling. This result simplifies the estimation of the probability amplitudes in QMWF-LM. Based on this idea, we provides a simple algorithm in a basic CNN architecture.

We implement our approach on the question answering task. Experiments on three QA datasets have demonstrated the effectiveness of our proposed QMWF based LM. It achieves a significant improvement over its quantum-like counterparts, i.e., QLM and NNQLM. It can also achieve a competitive performance compared with several convolutional neural network based approaches. Furthermore, based on the analytical and empirical evidence presented in this paper, we can conclude that the proposed approach has made the first step to bridge the quantum-inspired formalism, language modeling and neural network structure in a unified framework.

In the future, the quantum many-body inspired language model should be investigated in more depth from both theoretical and empirical perspectives. Theoretically, a more unified framework to explain another widely-adopted neural network architecture, i.e., Recurrent Neural Network (RNN), can be explored based on the mechanism of quantum many-body language modeling. Practically, we will apply and evaluate QMWF-LM on other IR tasks with larger scale datasets.

8 ACKNOWLEDGMENTS

This work is supported in part by the state key development program of China (grant No. 2017YFE0111900), Natural Science Foundation of China (grant No. U1636203, 61772363), and the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 721321.

REFERENCES

[1] Ivano Basile and Fabio Tamburini. 2017. Towards Quantum Language Models. In *Proc. of EMNLP*. 1840 –1849.
[2] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. 2016. Quantum machine learning. *arXiv preprint arXiv:1611.09347* (2016).
[3] William Blacoe, Elham Kashefi, and Mirella Lapata. 2013. A Quantum-Theoretic Approach to Distributional Semantics.. In *Proc. of HLT-NAACL*. 847–857.
[4] Peter D. Bruza, Zheng Wang, and Jerome R. Busemeyer. 2015. Quantum cognition: a new theoretical approach to psychology. *Trends in Cognitive Sciences* 19, 7 (2015), 383 – 393.
[5] Jerome R. Busemeyer and Peter D. Bruza. 2013. *Quantum Models of Cognition and Decision.* Cambridge University Press.
[6] G. Carleo and M. Troyer. 2017. Solving the quantum many-body problem with artificial neural networks. *Science* 355 (2017), 602–606.
[7] Nadav Cohen, Or Sharir, and Amnon Shashua. 2016. On the Expressive Power of Deep Learning: A Tensor Analysis. *Computer Science* (2016).
[8] Nick Craswell, W. Bruce Croft, Maarten de Rijke, Jiafeng Guo, and Bhaskar Mitra. 2017. SIGIR 2017 Workshop on Neural Information Retrieval (Neu-IR'17). In *Proc. of SIGIR*. 1431–1432.
[9] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. 2017. Neural Ranking Models with Weak Supervision. In *Proc. of SIGIR*. 65–74.
[10] Cıcero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *CoRR, abs/1602.03609* (2016).
[11] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proc. of CIKM*. 55–64.
[12] E. Haven and A. Khrennikov. 2013. *Quantum Social Science.* Cambridge University Press.
[13] Frank L Hitchcock. 1927. The expression of a tensor or a polyadic as a sum of products. *Studies in Applied Mathematics* 6, 1-4 (1927), 164–189.
[14] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Proc. of NIPS*. 2042–2050.
[15] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).
[16] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-Aware Neural Language Models.. In *Proc. of AAAI*. 2741–2749.
[17] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[18] Yoav Levine, David Yakira, Nadav Cohen, and Amnon Shashua. 2017. Deep Learning and Quantum Entanglement: Fundamental Connections with Implications to Network Design. *CoRR* abs/1704.01552 (2017). http://arxiv.org/abs/1704.01552
[19] Qiuchi Li, Jingfei Li, Peng Zhang, and Dawei Song. 2015. Modeling multi-query retrieval tasks using density matrix transformation. In *Proc. of SIGIR*. ACM, 871–874.
[20] Massimo Melucci and Keith van Rijsbergen. 2011. *Quantum Mechanics and Information Retrieval.* Springer Berlin Heidelberg, Berlin, Heidelberg, 125–155.
[21] Donald Metzler and W. Bruce Croft. 2005. A Markov random field model for term dependencies. In *Proc. of SIGIR*. 472–479.
[22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proc. of NIPS*. 3111–3119.
[23] Michael A. Nielsen and Isaac L. Chuang. 2011. *Quantum Computation and Quantum Information: 10th Anniversary Edition* (10th ed.). Cambridge University Press, New York, NY, USA.
[24] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proc. of EMNLP*. 1532–1543.
[25] Benjamin Piwowarski, Ingo Frommholz, Mounia Lalmas, and Keith van Rijsbergen. 2010. What can quantum theory bring to information retrieval. In *Proc. of CIKM*. 59–68.
[26] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proc. of SIGIR*. ACM, 373–382.
[27] Aliaksei Severyn and Alessandro Moschitti. 2016. Modeling relational information in question-answer pairs with convolutional neural networks. *arXiv preprint arXiv:1604.01178* (2016).
[28] Fei Song and W. Bruce Croft. 1999. A General Language Model for Information Retrieval (poster abstract). In *Proc. of SIGIR*. 279–280.
[29] Alessandro Sordoni, Yoshua Bengio, and Jian-Yun Nie. 2014. Learning Concept Embeddings for Query Expansion by Quantum Entropy Minimization.. In *Proc. of AAAI*, Vol. 14. 1586–1592.
[30] Alessandro Sordoni and Jian-Yun Nie. 2013. Looking at vector space and language models for ir using density matrices. In *Proc. of QI*. Springer, 147–159.
[31] Alessandro Sordoni, Jian-Yun Nie, and Yoshua Bengio. 2013. Modeling term dependencies with quantum language models for IR. In *Proc. of SIGIR*. ACM, 653–662.
[32] Yi Tay, Minh C Phan, Luu Anh Tuan, and Siu Cheung Hui. 2017. Learning to Rank Question Answer Pairs with Holographic Dual LSTM Architecture. In *Proc. of SIGIR*. ACM, 695–704.
[33] Cornelis Joost Van Rijsbergen. 2004. *The geometry of information retrieval.* Cambridge University Press.
[34] Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. 2016. A Deep Architecture for Semantic Matching with Multiple Positional Sentence Representations.. In *Proc. of AAAI*. 2835–2841.
[35] Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the Jeopardy Model? A Quasi-Synchronous Grammar for QA.. In *Proc. of EMNLP-CoNLL*, Vol. 7. 22–32.
[36] Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering.. In *Proc. of EMNLP*. Citeseer, 2013–2018.
[37] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193* (2015).
[38] Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer sentence selection. *arXiv preprint arXiv:1412.1632* (2014).
[39] ChengXiang Zhai. 2008. *Statistical Language Models for Information Retrieval.* Morgan & Claypool Publishers.
[40] Peng Zhang, Jiabin Niu, Zhan Su, Benyou Wang, Liqun Ma, and Dawei Song. 2018. End-to-End Quantum-like Language Models with Application to Question Answering. In *Proc. of AAAI*. 5666–5673.
[41] Xiaozhao Zhao, Peng Zhang, Dawei Song, and Yuexian Hou. 2011. A novel re-ranking approach inspired by quantum measurement. In *Proc. of ECIR*. 721–724.
[42] Guido Zuccon and Leif Azzopardi. 2010. Using the Quantum Probability Ranking Principle to Rank Interdependent Documents. In *Proc. of ECIR*. 357–369.