Article

Not peer-reviewed version

Geometric Linguistic Space

Hiroshi Isshiki *

Posted Date: 26 March 2025

doi: 10.20944/preprints202503.1909.v1

Keywords: Linguistic space; Mathematical approach to linguistics; word2vec; Transformation of words into numerical vectors; Linguistic translation as coordinates transformation

Article

Geometric Linguistic Space

Hiroshi Isshiki

Institute of Mathematical Analysis (Osaka), Hiroshi Isshiki; isshiki@dab.hi-ho.ne.jp

Abstract: The remarkable progress of generative AI has brought fresh air to linguistics. In animal communication and language translation, generative AI has been used to deepen consideration at the semantic space level, and as a result, geometric similarities in semantic space have been discovered. Even if the expressions in concrete space are different, when they are mapped to semantic space, the existence of geometric similarities that had been hidden until now has been discovered. In the end, this may seem obvious, but it is a surprising discovery. Inspired by this discovery, this paper proposes a basic mathematical theory necessary for understanding linguistic space. This study aims to clarify the geometric properties of the translation process, which are difficult to explain using conventional linguistic theory. In this study, linguistic space is regarded as a mathematical coordinate space, and translation between different languages is treated as a coordinate transformation, assuming the existence of a common invariant, the meaning of language. From this perspective, the approach of this study is called "geometric linguistic space".

Keywords: Linguistic space; Mathematical approach to linguistics; word2vec; Transformation of words into numerical vectors; Linguistic translation as coordinates transformation

---

1. Introduction

The author has studied multiple natural languages and artificial languages such as C for many years as a language hobby. In the process, he has developed a deep interest in language itself and the function of the brain that manipulates language [1]. This study aims to clarify the geometric properties of the translation process, which are difficult to explain using conventional linguistic theory.

In this study, we regard language space as a mathematical coordinate space, and treat translation between different languages as a coordinate transformation, assuming the existence of an invariant called the meaning of language. From this perspective, different languages correspond to relative coordinate systems, and translation is expressed as a transformation (mapping) between them, so we call this approach "geometric language space."

Even though English is linguistically close to German and French, an English speaker with no knowledge of German or French would not be able to understand the meaning of a German or French sentence.

However, through the process of translation, it becomes possible to understand. Why is the process of translation possible?

What does the translation process do mathematically? The introduction of generative AI into language analysis provided a clue to this question [2–14].

If we consider a sentence to be a point in language space, or a vector set or vector sum of "words: the smallest element vectors", it becomes possible to introduce geometry into language. If we consider a sentence to be a substance, then words correspond to the atoms or molecules that make up the sentence. Language space is ultra-multidimensional.

Thinking about it this way, a sentence vector becomes a set of point vectors that give a sequence (trajectory) of point vectors in a language space based on words.

Words in different languages can be matched up using the meaning of the word as a medium. Specifically, between English and French, there is a correspondence at the word level, for example,

 doi:10.20944/preprints202503.1909.v1

between "apple" in English and "pomme" in French. As a result, a close geometric relationship is created between the two different language spaces.

"A delicious apple" in English becomes "pomme délicieuse" in French, and "I eat a delicious apple" in English becomes "Je mange une pomme délicieuse" in French.

If you wanted to say "I eat a delicious apple at a restaurant" in English, you'd say "Je mange une pomme délicieuse au restaurant" in French.

If you wanted to say "I eat a delicious apple recommended by my friend at a restaurant" in English, you'd say "Je mange une pomme délicieuse recommandée par mon ami au restaurant" in French.

"I eat a delicious apple recommended by my friend at a restaurant, but I am disappointed" in English would be "Je mange une pomme délicieuse recommandée par mon ami au restaurant, mais je suis déçu".

At first glance, this unexpected correlation between languages seems puzzling. In the concrete space, the differences between different languages are large, but in the abstract space abstracted through meaning, the differences are small. This is to be expected as long as we are dealing with languages between humans on Earth.

In the above discussion, we have assumed that words are the basis, but other elements can also be used as the basis. For example, in the case of linguistic communication between humans and dolphins or whales, a communication medium and basis from a physically different perspective would be chosen.

In this direction, linguistics is likely to make great strides in the future.

2. Linguistic Geometry Space

Intuitively, the following equations seem to make sense:

USA-Washington = Japan-Tokyo

USA-Washington + Tokyo = Japan

Why? The abstractions of America and Japan are nations, and the abstractions of Washington and Tokyo are capitals, so at the abstract level, the following equations hold:

USA-Washington = nation-capital, Japan-Tokyo = nation-capital

→ US-Washington = Japan-Tokyo

The same goes for the following examples:

king-Man = Queen-Woman

king-Man+Woman = Queen

Such algebraic properties suggest that linguistic spaces have invariants and linearity, like the distance between two points in mathematical Euclidean space.

The simplest way to vectorize words is to arrange the words in a row and assign an N-dimensional one-hot vector from the beginning of the row:

$$(1,0,0,\ldots,0),\ (0,1,0,\ldots,0),\ \ldots,\ (0,0,0,\ldots,1)$$

where N is the number of words.

Words in a language have various relationships with each other. For example, they may have similar meanings, or they may be often used side by side, such as "cute baby." Although one-hot vectors can convert words into numeric vectors, they do not reflect the relationships between words at all. Therefore, this method is useless as it is.

To convert words into useful numeric vectors, the distributed representation of words described in references [2–4] is useful. An actual calculation method is word2vec. An explanation of the CBOW model, which is one of the word2vec models, is given in Appendix A. Below, we explain how to

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 March 2025 doi:10.20944/preprints202503.1909.v1

estimate the middle word of a sequence of three consecutive words from the two words to the left and right of it, using the idea of CBOW.

Each of the ten 10-dimensional one-hot vectors for the ten words (five fruit words and five animal words) shown in Figure 1 is compressed into ten 2- to 4-dimensional vectors using the neural network (NN) shown in Figure 3 by solving the fill-in-the-blank problem in Figure 2. In this way, the classification problem (feature extraction: fruit or animal) hidden in the fill-in-the-blank problem is solved.

Fruits:	apple, orange, banana, mango, pineapple
Animals:	dog, cat, tiger, lion, monkey

Figure 1. Two types of words: fruit and animal.

(apple, orange, banana),
(dog, cat, tiger),
⋮

Figure 2. Fill-in-the-blank question to guess the middle word from the two words on the left and right. (Choose 3 words from either fruits or animals).

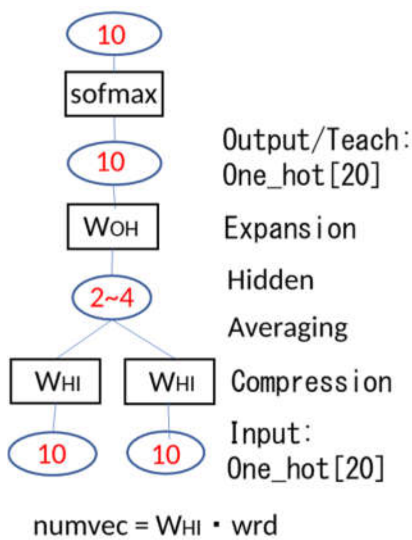


Figure 3. NN for fill-in-the-blank problems (red numbers indicate vector dimensions).

In the NN shown in Figure 3, two 10-dimensional one-hot vectors are given to the input layer, each of which is multiplied by the weight matrix W_{HI}, and an offset O_H is added, and then the average of the two is taken and becomes the input to the middle layer. The dimensions are compressed in the middle layer. Assuming a linear relationship between input and output, the input to the middle layer becomes the production of the middle layer as is. The output of the middle layer is multiplied by the weight matrix W_{OH} and an offset O_O is added, which becomes the input to the output layer and the production of the output layer as is. The dimensions are expanded in the output layer. The training data is given a single 10-dimensional one-hot vector used as the input data. Finally, it is converted to a probability by sofmax.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 March 2025 doi:10.20944/preprints202503.1909.v1

The structure of the neural network, i.e., the number of layers *NLYR*, the number of input cells *IU*, the number of hidden layer cells *HU*, and the number of output cells *OU*, are given in the SYSTEM.dat file. The parameters required for calculations are given in the PARAM.dat file. The number of learning iterations is specified by *times*.

Table 1. SYSTEM.dat file.

```
NLYR 3
IU 20
HU 2
OU 10
NT 40
```

Table 2. PARAM.dat file.

```
u0 0.5
intwgt 0.3
intoff 0.2
alpha 0.025
beta 0.025
moment 0.025
dmoment 0.002
erlimit 0.001
times 10000
```

We consider 40 pairs of three words in total, including 10 pairs taken only from fruits with no duplicates, 10 pairs taken only from animals in the same way, and 20 pairs created by swapping the two words on either side, for a total of 40 pairs (NT = 40).

Table 3. INPUT-TEACH.dat file.

```
0  apple  orange grape  0 1 2    10 dog    cat    tiger  5 6 7
1  orange grape  mango  1 2 3    11 cat    tiger  lion   6 7 8
2  grape  mango  banana 2 3 4    12 tiger  lion   monkey 7 8 9
3  mango  banana apple  3 4 0    13 lion   monkey dog    8 9 5
4  banana apple  orange 4 0 1    14 monkey dog    cat    9 5 6
5  apple  grape  banan  0 2 4    15 dog    tiger  monkey 5 7 9
6  orange mango  apple  1 3 0    16 cat    lion   dog    6 8 5
7  grape  banana orange 2 4 1    17 tiger  monkey cat    7 9 6
8  mango  apple  grape  3 0 2    18 liom   dog    tiger  8 5 7
9  banana orange mango  4 1 3    19 monkey cat    lion   9 6 8

20 grape  orange apple  2 1 0    30 tiger  cat    dog    7 6 5
21 mango  grape  orange 3 2 1    31 lion   tiger  cat    8 7 6
22 banana mango  grape  4 3 2    32 tiger  lion   monkey 7 8 9
23 apple  banana mango  0 4 3    33 dog    monkey lion   5 9 8
24 orange apple  banana 1 0 4    34 cat    dog    monkey 6 5 9
25 banana grape  apple  4 2 0    35 monkey tiger  dog    9 7 5
26 apple  mango  orange 0 3 1    36 dog    lion   cat    5 8 6
27 orange banana grape  1 4 2    37 cat    monkey tiger  6 9 7
28 grape  apple  mango  2 0 3    38 tiger  dog    liom   7 5 8
29 mango  orange banana 3 1 4    39 lion   cat    monkey 8 6 9
```

 doi:10.20944/preprints202503.1909.v1

When we calculate the fill-in-the-blank problem, we get the following interesting result in the dimensionally reduced hidden layer. The output cell of each cell is

input = output.

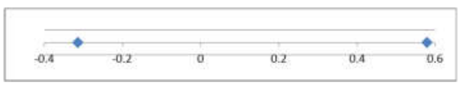
Therefore, this problem is linear. This gives linear algebraic properties to the number vector. The number vector *numvec* here is calculated as follows:

$$numvec[i] = \sum_{j=0}^{IU-1} W_{HI\,ij} w_j + O_H, \quad (i = 0, 1, \ldots, HU - 1),$$

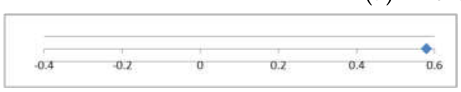
where, W_{HI} and O_H are the weights and offset between the hidden layer and the input layer, and the one-hot vectors of each word are given as input, respectively.

By solving the fill-in-the-blank problem, number vectors with a distributed word representation are obtained. The Figures 4-7 shows the learning results using the above input data and training data for the case where the number of training data pieces is NT = 40. This data has a hidden meaning of being classified into two categories, fruits and animals, and compact number vector representation is obtained from the one-hot vectors with this meaning reflected.

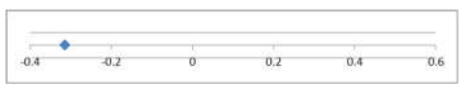
Figure 4 shows the number vector *numvec* when there is one hidden cell (HU = 1). We can see that the 40 data are classified into two categories, fruits and animals (two points on the line), according to the hidden meaning. However, since the fruit words and animal words are not separated, this representation cannot be used for translation or other purposes.



(a) Fruit+Animal



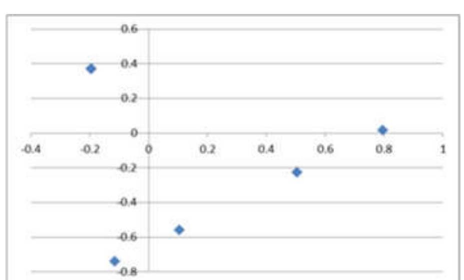
(b) Fruit



(c) Animal

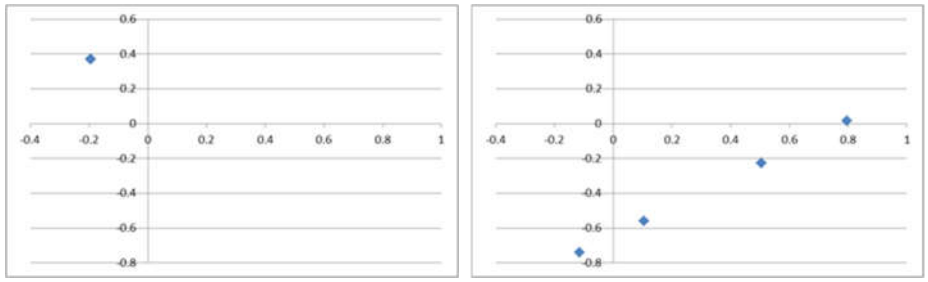
Figure 4. Numeric vector *numvec* for HU = 1.

Figure 5 shows the number vector *numvec* when there are two hidden cells (HU = 2). We can see that the 40 data points are classified into two categories: fruits and animals (two point groups on a plane). However, each fruit word is not separated, and the five animal words are reduced to four numeric vectors, so this representation is also unusable for purposes such as translation.



(a) Fruit+Animal

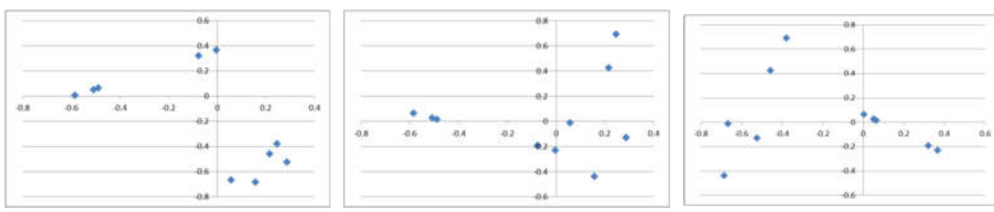
Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 March 2025 doi:10.20944/preprints202503.1909.v1



(b) Fruit (c) Animal

Figure 5. Numeric vector *numvec* for *HU* = 2.

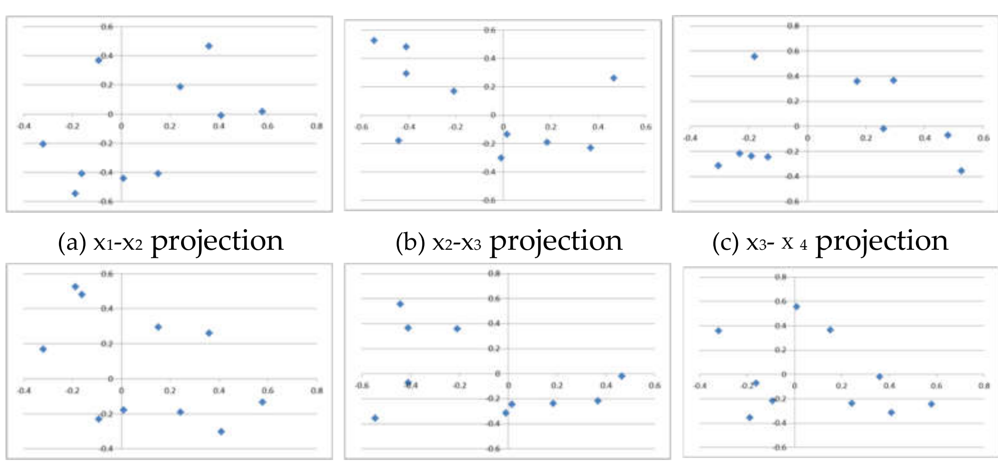
Figure 5 shows the number vector *numvec* when there are three hidden cells (*HU* = 3). It shows that the 40 data points are classified into two categories: fruits and animals (two point clouds in three-dimensional space). Furthermore, the components of the two point clouds for fruits and animals are completely separated. This representation seems usable for purposes such as translation.



(a) x-y projection (b) y-z projection (c) x-z projection

Figure 6. Numeric vector *numvec* for *HU* = 3.

Figure 6 shows the number vector *numvec* when there are four hidden cells (HU = 4). It can be seen that the 40 data points are classified into two categories: fruits and animals (two point clouds in four-dimensional space). Furthermore, the components of the two point clouds for fruits and animals are clearly separated. This expression seems to be useful for purposes such as translation.



(a) x_1-x_2 projection (b) x_2-x_3 projection (c) x_3- x_4 projection

(d) x_1- x_3 projection (e) x_2- x_4 projection (f) x_1- x_4 projection

Figure 7. Numeric vector *numvec* for *HU* = 4.

The number vector *numvec* for HU = 3 is shown in Table 4. From this result, for example, the inner product of apples and oranges (apple, orange), the inner product of dogs and cats (dog, cat), and the inner product of apples and dogs (apple, dog) are

$$(apple, orange) = (-0.584)(-0.003) + (0.004)(0.367) + (0.065)(-0.230) = -0.01173,$$

$$(apple, dog) = (-0.584)(0.287) + (0.004)(-0.524) + (0.065)(-0.129) = -0.17810,$$

$$(dog, cat) = (0.287)(0.057) + (-0.524)(-0.667) + (-0.129)(-0.013) = 0.36755.$$

The correlation between fruits and animals is smaller than the correlation between fruits and fruits and the correlation between animals and animals.

If we introduce Euclidean distance as the distance between words, for example, the distance *d*(*apple, orange*) between apples and oranges and the distance *d*(*dog, cat*) between dogs and cats are calculated as

$$d(apple, orange) = \sqrt{(-0.584 + 0.003)^2 + (0.004 - 0.367)^2 + (0.065 + 0.230)^2} = 0.746,$$

$$d(dog, cat) = \sqrt{(0.287 - 0.057)^2 + (-0.524 + 0.667)^2 + (-0.129 + 0.013)^2} = 0.293.$$

In addition, the average position of fruits (x_f, y_f, zf) and the average position of animals (x_a, y_a, z_a) are calculated as

$$\begin{aligned}
x_f &= (-0.584 - 0.003 - 0.508 - 0.752 - 0.488) / 5 = -0.467, \\
y_f &= (+0.004 + 0.367 + 0.051 + 0.321 + 0.064) / 5 = +0.161, \\
z_f &= (+0.065 - 0.230 + 0.027 - 0.193 + 0.016) / 5 = -0.063, \\
x_a &= (+0.287 + 0057 + 0.158 + 0.216 + 0.246) / 5 = +0.193, \\
y_a &= (-0.524 - 0.667 - 0.685 - 0.457 - 0.378) / 5 = -0.542, \\
z_a &= (-0.129 - 0.013 - 0.440 + 0.425 + 0.692) / 5 = +0.107.
\end{aligned}$$

The distances between the average positions *d*(*fruit, animal*) is calculated as

$$d(fruit, animal) = \sqrt{(-0.467 - 0.193)^2 + (0.161 + 0.542)^2 + (-0.063 - 0.107)^2} = 0.979.$$

Table 4. Number vector *numvec* for *HU* = 3.

k:	word	x	y	z
0	apple	-0.584172	0.00403	0.065173
1	orange	-0.002623	0.366662	-0.229709
2	grape	-0.508273	0.051358	0.026687
3	mango	-0.075167	0.321426	-0.192925
4	banana	-0.487712	0.064179	0.016262
5	dog	0.287074	-0.52356	-0.128992
6	cat	0.057498	-0.666714	-0.012583
7	tiger	0.157732	-0.684748	-0.439977
8	lion	0.215578	-0.45749	0.424644

 doi:10.20944/preprints202503.1909.v1

9	monkey	0.245844	-0.37816	0.691991

Although it cannot be confirmed by simple learning with a small amount of data as described above, the number vectors based on distributed word representations by word2vec can realize very interesting algebraic properties as described below. Figure 8 shows the algebraic properties of such number vectors:

King – Man + Woman = Queen

Queen – Woman + Man = King

Figure 8 illustrates the reason. This is probably because in learning methods like word2vec, the implicit relationships contained in the input data:

King = Head_of_State + Man

Queen = Head_of_State + Woman

are also learned during the explicit learning process of fill-in-the-blank questions, and these relationships are accumulated in the weights between the input layer and the hidden layers.

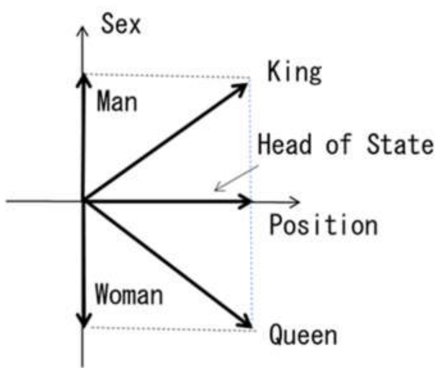


Figure 8. Algebraic properties.

Let's consider this using the following data. For five words, Head (Head of State), King, Man, Queen, and Woman, we use the following three types of INPUT-TEACH.dat files:

Table 5. INPUT-TEACH.dat file (OU=5, HU=2, IU=10, NT=4).

0	head	king	man	0 1 2
1	head	queen	woman	0 3 4
2	man	king	head	2 1 0
3	woman	queen	head	4 3 0

Table 6. INPUT-TEACH.dat file (OU=5, HU=2, IU=10, NT=12).

0	head	king	man	0 1 2
1	king	man	head	1 2 0
2	man	head	king	2 0 1
3	head	queen	woman	0 3 4

 doi:10.20944/preprints202503.1909.v1

4	queen	woman	head	3 4 0
5	woman	head	queen	4 0 3
6	man	king	head	2 1 0
7	head	man	king	0 2 1
8	king	head	man	1 0 2
9	women	queen	head	4 3 0
10	head	woman	queen	0 4 3
11	queen	head	woman	3 0 4

Table 7. INPUT-TEACH.dat file (OU=5, HU=2, IU=10, NT=16).

0	head	king	man	0 1 2
1	king	man	head	1 2 0
2	man	head	king	2 0 1
3	head	queen	woman	0 3 4
4	queen	woman	head	3 4 0
5	woman	head	queen	4 0 3
6	man	king	head	2 1 0
7	head	man	king	0 2 1
8	king	head	man	1 0 2
9	women	queen	head	4 3 0
10	head	woman	queen	0 4 3
11	queen	head	woman	3 0 4
12	king	head	queen	1 0 3
13	man	head	woman	2 0 4
14	queen	head	king	3 0 1
15	woman	head	man	4 0 2

The data includes information about the heads of state, including the King and Queen, and the genders of the King and Queen, Man and Woman. Therefore, the interesting relationships between titles (King, Queen) and job titles (Head of state) and gender (Man, Woman) mentioned above can be derived.

For NT = 4, NT = 12, and NT = 16, the distributed representation of the number vectors by the NN for the fill-in-the-blank problem in Figure 3 is as follows:

Table 8. Number vector (NT = 4).

	Head	King	Man	Queen	Woman
x	-0.01387	0.185897	0.635359	0.33299	-0.65818
y	0.049428	-0.1526	0.785826	0.130928	-0.69117

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 March 2025 doi:10.20944/preprints202503.1909.v1

Table 9. Number vector (NT = 12).

	Head	King	Man	Queen	Woman
x	-0.5415	0.767788	-0.08519	0.327094	-0.1742
y	-0.5789	-0.27271	0.482651	0.117548	0.561467

Table 10. Number vector (NT = 12).

	Head	King	Man	Queen	Woman
x	-0.54611	0.703868	-0.06764	0.307364	-0.14322
y	-0.74313	-0.06512	0.469316	0.209544	0.521668

From these vectors, we perform the algebraic operation King - Man + Woman.

Table 11. Algebraic operations King - Man＋Woman.

	NT = 4	NT = 12	NT = 16
x	-1.10764	0.678785	0.628294
y	-1.6296	-0.19389	-0.01277

When calculating the correlation coefficient ρ :

$$\rho(X,Y)=\frac{(X,Y)}{\sqrt{(X,X)}\sqrt{(Y,Y)}}$$

between King - Man + Woman and each of the words Head, King, Man, Queen, and Woman, it is seen that in the cases of NT = 12 and NT = 16, where the relationships between each word are sufficiently embedded, the correlation with Queen is large, except for King. In NT = 16, the correlation coefficients with Queen, Man, and Woman are higher than in NT = 12, and the correlation coefficients with King and Head are lower.

Table 12. Correlation coefficients between King - Man + Woman and Head, King, Man, Queen, and Woman.

	Head	King	Man	**Queen**	Woman
NT = 4	-0.64441	0.090258	-0.99656	**-0.82579**	0.986585
NT = 12	-0.45679	0.997775	-0.43692	**0.812236**	-0.54657
NT = 16	-0.57570	0.997461	-0.16275	**0.814677**	-0.28429

Similarly, if we perform the algebraic operation Queen - Woman + Man from these vectors, the following results are obtained:

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 March 2025 doi:10.20944/preprints202503.1909.v1

Table 13. Algebraic operations Queen - Woman + Man.

	NT = 4	NT = 12	NT = 16
x	1.626527	0.416097	0.382938
y	1.607926	0.038732	0.157192

Calculating the correlation coefficient ρ between Queen - Woman + Man and each word Head, King, Man, Queen, and Woman, we can see that in the cases of NT = 12 and NT = 16, where the relationships between each word are sufficiently embedded, the correlation with King is large, except for Queen. In NT = 16, the correlation coefficient with Queen, Man, and Woman is higher than in NT = 12, and the correlation coefficient between King and Head is lower.

Table 14. Correlation coefficients between Queen - Woman + Man and Head, King, Man, Queen, and Woman.

	Head	**King**	Man	Queen	Woman
NT = 4	0.484746	**0.10361**	0.99382	0.919092	-0.99954
NT = 12	-0.74787	**0.907247**	-0.0818	0.96837	-0.20652
NT = 16	-0.85381	**0.886176**	0.243888	0.978269	0.121283

The mathematics of translating from English to French will be explained using a simple example:

"delicious apple" → "pomme délicieuse",

"I ate a delicious apple" → "Je mange une pomme délicieuse".

Roughly speaking, when a human translates, they follow the following steps: first, they look up the dictionary and translate word by word, then they rearrange the word order. This can be expressed mathematically as follows:

$$\begin{pmatrix} delicious \\ apple \end{pmatrix} \xrightarrow{Dictioonary} \begin{pmatrix} d?licieuse \\ pomme \end{pmatrix} \xrightarrow{Grammer} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} d?licieuse \\ pomme \end{pmatrix} \rightarrow \begin{pmatrix} pomme \\ délicieuse \end{pmatrix},$$

$$\begin{pmatrix} I \\ ate \\ a \\ delicious \\ apple \end{pmatrix} \xrightarrow{Dictioonary} \begin{pmatrix} Je \\ mange \\ une \\ d?licieuse \\ pomme \end{pmatrix} \xrightarrow{Grammer} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} Je \\ mange \\ une \\ d?licieuse \\ pomme \end{pmatrix} \rightarrow \begin{pmatrix} Je \\ mange \\ une \\ pomme \\ d?licieuse \end{pmatrix}.$$

When we consider this together with the numerical vector representation of words by word2vec mentioned above, we can replace translation with computation. This is why we think of translation as coordinate transformation.

Let a sentence S_{A1} in the concrete space of a language A be mapped to a point $T_A S_{A1}$ in the semantic space by a transformation T_A from the concrete space A to the semantic space.

It seems that by introducing an abstract space, we can uniquely define distance. Thinking in this way, we can introduce scalar distance that is independent of the coordinate system into linguistic space, which becomes a traditional geometric space:

Distance in concrete space = Distance in abstract space,

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 March 2025 doi:10.20944/preprints202503.1909.v1

Namely,

The distance between sentence S_{A1} and sentence S_{A2} in concrete space A = $\left\| T_A S_{A1} - T_A S_{A2} \right\|$.

To turn mathematical linguistic space into a physical space, we will need a mathematical expression of the state of the brain.

3. Mathematical Representation of the State of the Brain

If the state of brain cell i is represented as $\left|\psi_i\right\rangle$, and assuming binary nature and the basis as $\left|0\right\rangle, \left|1\right\rangle$, then the state of the brain cell can be represented as

$$\left|\psi_i\right\rangle = c_{i0}\left|0\right\rangle + c_{i1}\left|1\right\rangle = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix}$$

[1], where, as in quantum mechanics, the square of the absolute value of the coefficient represents the probability.

The state of the entire cerebrum, $\left|\Psi\right\rangle$, can be written as

$$\begin{aligned} \left|\Psi\right\rangle &= \left|\psi_0\right\rangle \otimes \left|\psi_1\right\rangle \otimes \cdots \otimes \left|\psi_N\right\rangle = \prod_{i=0}^{2N-1} \left(c_{i0}\left|0\right\rangle + c_{i1}\left|1\right\rangle \right) \\ &= c_0\left|00\cdots 0\right\rangle + c_1\left|00\cdots 1\right\rangle + \cdots + c_{2N-1}\left|11\cdots 1\right\rangle \\ &= c_0\left|0\right\rangle + c_1\left|1\right\rangle + \cdots + c_{2N-1}\left|2N-1\right\rangle = \left(c_0, c_1, \cdots, c_{2N-1}\right)^T \end{aligned}$$

if it is the direct product of the individual cells.

On the other hand, the mathematical representation of a linear language space

$$\left|America\right\rangle - \left|Washington\right\rangle = \left|Japan\right\rangle - \left\langle Tokyo\right\rangle$$

$$\left|America\right\rangle - \left|Washington\right\rangle + \left|Tokyo\right\rangle = \left|Japan\right\rangle$$

holds.

If we consider a word to be the smallest unit of a sentence and a word to be the basis of a sentence, then it will be

$$\left|Word\right\rangle = \left|00\cdots 0\right\rangle, \left|00\cdots 1\right\rangle, \left|00\cdots 10\right\rangle, \cdots, \left|11\cdots 1\right\rangle,$$

$$\begin{aligned} \left|Sentence\right\rangle &= c_0\left|00\cdots 0\right\rangle + c_1\left|00\cdots 1\right\rangle + c_2\left|00\cdots 10\right\rangle + \cdots + c_{2^N-1}\left|11\cdots 1\right\rangle \\ &= c_0\left|0\right\rangle + c_1\left|1\right\rangle + c_2\left|2\right\rangle + \cdots + c_{2^N-1}\left|2^N-1\right\rangle = \left(c_0, c_1, c_2, \cdots, c_{2^N-1}\right)^T. \end{aligned}$$

The numerical vector representation of words described in the previous section corresponds to the above basis. In this way, the physical cellular space of the brain and the mathematical language space can be corresponded.

If we consider the process of translation from the above perspective,

translation from English to Japanese

= coordinate transformation from English space to Japanese space.

In other words, it can be thought of as a coordinate transformation between different language spaces:

$$\left|Japanese\ sentence\right\rangle = \left(Transformation\ matrix\right) \times \left|English\ sentence\right\rangle.$$

 doi:10.20944/preprints202503.1909.v1

Then, we can think as

Words = basis vectors in language space = eigenfunctions of an eigenvalue problem of the time-independent Schrödinger equation.

The evolution of the state of the brain over time is given by a time-dependent Schrödinger equation:

$$i\hbar \frac{\partial |\psi(t)\rangle}{\partial t} = \hat{H}|\psi(t)\rangle ,$$

and consideration may also arise from the aspect of the energy minimization problem.

In other words, the stability of memory can be thought of as the state of the brain having multiple stable states, each of which is assigned a word, and sentences being formed by combining multiple stable states.

4. Conclusion

Linguistics is an important academic field with a long history. Still, with the introduction of generative AI such as ChatGPT, it seems to be on the verge of achieving groundbreaking developments.

The introduction of generative AI has revealed clear geometric similarities between different languages. In hindsight, this may seem obvious, but it is a major discovery that exceeded expectations.

We can also expect groundbreaking advances in communication between different species of animals, such as humans and dogs, dolphins, and whales. As communication between humans and animals progresses, it will be required for changing the way humans treat animals significantly.

 doi:10.20944/preprints202503.1909.v1

By training using this CBOW model, it is possible to acquire a distributed representation of words. If one word before and after is used as the context, for example, in the following example (fill-in-the-blank question):

(I) (am) (always) (studying) (English) (recently) (.)

the (English) word is inferred from (studying) and (recently).

Then, we consider a fill-in-the-blank problem in which the t-th word in a sentence is estimated from a sequence of c words before and after it:

$$w_{t-c},\cdots,w_{t-2},w_{t-1} \text{ and } w_{t+1},w_{t+2},\cdots,w_{t+c}$$

Although we are trying to find a word embedding, we are trying to solve a fill-in-the-blank problem that may seem unrelated at first glance. However, solving this problem actually results in finding the word embedding. The Softmax part or its input becomes the training data.

The one-hot vector of each word is used as input. The training data can be interpreted as providing data (probabilities) in which the t-th element is 1 and the others are 0. Alternatively, the one-hot vector of the answer is used. The word distributed representation is calculated as follows:

$$\textit{The distributed representation of } w_t = W_I w_t .$$

References

1. H. Isshiki, Human Thoughts and Quantum Mechanics (predictive study through mathematical model), Scientific & Academic Publishing, (2025) To be published
2. Tomas Mikolov, Kai Chen, Greg Corrado & Jeffrey Dean, Efficient estimation of word representation in vector space,Cornell arXiv:1301.3781v3, 2013
3. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado & Jeffrey Dean, Distributed Representations of words and prases and their compositionality, Cornell arXiv:1310.4546v1, 2013
4. WikipediA, word2vec,(https://en.wikipedia.org/wiki/Word2vec)
5. Y. LeCun, Y. Bengio & G. Hinton, Deep learning, Nature, vol. 521, 2015, 436-444
6. A. Radford, T. Salimane & I. Sutskever, Improving Lanuag Understanding by Generative Pre-traing, 2018. (https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf)
7. A. Redford, J. W. R. Child, D. Luan, D. Amodei, & I. Sitskever, Language Models are UnsupervisedMultitask Lerners, OpenAI, Preprint 2019. (https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf)
8. J. Kaplan, S. McCandish, T. Henighan, T. B. B. Chess, C. Rewond, S. Gray, A. Radfold, J. Wu, & D. Amodei, Scaling Laws for Neural Language Models, Cornell University, arXiv:2001.08361, 2020.
9. Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q. & V. Le, XLNet: Generalized Autoregressive Pre Training for Language Understanding, Cornell University, arXiv:1906.08237, 2020.
10. C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li & P. J. Liu, Exploring the Limit of Transfer Learning with a Unified Text-to-Text Transformer, Cornell University, arXiv:1910.10683, 2023.
11. Masato Hagiwara, Benjamin Hoffman, Jen-Yu Liu, Maddie Cusimano Felix Effenberger, Katie Zacarian, BEANS: The Benchmark of Animal Sounds, Cornell ArXiv:22110.12300v1, 2022
12. Masato Hagiwara, AVES: Animal Vocalization Encoder Based on Self-Supervision, Cornell ArXiv: 2210.14493v1, 2022
13. Christian Rutz, Michael Bronstein, Aza Raskin, Sonja C. Vernes, Katherine Zacarian, Damián E. Blasi, Using machine learning to decode animal communication, Science, 2023
14. Using AI to Decode Animal Communication with Aza Raskin, 2024. (https://www.youtube.com/watch?v=3tUXbbbMhvk)