

# Word2Vec 学習ネットワークの性質を考慮した 分散表現の提案

内田 脩斗<sup>1,a)</sup> 吉川 大弘<sup>1,b)</sup> 古橋 武<sup>1,c)</sup>

**概要:** Word2Vec は、単語の分散表現を獲得する最も一般的な手法の一つであり、自然言語処理分野における構文解析や文書分類などに適用した研究も数多く報告され始め、その有用性が示唆されている。一般的に、分散表現を表しているのは Word2Vec 学習ネットワーク上の入力層-隠れ層間で学習される重み行列であり、これを実際に利用することになる。一方、学習の過程で同時に生成される隠れ層-出力層間の重み行列は一般的には利用されず、またこの重み行列に関する研究報告は数少ない。これに対し著者らは、入力側重み行列と対となる出力側重み行列の解釈について、Word2Vec の学習過程から考察し、本論文において、前者と後者を併用した単語ベクトル 2 種を提案する。また、意味関係テストを用いた単語ベクトルの性能評価実験を行い、提案手法の有効性を検証し、結果について考察する。加えて、単語ベクトルの応用例として、提案した単語ベクトルを用いた文書分類実験を行い、従来の分散表現よりも分類性能が向上することを示す。

**キーワード:** Word2Vec, 分散表現, 出力側の重み, 意味関係テスト, 文書分類

## A Proposal of Distributed Representation Considering Features of Learning Network on Word2Vec

SHUTO UCHIDA<sup>1,a)</sup> TOMOHIRO YOSHIKAWA<sup>1,b)</sup> TAKESHI FURUHASHI<sup>1,c)</sup>

**Abstract:** Word2Vec is one of the most common methods for acquiring a distributed representation of words. In the field of natural language processing, many studies applying Word2Vec to syntactic analysis and document classification have been reported and its usefulness is suggested. Generally, distributed representation indicates the learned weight matrix between the input layer and the hidden layer on a Word2Vec learning network, and this is actually used. On the other hand, the weight matrix between the hidden layer and the output layer is simultaneously generated in the learning process, which is usually not used. There are few research reports on this weight matrix. The authors consider the interpretation of the output side weight matrix paired with the input side one in the learning process of Word2Vec, and we propose two kinds of word vectors using the former and the latter together in this paper. We conduct experiments on performance evaluation of word vectors using semantic relation tests, and study the effectiveness of the proposed word vectors. In addition, as an application example of word vector, we carry out a document classification experiment using the proposed word vectors and show that classification performance is improved more than the conventional distributed representation.

**Keywords:** Word2Vec, distributed representation, output embedding, semantic relation tests, document classification

<sup>1</sup> 名古屋大学大学院工学研究科  
Graduate School of Engineering Nagoya University, Nagoya,  
Aichi, 464-8603, Japan

a) uchida@cplx.cse.nagoya-u.ac.jp

b) yoshikawa@cse.nagoya-u.ac.jp

### 1. はじめに

インターネットの普及に伴い、膨大な情報が生成・拡散

c) furuhashi@cse.nagoya-u.ac.jp

されている現代において、テキストデータの自動解析や情報抽出技術は、様々な場面での応用が期待されており、研究が進められている。多くの場合、言語処理分野においては、単語を原子単位（要素）として取り扱っており、その表現方法は根源的な問題として扱われている。従来から広く普及している単語の表現手法として、個々の単語に固有のインデックスを与えることで単語を表現する One-hot 表現がある [1][2]。この手法は非常にシンプルで分かりやすい反面、各単語が独立であることを前提としているため、同義語や類似語が全く関係のない単語として扱われることがある。また、1単語に1次元を割り当てるため、ボキャブラリ数が増えれば増えるほど高次元となり、計算コストがかかるという問題がある。これを解決するために、単語ベクトルを生成する手法が数多く研究されているが [3][4][5][6]、Micolov らが発表した Word2Vec [7] は、大規模コーパスから教師なし学習を行うことで、自動的に語義の似た単語が類似したベクトルを持つ分散表現と呼ばれる単語ベクトルを生成することができる。これにより、従来では困難であった単語ベクトル間での意味の演算が可能となり、例えば、学習された分散表現に対し、

$$\text{vector}(\text{Paris}) - \text{vector}(\text{France}) + \text{vector}(\text{Italy})$$

により算出されるベクトルが  $\text{vector}(\text{Rome})$  に、

$$\text{vector}(\text{king}) - \text{vector}(\text{man}) + \text{vector}(\text{woman})$$

により算出されるベクトルが  $\text{vector}(\text{queen})$  に、それぞれ近くなるという性質を持っている。この分散表現を適用した研究が多く報告され始めており [8][9][10][11]、分散表現の意味演算性能を高めることは極めて有益であると考えられる。

一般的に、分散表現とは Word2Vec 学習ネットワーク上の入力層-隠れ層間で学習される重み行列（以下、「入力側の重み」と呼ぶ）のことを意味しており、これを実際に利用することになる。また、学習の過程で同時に生成される隠れ層-出力層間の重み行列（以下、「出力側の重み」と呼ぶ）は利用されない。しかし、この出力側の重みは単語に対して入力側の重みとは異なる意味関係を捉えていると考えられ、その有用性があると考えられる。

本論文では、入力側の重みと出力側の重みの捉えている意味関係の違いについて検討し、両者を併用した単語ベクトルを提案する。また、意味関係性テストによる単語ベクトルの性能評価実験を行い、その有効性を検証する。さらに、提案した単語ベクトルを用いた文書分類実験を行い、従来法よりも分類性能が向上することを示し、提案手法の有用性を示す。

## 2. Word2Vec

本章では、Word2Vec の学習モデルについて説明する。

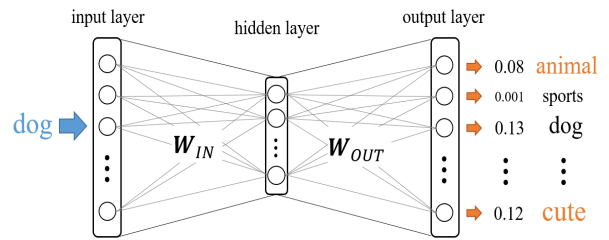


図 1 Word2Vec の学習モデル図  
Fig. 1 Learning model of Word2Vec.

図 1 は、Word2Vec の学習モデルを表した図である。このとき、入力層と出力層の次元数はボキャブラリ数、隠れ層は埋め込み次元数を表している。このように、Word2Vec は、ニューラルネットワークを用いて分散表現を自動的に獲得する。Word2Vec は、言語処理分野でよく用いられる分布仮説（同じ文脈で出現する単語は同じ意味を持つこと）[12] に基づいており、文脈上のある単語に対して、周辺に現れやすい単語を予測することをモデル化した構造をしている。つまり、入力層に入力された単語に対して、文脈上でその単語の周辺に出現している単語の出現確率が大きくなるように各層の重み（入力側の重み  $W_{IN}$ 、出力側の重み  $W_{OUT}$ ）を更新する。例えば、ある文脈上において、「dog」という単語の周辺に「animal」や「cute」という単語が出現したとすると、入力層の「dog」に対応した要素に 1、それ以外の要素に 0 が入力され、「animal」、「cute」に対応した出力層の出力が 1 に近づくように  $W_{IN}$  と  $W_{OUT}$  が更新される。それぞれの更新式は、(2)(3) 式ようになる。

$$p_j = \frac{\exp(W_{Ith} \cdot W'_{jth})}{\sum_{v \in vocab} \exp(W_{Ith} \cdot W'_{vth})} \quad (1)$$

$$W_{Ii}^{(new)} = W_{Ii}^{(old)} - \eta \sum_{v \in vocab} (p_v - t_v) W'_{iv} \quad (2)$$

$$W'_{ij}{}^{(new)} = W'_{ij}{}^{(old)} - \eta(p_j - t_j)W_{Ii} \quad (3)$$

$W_{Ith}$  は  $I$  番目の入力単語の行ベクトル、 $W'_{jth}$  は  $j$  番目の出力単語の列ベクトル、 $i$  は各ベクトルの  $i$  番目の重み、 $\eta$  は学習率、 $t_j$  は  $j$  番目の出力単語が  $I$  番目の入力単語に対して正解単語か否かのフラグを表している。また (1) 式は、 $I$  番目の入力単語と  $j$  番目の出力単語の共起確率を表している。なお、ここでは  $W_{IN}$  を  $W$ 、 $W_{OUT}$  を  $W'$  で表記している。

入力側の重み  $W_{IN}$  は周辺単語の情報を元に、単語の意味関係を学習している。更新式は (2) 式のようになり、「dog」や「cat」という単語の周辺に「animal」や「cute」という単語が共通して出現することで類似した重みを学習し、 $\text{vector}(\text{dog}) \simeq \text{vector}(\text{cat})$  に近づくように重みの更新が行われる。よって、 $W_{IN}$  は個々の次元に単語の意味関係を分散して埋め込んでいる形となり、分散表現と呼ばれる。また、 $W_{IN}$  の各行ベクトルが個々の単語ベクトルと

対応しており、Word2Vec では  $W_{IN}$  を分散表現として利用することを前提とした学習モデルとなっていることがわかる。また、学習アルゴリズムの詳細については、[13] を参照されたい。

一方、出力側の重み  $W_{OUT}$  は、ある単語の周辺に出現する単語を予測するための重みであるといえる。更新式は (3) 式のようになり、「dog」や「cat」という単語の周辺に「animal」や「cute」という単語が現れる確率を大きくするため、入力単語のベクトルと周辺単語のベクトルの内積値が大きくなるように重みを学習する。これより、 $W_{OUT}$  は、単語ベクトルを共起しやすい単語に展開する共起単語ベクトルであると捉えることができる。つまり、 $W_{OUT}$  においては、 $vector(animal) \simeq vector(cute)$  のような共起関係を学習する傾向にある。また、 $W_{OUT}$  の各列ベクトルが個々の共起単語ベクトルに対応している。

ただし、 $W_{OUT}$  は  $W_{IN}$  を獲得するために生成される副産物的な重みとされており、一般的に  $W_{OUT}$  を単語ベクトルとして利用することはない。 $W_{OUT}$  に関する関連研究として、Mitra ら [14] は、 $W_{IN}$  と  $W_{OUT}$  から得られる共起情報に注目し、文書検索に適用した手法を提案している。また、Press ら [15] は、 $W_{OUT}$  が  $W_{IN}$  に意味性能の面で劣っていることを報告している。しかし、 $W_{OUT}$  を単語ベクトルとして、独立して用いる研究はこれまで報告されていない。

本研究では、 $W_{IN}$  と  $W_{OUT}$  を融合し、新たな単語ベクトルを生成する手法を提案する。上述したように、 $W_{IN}$  と  $W_{OUT}$  では学習される重みの性質が異なっており、両者を融合することで、より性能の高い単語ベクトルとなることを期待する。

### 3. 提案手法

本章では、 $W_{IN}$  と  $W_{OUT}$  を融合した新たな単語ベクトルの生成手法を提案する。

#### 3.1 連結型単語ベクトル

複数の分散表現を結合する手法の一つに、Yin らが提唱した連結型単語ベクトル [16] がある。具体的には、複数のコーパスを用いて独立に分散表現を獲得し、ベクトルの次元を拡張してそれぞれの分散表現を連結する。例えば、次元数が 100, 50, 300 の 3 種の分散表現を生成したとき、次元数  $k = 100 + 50 + 300 = 450$  となる分散表現を新たに生成する。これにより、単語ベクトルの表現力の拡張と単語のカバレッジの向上が期待できると報告している。本論文では、この手法を単一のコーパスから生成される  $W_{IN}$  と  $W_{OUT}$  に適用し、 $W_{CONC}$  を生成する。これにより、 $W_{IN}$  と  $W_{OUT}$  の捉えているそれぞれの特徴を含有したベクトルが生成されると考えられる。

#### 3.2 加算平均型単語ベクトル

ここでは、(4) 式を用いて、加算平均型単語ベクトル  $W_{MEAN}$  を生成する。

$$\overrightarrow{W_{MEAN\_word}} = \frac{1}{2}(\overrightarrow{word_{IN}} + \overrightarrow{word_{OUT}}) \quad (4)$$

$W_{MEAN\_word}$  は、新しく生成される単語ベクトルを表している。また、 $W_{MEAN}$  では単語ベクトルの次元数に変化はない。 $W_{IN}$  と  $W_{OUT}$  の加算平均をとることで、両者の捉えているそれぞれの特徴を含有したベクトルが生成されると考えられる。

### 4. 実験

本章では、[6] で紹介されている手法を用いて、単語ベクトルの性能評価を行う。

#### 4.1 統語論的意味関係テスト

統語論的意味関係テストとは、単語ベクトルがどの程度単語間の意味関係を捉えられているかを評価する方法である。

本実験では、マイクロソフト社が公開している MSR Word Relatedness Test Set<sup>\*1</sup> を用いた。上記のデータセットを、表 1 に示す。Category は単語の品詞、Relation は単語間の関係性、Patterns Tested は単語に付与されたタグ、#Questions はテストセット数、Example はそれに含まれる一例を示している。これより、このテストセットには、文法的な意味関係を反映した単語セットが計 8000 セット含まれている。

#### 4.2 評価方法

- (1)  $a : b, c : d$  という関係性の単語セットであるとき、 $d$  を未知データとする。
- (2) 各々の分散表現  $W$  内の  $\vec{a}, \vec{b}, \vec{c}$  を利用し、 $\vec{y} = \vec{b} - \vec{a} + \vec{c}$  を算出する。
- (3)  $\vec{y}$  と  $\vec{d}$  の Cos 類似度と他単語ベクトルとの Cos 類似度を比較することで、単語ベクトルの性能を測定する。

また、本実験では、単語ベクトルが存在しない単語が含まれている単語セットの場合は、除外して評価を行う。

#### 4.3 Word2Vec 学習条件

Word2Vec の学習には英語 Wikipedia を用いた。また、各種パラメータは、 $window = 5, size = 300$  とした。「 $window$ 」は、前後何単語を教師データとするかを指定するオプション、「 $size$ 」は、学習する単語ベクトルの次元数を指定するオプションである。Word2Vec の実装は Python の gensim ライブラリを使用し、また、 $W_{OUT}$  はライブラリ内の syn1neg に保存されているものを用いた。

<sup>\*1</sup> <https://www.microsoft.com/en-us/research/project/recurrent-neural-networks-for-language-processing/>

表 1 意味関係単語セット

Table 1 MSR Word Relatedness Test Set.

Category	Relation	Patterns Tested	# Questions	Example
Adjectives	Base/Comparative	JJ/JJR, JJR/JJ	1000	good:better rough:---
Adjectives	Base/Superlative	JJ/JJS, JJS/JJ	1000	good:best rough:---
Adjectives	Comparative/Superlative	JJR/JJS, JJS/JJR	1000	better:best rougher:---
Nouns	Singular/Plural	NN/NNS, NNS/NN	1000	year:years law:---
Nouns	Non-possessive/Possessive	NN/NN_POS, NN_POS/NN	1000	city:city's bank:---
Verbs	Base/Past	VB/VBD, VBD/VB	1000	see:saw return:---
Verbs	Base/3rd Person Singular Present	VB/VBZ, VBZ/VB	1000	see:sees return:---
Verbs	Past/3rd Person Singular Present	VBD/VBZ, VBZ/VBD	1000	saw:sees returned:---

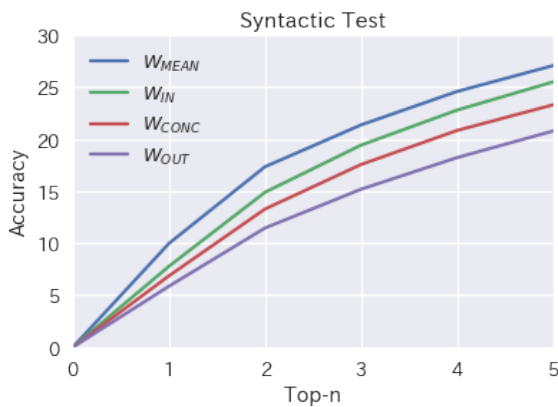


図 2 統語論的意味関係テスト  
 Fig. 2 Syntactic Test.

#### 4.4 実験結果

図 2 に、統語論的意味関係テストの結果を示す。評価できない単語セットを除いたところ、評価可能な単語セット数は計 6820 セットであった。また、 $\bar{y}$  の算出に利用した単語 ( $\vec{a}$ ,  $\vec{b}$ ,  $\vec{c}$  に相当するもの) は比較対象から除外した。Top-n は、上位何単語まで許容するかを表すパラメータである。よって、Accuracy は、Top-n 単語内に正解単語が出現する割合を表している。また、Word2Vec は内部にランダム性を保持しているため、今回は 5 試行平均での結果を表示している。

図 2 より、 $W_{IN}$  と  $W_{OUT}$  では、 $W_{IN}$  の方が精度が高いことが確認できる。 $W_{OUT}$  は  $W_{IN}$  と比較して、相対的に共起関係を含有しているものの、単語ベクトルとしての性能は  $W_{IN}$  よりも低いと考えられ、[15] での報告と一致している。また、 $W_{CONC}$  は、 $W_{IN}$  と  $W_{OUT}$  の中間を推移しており、従来の“分散表現”である  $W_{IN}$  よりも精度が低いことが確認できる。これにより、 $W_{CONC}$  は両者の異なる意味関係の捉え方を平均化したベクトルになっていると考えられるが、単語ベクトルとしての性能向上は期待できないことがわかった。また、図 2 において、 $W_{MEAN}$  が、最も単語ベクトルとしての性能が高いことが確認でき、提案手法の有効性を示唆している。

以上の結果を考察するために、まず単語ベクトル間の角

表 2 各単語ベクトルにおける Cos 類似度平均

Table 2 Cosine similarity in each word vector.

単語ベクトル	$W_{IN}$	$W_{OUT}$	$W_{CONC}$	$W_{MEAN}$
平均値	0.16	0.21	0.19	<b>0.052</b>

度の違いに注目した。意味関係が良く表現できるベクトルは、他単語との区別が明瞭であると考えられる。このことは、単語ベクトル間の角度の大きさで表現が可能である。そこで、意味関係テストで用いた 930 単語を対象とし、それぞれの単語ベクトルで総当たりによる単語ベクトル間の Cos 類似度を算出し、その平均値を求めた。表 2 に、結果を示す。

単語ベクトル間の角度の大きさは、ベクトル空間をどれだけ大きく利用しているかを表していると考えられる。そのため、Cos 類似度の平均値は、小さい方が単語間の角度が大きく、良い単語ベクトルであると考えられる。表 2 より、各値の大小関係と意味関係テストの結果の大小関係が一致していることが確認できる。これにより、単語ベクトル性能と単語ベクトル間角度の大きさには密接な関係があることが確認された。また、 $W_{MEAN}$  についても、単語ベクトル同士の意味的な区別がしやすくなったことが、意味関係性能が向上した要因の一つとなったと考えられる。

$$Cos\_sim = \overline{y_{MEAN}} \cdot \overline{d_{MEAN}} \quad (5)$$

$$= \left\{ \frac{1}{2} (\overline{y_{IN}} + \overline{y_{OUT}}) \right\} \cdot \left\{ \frac{1}{2} (\overline{d_{IN}} + \overline{d_{OUT}}) \right\} \quad (6)$$

$$= \frac{1}{4} (\overline{y_{IN}} \cdot \overline{d_{IN}} + \overline{y_{OUT}} \cdot \overline{d_{OUT}}) + \frac{1}{4} \overline{y_{IN}} \cdot \overline{d_{OUT}} + \frac{1}{4} \overline{y_{OUT}} \cdot \overline{d_{IN}} \quad (7)$$

次に、 $W_{MEAN}$  の定義式 (4) より、意味演算を数式展開し、 $W_{MEAN}$  の解釈を行った。(5) 式は、 $W_{MEAN}$  を用いた際の意味演算出力  $\bar{y}$  と正解単語  $\vec{d}$  の Cos 類似度を表している。これより、(7) 式の第 1 項は、 $W_{IN}$  と  $W_{OUT}$  それぞれの出力結果を平均化していることを意味している。加えて第 2 項、第 3 項は、意味演算出力  $\bar{y}$  と正解単語ベクトルとの共起性を表していることがわかる。さらに、[14] でも報告されている通り、 $W_{IN}$  と  $W_{OUT}$  の内積値を算出

表 3 内積上位単語ペアの単語一致率

Table 3 Word coincidence rate of inner product of word pair.

Top-n	Match rate [%]
1	91.2
2	94.2
3	95.1

表 4 分類対象データセット

Table 4 Dataset for classification.

データセット	学習文書数	テスト文書数	クラス数
livedoor	4420	2947	9
Reuters 21578	5485	2189	8

した際の最上位ペアは同一単語である ( $\overrightarrow{dog_{IN}}$ と $\overrightarrow{cat_{OUT}}$ よりも、 $\overrightarrow{dog_{IN}}$ と $\overrightarrow{dog_{OUT}}$ の内積の値が大きい) 傾向が確認されている。そこで、本実験で用いた分散表現に対して、同様の確認を行った。その結果を表 3 に示す。表 3 は、 $W_{IN}$ と $W_{OUT}$ の内積上位単語ペアにおける同一単語となった単語ペアの割合を示している。Top-n は上位何単語まで許容するかを表すパラメータである。

表 3 より、実際に  $W_{IN}$  と  $W_{OUT}$  の内積値を算出した際の最上位ペアは、同一単語であることが多いことが確認できる。よって、 $W_{IN}$  と  $W_{OUT}$  のどちらか一方で  $\vec{y}$  が正解単語ベクトルに類似していれば、(7) 式の第 2 項、第 3 項により大きく重みがかかることになる。これにより、意味演算出力  $\vec{y}$  に補正がかかり、 $W_{MEAN}$  の意味関係性能が向上した要因の一つとなったと考えられる。またこれは、期待した通り、学習された重みの性質が異なる  $W_{IN}$  と  $W_{OUT}$  を融合することで、より性能の高い単語ベクトルが生成されたことを意味していると考えられる。

## 5. 文書分類への応用

文書分類とは、与えられた文書を予め定められたクラスのいずれかに分類することである。スパムメール分類や Web 記事分類などに広く実用化されている。分散表現の応用例として、分散表現を文書の素性とした文書分類手法が報告されており [17][18]、本章では、提案手法を文書分類に応用し、分類精度への貢献を検証する。

### 5.1 分類対象データセット

実験で使用した分類対象データセットを表 4 に示す。livedoor ニュースコーパスは日本語のテキストデータとなっており、ウェブサイト \*2 からダウンロードして利用できる。また、Reuters 21578 は英語のテキストデータとなっており、[19] の著者のウェブサイト \*3 からダウンロードして利用できる。

\*2 <https://www.rondhuit.com/download.html>

\*3 <http://web.ist.utl.pt/acardoso/datasets/>

### 5.2 Word2Vec 学習コーパス

Word2Vec の学習コーパスには、日本語 Wikipedia と英語 Wikipedia、さらに、各データセットの学習文書を用いた。これにより、コーパスの違いによる精度に対する貢献を確認する。諸条件は 4.3 と同様である。

### 5.3 分類器

分類器には SVM を用いた。SVM は RBF カーネルを用い、ハイパーパラメータはライブラリのデフォルト値である  $C = 1$  と  $\gamma = 1/\text{次元数}$  (300) を用いた。

### 5.4 実験結果

表 5 に、各データセットと、Word2Vec 学習コーパス別に、個々の単語ベクトルを用いて文書分類を行った際の分類精度を示す。本実験では、文書をベクトル化する手法として、[17] で紹介されている文書内単語の平均ベクトルを用いた。表の値は 5 試行の平均であり、分類精度は、テスト文書に対して正しく分類された割合である。

表 5 より、まず次元連結を行う  $W_{CONC}$  は、 $W_{IN}$  (従来手法) よりも精度が向上していることが確認できる。この原因として考えられることは、一般的に単語ベクトルの次元数は、大きい方が分類精度が向上する傾向があることが挙げられる。4 章において  $W_{CONC}$  の単語ベクトルとしての性能は  $W_{IN}$  よりも低いという結果を示したが、単語の意味関係の学習に対する良し悪し以上に、次元数の増大の効果が発揮され、分類精度の向上につながったと考えられる。また、どのデータセットにおいても、 $W_{MEAN}$  の精度が最も高いことが確認できる。一般的に、文書分類の精度と単語ベクトルの性能には相関があるため、4.4 で示した通り、 $W_{MEAN}$  が単語ベクトルとしての性能が高いことで、文書を的確に特徴づけることが可能となり、分類精度の向上につながったと考えられる。

Word2Vec の学習コーパスの違いに注目すると、Wikipedia から学習文書 (ニュース記事) にすることで、全体的に精度が向上していることがわかる。これは、文書特有の単語や表現を学習することが可能となり、分類対象のデータにフィッティングすることで精度の向上を達成したと考えられる。よって、分類用のデータ以外にも、ニュース記事などのコーパスを Word2Vec の学習に利用することで、さらなる分類精度の向上が期待できる。また、 $W_{IN}$  と  $W_{OUT}$  を比較すると、それぞれの精度の大小関係が入れ替わっていることが確認できる。学習文書を Word2Vec の学習に利用した場合、 $W_{OUT}$  の単語ベクトルに共起性、つまり、文書情報が含有されると考えられる。よって、分類文書の文書傾向が一致したことにより、 $W_{IN}$  よりも文書を的確に表すことが可能となり、精度の向上につながったと考えられる。

表 5 各データセットにおける文書分類精度  
Table 5 Document classification accuracy in each dataset.

データセット (Word2Vec)	$W_{IN}$ [%]	$W_{OUT}$ [%]	$W_{CONC}$ [%]	$W_{MEAN}$ [%]
livedoor(Wiki)	87.47	87.41	87.73	<b>87.81</b>
livedoor(学習文書)	91.07	91.20	91.55	<b>91.83</b>
Reuters(Wiki)	93.22	92.94	93.33	<b>93.58</b>
Reuters(学習文書)	93.84	93.88	94.03	<b>95.31</b>

## 6. まとめ

本研究では、分散表現の意味演算性能を高めることを目的とし、Word2Vecの学習過程で副産物的に生成される出力側の重み $W_{OUT}$ に注目し、従来の入力側の重み $W_{IN}$ と併用した単語ベクトルを提案した。意味関係テストを用いて単語ベクトルの性能評価実験を行い、提案する単語ベクトルの性能が、従来分散表現として用いられてきた $W_{IN}$ よりも高いことを示し、その原因について考察を行った。さらに、分散表現の応用例として、文書分類に適用した実験においても、提案する単語ベクトルの有効性が確認できた。

最後に本手法の利点について述べる。出力側の重み $W_{OUT}$ は、 $W_{IN}$ の生成と同時に、常に副産物的に生成されるため、 $W_{MEAN}$ の生成に必要な追加データや時間的なコストがほぼ0である。さらに、ベクトルの次元数を増やすことなく単語ベクトルの性能を向上させられるため、使用時の計算コストも抑えることが可能であると考えられる。

単語のベクトル化手法は、自然言語処理分野の根本を支える技術であり、今後益々の発展が期待される。

## 参考文献

[1] Golson, S.: One-hot state machine design for FPGAs, *3rd PLD Design Conference*, pp. 1–6 (1993).

[2] Sivic, J. and Zisserman, A.: Efficient Visual Search of Videos Cast as Text Retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pp. 591–606 (2009).

[3] Deerwester, S., Dumais, S., Furnas, G., Landauer, T. and Harshman, R.: Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science*, Vol. 41, No. 6, pp. 391–407 (1990).

[4] Blei, D., Ng, A. and Jordan, M.: Latent Dirichlet Allocation, *The Journal of Machine Learning Research*, Vol. 3, pp. 993–1022 (2014).

[5] Maas, A. and Ng, A.: A Probabilistic Model for Semantic Word Vectors, *In NIPS 2010 Workshop on Deep Learning and Unsupervised Feature Learning* (2010).

[6] Mikolov, T., Yih, W. and Zweig, G.: Linguistic Regularities in Continuous Space Word Representations, *NAACL HLT* (2013).

[7] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Efficient Estimation of Word Representations in Vector Space, *In Proceedings of ICLR Workshops Track* (2013).

[8] Xue, B., Fu, C. and Shaobin, Z.: A Study on Sentiment Computing and Classification of Sina Weibo with Word2vec, *Proceedings of the IEEE International*

*Congress on the Big Data (BigData Congress)*, IEEE, pp. 358–363 (2014).

[9] Ma, L. and Zhang, Y.: Using Word2Vec to process big text data, *In: 2015 IEEE International Conference on Big Data (Big Data)*, pp. 2895–2897 (2015).

[10] Ju, R., Zhou, P., Li, C. and Liu, L.: An Efficient Method for Document Categorization Based on Word2vec and Latent Semantic Analysis, *proceedings of IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable Autonomic and Secure Computing; Pervasive Intelligence and Computing. IEEE*, pp. 2276–2283 (2015).

[11] Sien, S.: Adapting word2vec to Named Entity Recognition, *proceedings of Nordic Conference of Computational Linguistics* (2015).

[12] Harris, Z.: Distributional Structure, *Word, IO*, pp. 140–162 (1954).

[13] Rong, X.: Word2vec Parameter Learning Explained, *arXiv preprint arXiv:1411.2738* (2014).

[14] Mitra, B., Nalisnick, E., Craswell, N. and Caruana, R.: A Dual Embedding Space Model for Document Ranking, *CoRR abs/1602.01137* (2016).

[15] Press, O. and Wolf, L.: Using the Output Embedding to Improve Language Models, *In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Volume 2, Short Papers, Valencia, Spain*, pp. 157–163 (2017).

[16] Yin, W. and Schutze, H.: Learning Word Meta-embeddings, *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pp. 1351–1360 (2016).

[17] Liu, R., Wang, D. and Xing, C.: Document Classification Based on Word Vectors, *In ISCSLP'14* (2014).

[18] Xing, C., Wang, D. and Zhang, X.: Document Classification with Distributions of Word Vectors, *In Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pp. 1–5 (2014).

[19] Cardoso-Cachopo, A.: Improving Methods for Single-label Text Categorization, *PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa* (2007).