Quantum Inspired Word Representation and Computation

Shen Li
shen@mail.bnu.edu.cn

Renfen Hu
irishu@mail.bnu.edu.cn

Jinshan Wu
jinshanw@bnu.edu.cn

Beijing Normal University

Abstract

Word meaning has different aspects, while the existing word representation “compresses” these aspects into a single vector, and it needs further analysis to recover the information in different dimensions. Inspired by quantum probability, we represent words as density matrices, which are inherently capable of representing mixed states. The experiment shows that the density matrix representation can effectively capture different aspects of word meaning while maintaining comparable reliability with the vector representation. Furthermore, we propose a novel method to combine the coherent summation and incoherent summation in the computation of both vectors and density matrices. It achieves consistent improvement on word analogy task.

1 Introduction

One of the most fundamental questions in Natural Language Processing is how to represent language units (words, sentences, texts) in a computational way, i.e. let the machine “understand” human language. The existing solutions evolve from symbolic expressions to mathematical vectors. Recently, the dense vector representation has achieved considerable success in various fields of NLP. The representative works include static word embeddings (Mikolov et al., 2013b; Pennington et al., 2014) and dynamic contextual embeddings (Peters et al., 2018; Devlin et al., 2018).

When training word representations, the meaning of a word depends on its various contexts in a large corpus. Naturally, the word representation may have different aspects of properties by gaining information from different contexts. For example, the word “king” has high-level properties such as royalty and male gender (Levy and Goldberg, 2014). This phenomenon could be easily observed from the nearest neighbors of its word embeddings. However, the existing word representation “compresses” different aspects into a single vector, and it needs further analysis to recover the information in different dimensions (Rogers et al., 2018). The “compression” also exists in sentence representations. Obviously, different extrinsic tasks may rely on different aspects of the representations. It naturally raises the following questions: Can we represent different aspects of language meaning inherently?

To address this issue, this paper introduces quantum probability into language representation, and attempts to represent words with density matrices. In physics, density matrix is used to describe a state of a particle. Each eigenstate of the particle is represented by a eigenvector in the density matrix, and each eigenvalue can be regarded as the probability of the particle in its corresponding eigenstate (Fano, 1957). Therefore, the density matrix representation is capable of storing mixed states. Meanwhile, it allows both coherent summation and incoherent summation of the eigenstates, which can be considered as vector summation and probability summation.

After obtaining the word representation, we are interested in the computation between the representations. It is because the computation could illustrate the relations between words, and moreover, it is necessary for modeling higher level language units (sentences or texts) that are built upon the combination of word meaning in a specific way, e.g. Bag-of-words, CNNs, LSTMs and Transformers. If we can use vector summation and probability summation to combine different eigenstates of a word, can we apply them to the computation between words? Motivated by this question, we incorporate both vector summation and probability summation into computation of words.
In the experiments, we implement our methods on skip-gram model with negative sampling (SGNS) (Mikolov et al., 2013b), since it is one of the most popular models to train word representations. Meanwhile, as it is based on a simple yet effective neural network architecture, we could easily transfer the methods that work well in SGNS to other neural models. In addition, we impose the probability summation on both pre-trained vectors and pre-trained density matrices, as we will see from Equation (19). The effectiveness is testified by the analogy task, i.e. “a is to b as c is to ?” the solution of which traditionally relies on vector summation.

The experiments show the density matrix representation can effectively capture different aspects of word meaning while maintaining comparable reliability with the vector representation. Moreover, after further imposing probability summation, both the pre-trained vectors and density matrices achieve consistent improvement on word analogy tasks. It suggests that the probability summation could capture meaningful language information different from that of vector summation. Thus the contribution of this paper is two-fold:

- We introduce a novel word representation based on density matrix, and successfully apply it to neural network models. This representation can effectively capture different aspects of word meaning.
- We integrate both probability summation and vector summation into the computation between words in both vector form and density matrix form. This computation method could better sketch the morphological and semantic relations between words.

2 Theoretical background

Before introducing density matrix in quantum mechanics, we need to look at how to describe a state of a classical object. In classical mechanics, objects can be in a mixture of states described by a probability distribution. For example, a coin can be in “tail” state (denoted as event $|\psi_0\rangle = |0\rangle$) or “head” state (denoted as event $|\psi_1\rangle = |1\rangle$), where $|\psi\rangle$, $|0\rangle$ and $|1\rangle$ are column vectors in Dirac notation (Dirac, 1939). Since the two states have nearly equal probability, the mixture state of a coin can be described as $\rho = \sum_{i=0}^{1} p_i |\psi_i\rangle \langle \psi_i | = \frac{1}{2} (|0\rangle \langle 0 | + |1\rangle \langle 1 |)$, where $|\psi\rangle$, $|0\rangle$ and $|1\rangle$ are row vectors. Here Dirac notation $|0\rangle \langle 0 |$ denotes the event that the coin is in state 0. For classical object, neither $|0\rangle + |1\rangle$ nor $|0\rangle \langle 1 |$ is meaningful. That is to say, vector summation is meaningless for classical states, while probability summation is meaningful. In principle, there can be object’s states derived from probability summations of simple events, e.g. the top of a coin $\rho = \sum_{i=0}^{1} p_i |\psi_i\rangle \langle \psi_i |$ for any $p_i$ or the top of a dice $\rho = \sum_{i=1}^{6} p_i |\psi_i\rangle \langle \psi_i |$ for any $p_i$. However, there is no such classical object whose states derived from $|0\rangle + |1\rangle$ or equivalently $|0\rangle \langle 0 | + |1\rangle \langle 1 | = |0\rangle (|0\rangle + |1\rangle) \langle 0 | + |1\rangle (|1\rangle + |0\rangle) \langle 1 | + |1\rangle \langle 1 |$ (as we will see this equivalence later).

In quantum mechanics, quantum objects can be in a state derived from either vector summation or probability summation of simple events. In fact, the concept of simple events is slightly different from the classical simple events. All the pure states, which are the states involving only vector summation of simple events, such as $|\psi_i\rangle = \alpha_i |0\rangle + \beta_i |1\rangle$, form a vector space, so that any normalized orthogonal basis of the vector space can be seen as the set of simple events. This also means that a linear transformation of a set of chosen simple events (basis) is another set of simple events. The representation of classical objects is a particular situation in which only one of $\alpha$ and $\beta$ is not zero. Therefore, we can use the same mathematical form to represent mixture states of objects, i.e. density matrix

$$\rho = \sum_{ij} \rho_{ij} |i\rangle \langle j |,$$  \hspace{1cm} \text{(1)}

as long as $\rho$ is hermitian, normalized and semi-positively defined, meaning correspondingly

$$\rho^\dagger = \rho \Rightarrow \rho_{ij} = \rho^{\ast}_{ji},$$  \hspace{1cm} \text{(2a)}

$$tr (\rho) \triangleq \sum_{i} \langle i | \rho | i \rangle = 1 \Rightarrow \sum_{i} \rho_{ii} = 1,$$  \hspace{1cm} \text{(2b)}

$$\rho_{\mu \mu} = \langle \mu | \rho | \mu \rangle \geq 0.$$  \hspace{1cm} \text{(2c)}
Here in linear algebra terms, \( \langle \mu \mid \nu \rangle \) is the inner product of left (row) vector \( \langle \mu \mid \) and the right (column) vector \( \mid \nu \rangle \) (or of the right vectors \( \mid \mu \rangle \) and \( \mid \nu \rangle \)), where if \( \mid \mu \rangle \) and \( \mid \nu \rangle \) belong to a normalized and orthogonal vector basis of the space then,

\[
\langle \mu \mid \nu \rangle = \delta_{\mu \nu},
\]

where \( \delta_{\mu \nu} \) is the Kronecker delta; \( \langle \mu \mid \rho \mid \mu \rangle \) means a row vector \( \langle \mu \mid \) times a matrix \( \rho \) and then time a column vector \( \mid \mu \rangle \), thus its end result is also a number; differently, the product of a column vector first and then a row vector, such as \( \mid i \rangle \langle j \mid \), is in fact a matrix. Now we can see both the classical mixture states and quantum pure states can be represented in a density matrix form.

It is experimental observation on quantum states that drives the theory of quantum systems to go beyond probability summation, and allow vector summation. In the current case of language phenomena, we assume that word meaning is a kind of mixed states, and both probability summation and vector summation of word representations are meaningful.

3 Density matrix representation of words

Although it is not common to represent probabilities with negative values (eigenvectors in density matrix can be considered as probabilities), in fact, negative probabilities have been considered in quantum mechanics (Dirac, 1942; Feynman, 1987). Thus, in this paper, indefinite matrices are used to represent words instead of positive semi-definite matrices which are widely used in quantum probability. When negative eigenvalues are allowed in density matrix, it can give rise to more flexible representations.

3.1 Definitions

Given space \( \mathcal{L}^{n \times n} \), which belongs to Hilbert Space and consists of hermitian matrices, we have the following definitions on density matrix \( \rho \).

Definition 1 Given \( \rho^A, \rho^B \in \mathcal{L} \), then the inner product is defined by

\[
\langle \rho^A, \rho^B \rangle = \text{Tr} \left( \rho^A \rho^B \right).
\]

Definition 2 Given \( \rho^A \in \mathcal{L} \), then the norm is defined by

\[
\| \rho^A \| = \sqrt{\langle \rho^A, \rho^A \rangle} = \sqrt{\text{Tr} \left( \rho^A \rho^A \right)}.
\]

Definition 3 Given \( \rho^A, \rho^B \in \mathcal{L} \), then the distance between \( \rho^A \) and \( \rho^B \) is defined by

\[
d_{\rho^A, \rho^B} = \| \rho^A - \rho^B \|.
\]

Definition 4 Given \( \rho^A, \rho^B \in \mathcal{L} \), then the cosine distance between \( \rho^A \) and \( \rho^B \) is defined by

\[
\cos \theta = \frac{\langle \rho^A, \rho^B \rangle}{\| \rho^A \| \| \rho^B \|} = \frac{\text{Tr} \left( \rho^A \rho^B \right)}{\| \rho^A \| \| \rho^B \|}.
\]

3.2 Learning Word Representations

In this paper, we implement our method in the skip-gram model with negative sampling (SGNS) (Mikolov et al., 2013b), since it is one of the most popular models to train word representations. It should also be noted that SGNS is based on a simple yet effective neural network architecture. We could transfer the methods that work well in SGNS to other neural models.
To train word embeddings with SGNS, the optimization objective is

$$J = \arg \max_{\theta} \sum_{(w,c) \in D} \log \sigma (v^c \cdot v^w) + \sum_{(w,c) \in D'} \log \sigma (-v^c \cdot v^w),$$

(8)

where $\theta$ denotes the parameters of all vectors $v$, $w$ and $c$ are central word and context word respectively, $v^w$ and $v^c$ are their embeddings, $\sigma$ is sigmoid function and $(w, c) \in D$ means that $w$ and $c$ cooccur in a window in the document $D$ and $(w, c) \in D'$ represents that two words cooccur in Negative Sampling.

The objective can be rewritten as

$$J = \arg \max_{\theta} \sum_{(w,c) \in D} \log \sigma (\langle v^c, v^w \rangle) + \sum_{(w,c) \in D'} \log \sigma (-\langle v^c, v^w \rangle).$$

(9)

If words are represented by density matrices, then the objective becomes

$$J = \arg \max_{\theta} \sum_{(w,c) \in D} \log \sigma (\langle \rho^c, \rho^w \rangle) + \sum_{(w,c) \in D'} \log \sigma (-\langle \rho^c, \rho^w \rangle)
= \arg \max_{\theta} \sum_{(w,c) \in D} \log \sigma (\text{Tr} (\rho^w \rho^c)) + \sum_{(w,c) \in D'} \log \sigma (-\text{Tr} (\rho^w \rho^c)).$$

(10)

In word2vec, to update parameters in SGNS, the process is

$$\theta (t + 1) = \theta (t) - \eta \frac{\partial J}{\partial \theta (t)},$$

(11)

where $t$ is the number of steps and $\eta$ is the learning rate. Taking a central word $w$ and a context word $c \in D \cup D'$ as an example, to update $v^w$, the process can be written like

$$v^w (t + 1) = v^w (t) - \eta \frac{\partial J}{\partial v^w (t)}
= v^w (t) + \eta (l - \sigma (\langle v^w (t), v^c (t) \rangle)) v^c (t)
= v^w (t) + \eta (l - \sigma (\langle v^w (t), v^c (t) \rangle)) v^c (t)
= v^w (t) + \eta \left(l - \sigma \left( \sum_i v^w_i (t) v^c_i (t) \right) \right) v^c (t),$$

(12)

where $l$ is 1 if $c \in D$ or 0 if $c \in D'$.

If words are represented by hermitian matrices, the equation becomes

$$\rho^w (t + 1) = \rho^w (t) + \eta (l - \sigma (\langle \rho^w (t), \rho^c (t) \rangle)) \rho^c (t)
= \rho^w (t) + \eta \left(l - \sigma \left( \text{Tr} (\rho^w (t) \rho^c (t)) \right) \right) \rho^c (t)
= \rho^w (t) + \eta \left(l - \sigma \left( \sum_{ij} \rho^w_{ij} (t) \rho^c_{ij} (t) \right) \right) \rho^c (t).$$

(13)

For instance, given

$$v^w (t) = \left( a^w_0 \ a^w_1 \ a^w_2 \ b^w_1 \right), \ v^c (t) = \left( a^c_0 \ a^c_1 \ a^c_2 \ b^c_1 \right),$$

$$\rho^w (t) = \left( \begin{array}{cc} a^w_0 & a^w_1 + b^w_1 i \\ a^w_1 - b^w_1 i & a^w_2 \end{array} \right), \ \rho^c (t) = \left( \begin{array}{cc} a^c_0 & a^c_1 + b^c_1 i \\ a^c_1 - b^c_1 i & a^c_2 \end{array} \right),$$

(14a,b)
then
\[
\langle v^w(t), v^c(t) \rangle = \langle a_0^w a_1^c, a_2^w b_1^c \rangle = a_0^w a_0^c + a_1^w a_1^c + b_1^w b_1^c + a_2^w a_2^c,
\]
\[
(15a)
\]
\[
\langle \rho^w(t), \rho^c(t) \rangle = \text{Tr} \left( \rho^w(t) \rho^c(t) \right)
\]
\[
= \left( \begin{array}{cc}
a_0^w & a_1^w + b_1^w i \\
b_1^w - b_1^w i & a_2^w
\end{array} \right)
\left( \begin{array}{cc}
a_0^c & a_1^c + b_1^c i \\
b_1^c - b_1^c i & a_2^c
\end{array} \right)
\]
\[
= a_0^w a_0^c + 2 (a_1^w a_1^c + b_1^w b_1^c) + a_2^w a_2^c.
\]
\[
(15b)
\]
It can be seen that two kinds of inner product are similar. The inner product of density matrices has more items coming from off-diagonal elements. Thus, it is easy to extend SGNS to support density matrix by double off-diagonal elements in computation of inner product.

3.3 Performance of the density matrix representation

| Window | Iteration | Sub-sampling | Low-frequency threshold | Negative Sampling |
|--------|-----------|--------------|-------------------------|-------------------|
| 5      | 5         | 1e-3         | 20                      | 10                |

Table 1: Hyperparameters in the training of word matrices.

| Dataset                        | Vector \(dim = 36\) | Matrix \(dim = 8\) | Vector \(dim = 136\) | Matrix \(dim = 16\) | Vector \(dim = 300\) | Matrix \(dim = 24\) | Vector \(dim = 528\) | Matrix \(dim = 32\) |
|--------------------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|---------------------|-------------------|
| Google syntactic               | 32.85%              | 33.33%            | 62.03%              | 61.23%            | 64.59%              | 65.55%            | 64.68%              | 65.36%            |
| Google semantic                | 34.18%              | 36.64%            | 66.13%              | 66.79%            | 74.79%              | 74.68%            | 76.05%              | 74.82%            |
| BATS Inflectional morphology   | 35.32%              | 36.27%            | 59.25%              | 59.49%            | 61.11%              | 62.28%            | 60.30%              | 60.80%            |
| BATS Derivational morphology   | 5.28%               | 5.16%             | 11.47%              | 12.02%            | 12.74%              | 12.81%            | 11.47%              | 11.98%            |
| BATS Encyclopedic semantics    | 16.74%              | 17.99%            | 34.71%              | 34.74%            | 38.50%              | 38.79%            | 37.36%              | 38.73%            |
| BATS Lexicographic semantics   | 4.11%               | 5.28%             | 9.09%               | 10.56%            | 9.97%               | 10.56%            | 9.38%               | 10.56%            |

Table 2: The results of word analogy task with different dimensions. For fair comparison, the number of parameters of vector form is the same as that of density matrix in each group. For instance, the number of parameters of 300 dimensional vector is equal to that of 24 dimensional density matrix.

In the experiments, we train word vectors and word matrices with the same number of parameters on Wiki2010 corpus. The settings can be seen in Table 1. To testify the reliability of word representations, we evaluate the vectors and matrices on word analogy datasets BATS (Gladkova et al., 2016) and Google Analogy (Mikolov et al., 2013a).

As shown in Table 2, density matrices and vectors (with the same number of parameters) achieve similar accuracies consistently from low to high dimensions. It indicates that matrices can represent...
Table 3: Nearest neighbors of \textit{bank} and \textit{fiction} according to density matrix \( \rho \) and different eigenvectors of the matrices.

(a) Nearest neighbors of \textit{bank}.

| \( \rho \) | Eigenvalues |
|---------|-------------|
| 0.3572 | -0.2969     |
| 0.0119 | -0.1852     |
| -0.0917| 0.3605      |
| 0.0119 | 0.4591      |
| banks  | receivership |
| bank's | liquidation  |
| citibank| bankrupt    |
| bactyls| scrapping    |
| lobe   | parcels      |
| trading| thoroughbreeds|
| savings| distilleries |
| lloyds | nationalised |
| capital| leases       |
| banking| franchise    |
| depository| sale      |
| deposit| freight      |
| banque| government-owned |
| citigroup| tramways  |
| river  | plying      |
| parcel | turlongs    |
| canal  | subsidiaries |
| brokerage| dealerships |

(b) Nearest neighbors of \textit{fiction}.

| \( \rho \) | Eigenvalues |
|---------|-------------|
| -0.5497| -0.0157     |
| -0.0119| 0.3008      |
| -0.4772| 0.5184      |
| 0.1921 | 0.5184      |
| 0.38    | 0.11        |
| 0.38    | 0.0115      |
| 0.54    | 0.0115      |
| 0.21    | 0.0115      |
| 0.43    | 0.0115      |
| 0.54    | 0.0115      |
| 0.21    | 0.0115      |
| 0.43    | 0.0115      |

| science-fiction | explored | comedies | whirlpool | carving | 1603 | etto | rotten | astronom | glock |
| non-fiction     | exploring | novelists | eau       | vase    | 1604 | mle   | inaccuracies | observer | 1651 |
| nonfiction      | spin-offs | ziefeld | racetrack | dinghy | 1651 | ddr   | tracts | skeptic | 1651 |
| novels          | adv       | beauties | lode      | archery | 1599 | lugi  | negatives | watcher | 1651 |
| anthology       | evolutions | sattirized | aqueduct | blade | 1584 | renzo | plagiarism | creationism | 1651 |
| thrillers       | researching | hollywood | misis | springboard | 1578 | e5 | libel | airman | 1651 |
| comic           | viz       | follies | euclid    | finish | 1597 | brom | seditious | shooter | 1651 |
| mystery         | documenting | operetas | woodbine | teak | 1707 | steyr | forgeries | seeker | 1651 |
| novel           | independently | burlesque | brea | silverware | 1593 | bugatti | clippings | naismith | 1651 |
| bestselling     | revisited | churchill's | raceway | round | 1671 | 35mm | allegations | sailor | 1651 |
| anthologies     | spawned | revues | biel | leg | 1644 | krzyzstol | op-ed | horseman | 1651 |
| asimov's        | recent | wholesome | haute | quadrangular | 1661 | dV | obscenity | communicator | 1651 |
| comics          | adapting | actresses | chasm | berth | 1642 | glock | unsubstantiated | creationism | 1651 |
| heinlein        | extensively | parodying | hairpin | bowls | 1662 | dS | antiquities | atheist | 1651 |
| cyberpunk       | predating | heroines | downhill | putt | 1669 | dac | fingerprints | athlete | 1651 |
| dystopian       | episodic | musicals | descartes | spectacularly | 1592 | carlo | archival | mountaineer | 1651 |
| suspense        | aspects | astaire | valhalla | lancers | 1606 | bosch | evidence | starleet | 1651 |
| stories         | adaptations | baum's | bvld | turf | 1723 | riccardo | misrepresentation | hawkeye | 1651 |
| fandom          | interrelated | wildes | edma | jetty | 1601 | e5 | accusations | sportswriter | 1651 |
| comic-book      | latest | entertainments | seca | canoes | dafydd | bolzano | sedition | parallax | 1651 |
what vectors encode. Equation (15) can easily explain the experiment results. If off-diagonal elements of density matrices time $\frac{1}{\sqrt{2}}$, these two methods have similar effects in computation of inner product.

As we all know, given a density matrix $\rho$, it can be written as the combination of its eigenvectors, $\rho = \sum_i |i\rangle\langle i|$, where $i$ is an eigenvalue and $|i\rangle$ is its corresponding eigenstate. As eigenvectors of a density matrix can represent pure states, we could examine the nearest neighbors of each pure state by what vectors encode. Equation (15) can easily explain the experiment results. If off-diagonal elements of density matrices time $\frac{1}{\sqrt{2}}$, these two methods have similar effects in computation of inner product.

As we all know, given a density matrix $\rho$, it can be written as the combination of its eigenvectors, $\rho = \sum_i |i\rangle\langle i|$, where $i$ is an eigenvalue and $|i\rangle$ is its corresponding eigenstate. As eigenvectors of a density matrix can represent pure states, we could examine the nearest neighbors of each pure state by using $|i\rangle\langle i|$ if its eigenvalue $i \geq 0$ or $-|i\rangle\langle i|$ if its eigenvalue $i < 0$ rather than $\rho$. $-|i\rangle\langle i|$ is the natural result when we allow negative probabilities in density matrices. As a reference, we also retrieve the nearest neighbors of $\rho$.

As shown in Table 3, we find that the eigenvectors in density matrices could effectively capture different aspects of word meaning, and the absolute eigenvalues are also meaningful. Eigenvectors with larger values have greater impact on word meaning than those with smaller values. Table 3a shows an example of the word bank. The first column is the nearest neighbors of $\rho$ which represents the whole word meaning, and the other columns are the nearest neighbors of each eigenvector. If we use $\rho$ to find the nearest neighbors, the neighbors of “bank” in different senses are mixed together. But if we use $|i\rangle\langle i|$ (or $-|i\rangle\langle i|$), the nearest neighbors can well reflect different aspects of word meaning. For instance, the words in the first three columns and the last two columns have relatively high absolute eigenvalues. The similar phenomenon can also be found in Table 3b regarding the word fiction. The first column is related with fiction content, the second column is about drama and movie, and the last column involves characters.

4 Computation with probability summation and vector summation

The computation method between words plays crucial roles in natural language representation, since it could sketch the relations between words, and serve as a basis for modeling higher level language units such as sentences or texts. As we introduced in Section 2 probability summation and vector summation are two important forms of computation in quantum probability. However, the existing vector representation utilizes only vector summation. In this study, we impose the probability summation on both pre-trained vectors and pre-trained density matrices. Here we illustrate this method with analogy task as an example.

In an analogy question e.g. $king - man + woman \approx queen$, we have 4 word vectors $|a\rangle$, $|b\rangle$, $|c\rangle$ and $|d\rangle$ accordingly. The process can be written as

$$|x\rangle = |b\rangle - |a\rangle + |c\rangle .$$

If the nearest neighbor of $|x\rangle$ is $|d\rangle$, $|x\rangle$ is the correct answer. If not, $|x\rangle$ is a wrong answer.

We can transform vector summation Equation (16) into

$$|x\rangle \langle x| = (|b\rangle - |a\rangle + |c\rangle)(\langle b| - \langle a| + \langle c|).$$

According to Section 2 the probability summation can be written as

$$|x\rangle \langle x| = |b\rangle \langle b| - |a\rangle \langle a| + |c\rangle \langle c| .$$

After combining Equation (17) and Equation (18), we can get

$$|x\rangle \langle x| = (|b\rangle - |a\rangle + |c\rangle)(\langle b| - \langle a| + \langle c|) + \alpha(|b\rangle \langle b| - |a\rangle \langle a| + |c\rangle \langle c|),$$

where $\alpha$ is a factor to balance the ratio of vector summation to probability summation. A larger $\alpha$ denotes higher weight of probability summation in the computation.

We can also easily apply this method to density matrix representations of words,

$$\rho^x \otimes \rho^y = (\rho^b - \rho^a + \rho^c) \otimes (\rho^b - \rho^a + \rho^c) + \alpha(\rho^b \otimes \rho^b - \rho^a \otimes \rho^a + \rho^c \otimes \rho^c),$$

where $\otimes$ means Kronecker product.
| Type          | $\alpha$ | Google syntactics | Google semantics | BA TS Inflectional morphology | BA TS Derivational morphology | BA TS Encyclopedic semantics | BA TS Lexicographic semantics |
|--------------|----------|------------------|-----------------|--------------------------|-------------------------------|-------------------------------|-------------------------------|
| Vector       | 0.0      | 32.85%           | 34.18%          | 35.32%                   | 4.28%                         | 16.74%                       | 4.11%                         |
| dim = 36     | 0.1      | 33.52%           | 34.46%          | 36.07%                   | 5.40%                         | 16.78%                       | 4.69%                         |
|              | 0.2      | 33.77%           | 34.71%          | 36.51%                   | 5.62%                         | 16.67%                       | 4.99%                         |
|              | 0.3      | 34.15%           | 34.85%          | 36.65%                   | 5.70%                         | 16.54%                       | 5.28%                         |
|              | 0.4      | 34.40%           | 35.03%          | 36.91%                   | 5.81%                         | 16.72%                       | 5.28%                         |
|              | 0.5      | 34.49%           | 34.98%          | 36.83%                   | 5.87%                         | 16.87%                       | 5.28%                         |
|              | 0.6      | 34.58%           | 35.19%          | 36.78%                   | 5.95%                         | 16.99%                       | 5.28%                         |
|              | 0.7      | 34.33%           | 35.05%          | 36.75%                   | 5.88%                         | 16.97%                       | 5.28%                         |
|              | 0.8      | 34.53%           | 35.07%          | 36.76%                   | 5.90%                         | 17.01%                       | 5.57%                         |
|              | 0.9      | 34.41%           | 35.03%          | 36.66%                   | 5.97%                         | 17.03%                       | 5.87%                         |
|              | 1.0      | 34.26%           | 34.94%          | 36.50%                   | 5.97%                         | 16.99%                       | 6.16%                         |
|              | 1.1      | 34.30%           | 34.98%          | 36.35%                   | 5.94%                         | 16.99%                       | 6.16%                         |
|              | 1.2      | 34.20%           | 34.98%          | 36.29%                   | 5.97%                         | 16.99%                       | 6.16%                         |

| Table 4: The results of word analogy task with different $\alpha$. Larger $\alpha$ indicates larger weight of probability summation. The pre-trained vectors and density matrices come from Section 3.3. |
Table 5: Different answers and candidates of analogical questions when factors $\alpha$ are set different values. Words are arranged from top to bottom according to their scores. The top one has the largest score.

(a) Comparison in vector form ($\text{dim} = 300$).

(b) Comparison in density matrix form ($\text{dim} = 24$).

There are several works which leverage density matrix to enhance language representations. Sordoni et al. (2013) transform one-hot representation of a sentence into density matrix (quantum language model, QLM) with $R\rho R$ algorithm, and it outperforms classical models on ad-hoc retrieval tasks. Xie et al. (2015) use Unconditional Pure Dependence (UPD) to enhance QLM, leading to an improvement in information retrieval tasks. Zhang et al. (2018) develop a model integrating word embeddings into density matrices to represent questions and answers, and the model is trained to compare questions and answers. Li et al. (2018) propose a similar strategy in classification tasks. Among the existing studies, Sordoni et al. (2013) and Xie et al. (2015) focus on modeling the relations between questions and

5 Related work

There are several works which leverage density matrix to enhance language representations. Sordoni et al. (2013) transform one-hot representation of a sentence into density matrix (quantum language model, QLM) with $R\rho R$ algorithm, and it outperforms classical models on ad-hoc retrieval tasks. Xie et al. (2015) use Unconditional Pure Dependence (UPD) to enhance QLM, leading to an improvement in information retrieval tasks. Zhang et al. (2018) develop a model integrating word embeddings into density matrices to represent questions and answers, and the model is trained to compare questions and answers. Li et al. (2018) propose a similar strategy in classification tasks. Among the existing studies, Sordoni et al. (2013) and Xie et al. (2015) focus on modeling the relations between questions and
answers. They are interested in whether users and documents have quantum entanglements, and how to use quantum entanglements to improve the accuracy of information retrieval.

It should be noted that all of the existing studies attempt to represent sentence-level information with density matrices. If words are still represented by vectors, the advantage of sentence-level density matrices would be limited, and different aspects of word meaning could not be uncovered either. In addition, none of these works incorporate probability summation and vector summation between language units. Therefore, it is necessary to conduct a systematic study of using density matrices to represent language units from word level, and introduce probability summation into the computation between words.

6 Conclusion and Future work

In this paper, inspired by quantum probability, we represent words as density matrices, and successfully apply it to neural network models. This representation can effectively capture different aspects of word meaning while maintaining comparable reliability with the vector representation. In addition, we introduce two basic operations of quantum probability, i.e. probability summation and vector summation, into the computation between words in both vector form and density matrix form. This computation method could better sketch the morphological and semantic relations between words. With a novel representation method and two operations, this work sheds some light on different mathematical forms of language representation.

Language is a combinatorial system, thus the representations of higher level language units e.g. sentences or texts are mostly built upon the computation of word meaning in a specific way. Meanwhile, different NLP tasks rely on different aspects of these representations. Therefore, in the future, we will investigate the density matrix representation of sentences or texts based on the word representations, and apply them to NLP downstream tasks in more complex neural networks.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Paul Adrien Maurice Dirac. 1939. A new notation for quantum mechanics. In Mathematical Proceedings of the Cambridge Philosophical Society, volume 35, pages 416–418. Cambridge University Press.

Paul Adrien Maurice Dirac. 1942. Bakerian lecture-the physical interpretation of quantum mechanics. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, 180(980):1–40.

Ugo Fano. 1957. Description of states in quantum mechanics by density matrix and operator techniques. Reviews of Modern Physics, 29(1):74.

Richard P Feynman. 1987. Negative probability. Quantum implications: essays in honour of David Bohm, pages 235–248.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn’t. In Proceedings of the NAACL Student Research Workshop, pages 8–15.

Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In Proceedings of the eighteenth conference on computational natural language learning, pages 171–180.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. Transactions of the Association for Computational Linguistics, 3:211–225.

Qiuchi Li, Sagar Upreti, Benyou Wang, and Dawei Song. 2018. Quantum-inspired complex word embedding. arXiv preprint arXiv:1805.11351.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119.
Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In Proceedings of NAACL-HLT, pages 2227–2237.

Anna Rogers, Shashwath Hosur Ananthakrishna, and Anna Rumsky. 2018. What’s in your embedding, and how it predicts task performance. In Proceedings of the 27th International Conference on Computational Linguistics, pages 2690–2703.

Alessandro Sordoni, Jian-Yun Nie, and Yoshua Bengio. 2013. Modeling term dependencies with quantum language models for ir. In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, pages 653–662. ACM.

Mengjiao Xie, Yuexian Hou, Peng Zhang, Jingfei Li, Wenjie Li, and Dawei Song. 2015. Modeling quantum entanglements in quantum language models. In Twenty-Fourth International Joint Conference on Artificial Intelligence.

Peng Zhang, Jiabin Niu, Zhan Su, Benyou Wang, Ligun Ma, and Dawei Song. 2018. End-to-end quantum-like language models with application to question answering. In Thirty-Second AAAI Conference on Artificial Intelligence.