2015 年度 修士論文

教師なしデータを利用した単語の分散表現と事前学習を用いた Web ニュースデータのカテゴリ分類

法政大学大学院　理工学研究科

システム工学専攻　経営系　修士課程

14R6204　加藤（カトウ）　諒磨（リョウマ）

指導教員

五島　洋行教授

Abstract

In this research, we examine that text classification in a small amount of training data.

In a field of text classification, supervised learning like the Naive Bayes method and support vector machine is often used. Supervised learning can output good accuracy if we have a large amount of training data. However, if we have only a small amount of training data, there is a problem in that supervised learning often output bad accuracy. Moreover, it is costly that we make a large amount of training data. Semi-supervised learning is proposed to reduce training data. However, there is a problem that the classification accuracy decreases if the amount of training data increased. On the other hand, we can handle unsupervised data in pre-training of deep learning. Therefore, we propose reducing necessary amount of training data to output good accuracy by using unsupervised data in pre training at deep learning.

Text data needs to be quantified to handle in neural network. Bag-of-words is a method that is often used to quantify document. However, the dimension of sentence quantified by using bag-of-words equals the amount of vocabulary in all text data, moreover it become too large if we handle large text data. Computational complexity explosive increases in neural network if dimension of input become large. Therefore, sentence quantified by using bag-of-words is not suitable to handle as input in neural network. On the other hand, Word2Vec which can express a word in a low dimension was developed in recent years. Moreover, Word2Vec is not supervised learning but unsupervised learning. Thus, we propose quantifying document by using Word2Vec instead of bag-of-words.

The result of examining accuracy of proposed method, we found that proposed method can output good accuracy in a small amount of training data.

1.はじめに

1.1. 研究背景

近年，インターネット環境の改善やスマートフォンの普及により，インターネットの利用者が増したことで，Web 上のデータ量が膨大に増している[1]．特にテキストデータは，近年ソーシャルネットワークサービスの発展や従来の様々な文書の電子化により図 1 のようにデータ量が増し，今後も更にデータ量が増していくと考えられており，その活用が重要視されている．また，データの量が増したことにより，人力でそのような量のテキストデータを処理することは困難になっている．そこで，コンピュータを用いてテキスト分析を行うテキストマイニングの技術が研究されている．テキストマイニングの目的は，未知の特徴の発見や，人間がテキストデータを扱う際の補助等があげられ，本研究では，その中で人間がテキストデータを扱う際の補助に注目して研究を行う．

テキストデータを扱う際の補助とは具体的には，長い文章を要約し，短時間で文章の内容を理解するための補助や，テキストデータに何らかの属性を付与し，興味のある文章や目的の文章を探す際の補助のことを指している．

また，自動文書要約の技術は，まだ実用的な段階ではないとされている．その理由は，文書要約ではコンピュータでテキストの重要な箇所を探す必要があり，それにはテキストがどのような内容であるかの分類が前提で必要であり，まだ前提になっている分類処理を十分に行えていないため，要約の精度や内容も不十分になってしまうと考えられる．そのため分類技術の発展が，後続する他のテキストマイニングの技術に必須であると考え，テキスト分類技術の研究を行う．

自動的にテキスト分類を行う機械学習の研究は，様々なテキストデータや様々な機械学習の手法を対象に盛んに行われてきた．手法としては，ナイーブベイズを用いた分類[2]や SVM を用いた分類[3]等の一般的な機械学習アルゴリズムの手法は一通り試され，成果をあげている．しかし，これらの手法は機械学習の手法のうち教師あり学習のみを用いており，教師あり学習では，正解の情報が付与された教師ありデータのみしか扱うことができない．しかし，実際に扱う Web 上のテキストデータの多く

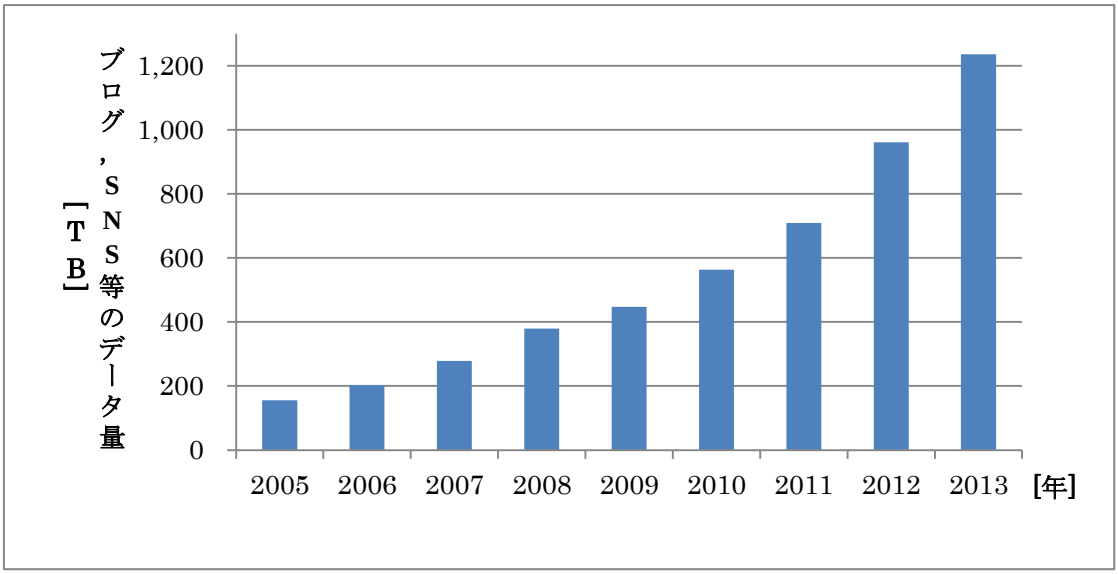

図 1 Blog, SNS 等のテキストデータ量の推移[1]

は，正解の情報が付与されていない教師なしデータであり，上記の手法ではそれらの多くのデータを扱うことができない．

一方で，正解ラベルが存在しない教師なしデータを扱う分類機械学習アルゴリズムとして，クラスタリングがあるが，クラスタリングでは，機械が自動的にカテゴリを定めて分類をするため，人の定めたカテゴリに分類するという本研究の目的とは異なっているため扱わない．

また，テキストデータを機械学習で扱う際の課題として，文章を何らかの数値に変換するという課題がある．機械学習でテキストデータを扱うには，単語や文字列ではなく何らかの数値に変換する必要があり，その変換方法も様々に研究されている．また，多くの機械学習のアルゴリズムでは，ネットワークや分類器の大きさや形を事前に定める必要がある．テキストデータは単語の数や文字の数，文章の長さがそれぞれのテキストデータで異なっているため，そのような可変長なデータを固定長に変換する課題もある．そのような可変長の文章を固定長の数値ベクトルに変換する手法として，最も用いられている手法が bag-of-words である．bag-of-words は図 2 のように，文章を単語の集合として捉える手法である．

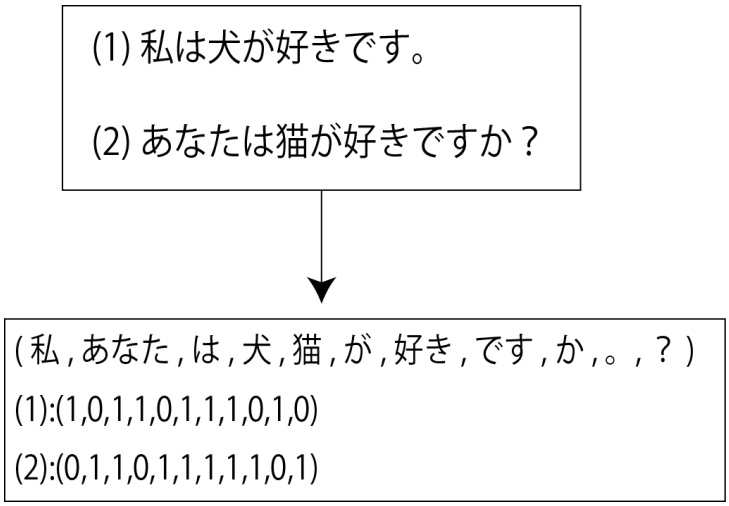

図 2　bag-of-words の一例

bag-of-words では，ある単語が文章に含まれていたら，ベクトル中のその単語に割り当てられた要素の値を 1 にし，含まれていなければ 0 をとるバイナリ表現を利用している．

図 2 では，二つの文章のみから作られるコーパスの例で説明している．変換後の 1 次元目は「私」としたとき，(1)の文章には出現するため，1 次元目を 1 とし，(2)の文章には「私」は出現しないため，1 次元目は 0 とする．これをコーパス中の全ての語彙数分処理を行い，文章をベクトル化する．この表現を用いることで，文章を一定の固定長に変換することが可能であるが，学習するコーパスに含まれている語彙の数だけベクトルの次元数が大きくなってしまう．これにより，大規模なコーパスを学習する際には入力次元が数万次元になり，ニューラルネットワーク等のネットワークを用いた機械学習で学習を行う際にネットワークが指数関数的に広がり，計算時間が膨大になってしまう問題がある bag-of-words の次元数を圧縮するために，特異値分解による次元圧縮を行い，それを入力とした分類[4]も行われているが，1,000 次元まで削減してしまうと精度が悪くなるという結果が得られている．

図 3　三つのニュースサイトのカテゴリ一覧とその関係

また，今回分類を行うテキストデータとして，Web ニュースサイト上のニュースデータを選択した．現在，Web ニュースサイトは数多く存在し，配信しているニュースの数も膨大である．また，ニュースデータの配信件数は Yahoo!ニュースのみで，一日数千件あり，全てのサイトの全てのニュースを一人で見ることは，現実的ではないと考えられる．そのため，ユーザは閲覧する情報を取捨選択する必要があり，それを支援する際に，ニュースが属しているカテゴリがよく用いられている．本研究では，そのカテゴリを自動的に分類する．

しかし，Web ニュースサイトのカテゴリは図 3 のように，そのニュースが配信されているサイトごとに，定めているカテゴリの種類が異なっており，一部のニュースサイトにのみ存在し，他の Web ニュースサイトには存在しないカテゴリが存在してしまうことがある．そのため，ニュースサイトが提示しているカテゴリは，ニュースを探すための補助として役に立たない場合がある．そこで，Web ニュースを何らかの統一されたカテゴリに自動分類することが可能になれば，サイト間のカテゴリの差異をなくすことが可能になり，全てのカテゴリを目的のニュースを探す補助とすることが可能になると考えられる．そのため，Web ニュースデータには，自動文書分類の技術が必要であり，文書分類の技術を活かすには，Web ニュースデータは適したデータで

あると考えられる．

1.2. 研究目的

本研究では，Web ニュースデータを定められたカテゴリに自動で精度よく分類することを目的とする．また，そのために研究背景で述べた従来の手法の二つの問題を解決することを目指す．

第一の問題である教師なしデータを扱うことができない問題は，近年成果を上げている深層学習を用いることで解決を目指す．深層学習には教師あり学習の前に，事前学習というプロセスがあり，そこでは教師なしデータを扱うことが可能であるため，それを利用し，教師なしデータを利用した学習を行う．

第二の問題である入力の次元数が膨大になってしまう問題は，bag-of-words を用いず，2013 年に Google で開発された Word2Vec という単語を低次元のベクトルに変換するツールを用いて，文章の固定長ベクトル化を行い，次元数を削減することを目指す．また，Word2Vec は単語を，意味を持つベクトルに変換できるという結果が出ているため，今までの次元圧縮手法と比較して，良い精度になることが期待できる．

また，本研究を活かすことができる分野として，新しく作られるデータを利用するサービスがあげられる．新しく作られるサービスでは，顧客の情報が少なく，コンピュータの環境が潤沢ではないと考えられる．そのような環境では，数少ない顧客の教師データを頼りに，商品のレコメンドサービス等のサービスを，なるべく少ない計算時間で実現させる必要がある．本研究は，事前学習により，顧客の教師データの必要性を減らし，Word2Vec を利用することで bag-of-words 等のベクトル表現を使うよりも，計算回数を短縮することが可能であるため，そのようなサービスの初期段階で有用性があると考えられる．

2.関連知識・関連研究

この章では，関連知識と関連研究の説明を行う．関連知識の節では，提案手法で利用するツールや技術の知識を紹介し，どのように本研究で利用するのかを説明する．また，関連研究の節では，本研究の目的である文書分類の研究と，特徴である Word2Vec を用いた研究を紹介し，従来の研究と本研究との違いを説明する．

2.1. 関連知識

2.1.1.ニューラルネットワーク

ニューラルネットワークとは，機械学習の手法の一つである[8]．また，次項で説明する深層学習を除くニューラルネットワークは，教師あり学習で学習を行う．本研究では，教師なしデータを利用するために，深層学習を利用するため，その基礎であるニューラルネットワークの説明と，本研究で具体的にどのように利用するかを説明する．

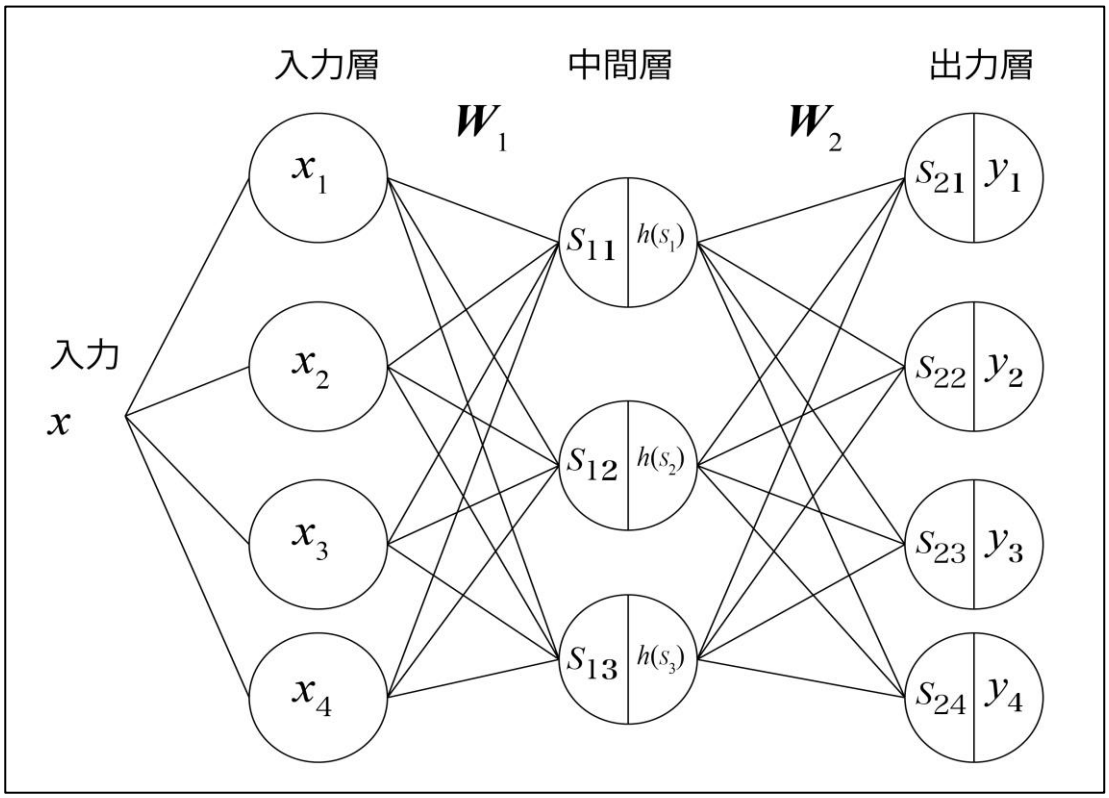

図 4　階層型ニューラルネットワークの例

図 4 は階層型ニューラルネットワークの一例である．ニューラルネットワークの中でも，本研究で用いた階層型ニューラルネットワークは図 4 のように，ノードとノード同士を繋ぐリンクによって構成される全連結型ネットワークの学習器である．また，深層学習ではない一般的なニューラルネットワークは，入力層と中間層と出力層を各一つずつ持つ構成であることが多い．

ニューラルネットワークの仕組みは，簡潔に表すと入力層から値を与え，その値を中間層で変換し，出力層に伝達し答えを出力させる仕組みである．学習は，中間層で値を変換する際に利用するノードの重みを最適化することで行われる．また，層の数やノードやリンクの数などのニューラルネットワークの形は事前に決定する必要がある．

次に，入力層，中間層，出力層からなるニューラルネットワークの学習の手順を詳細に説明する．

はじめに，入力層のノード数に適したデータを入力層に入力する．そして，入力層の各ノードから中間層の各ノードへのリンクを通じて，中間層のノードへ受け渡す．このとき，それぞれの値が通るリンクは，それぞれ重みを持っており，入力値とその重みとの積の値が中間層へ伝えられる．また中間層のノードでは，まず値の結合が行われる．入力層の i 番目のノードの値を x_{ij} とし，その i 番目のノードから j 番目の中間層のノードへつながっているリンクの重みを $wi_{ij,}$ 中間層の j 番目のノードの閾値を $b_{j,}$ 入力層のノード数を n とする．そうすると一つ目の中間層の j 番目のノードに渡る値 s_{1j} は式(1)のように表すことができる．

$$s_{1j} = \sum_{i=0}^{n} w_{ij}x_i - b_j \tag{1}$$

次に，s_{1j} を活性化関数 h を用いて，非線形変換を行う．このときの活性化関数 h は非線形の関数である必要がある．これは，活性化関数を線形にすると単純パーセプトロンと表現力が同じになってしまい，多層パーセプトロンであるニューラルネットワークにする意味がなくなってしまうためである．また，今回は活性化関数 $h(s_{1j})$ を tanh 関数とする．tanh 関数は式(2)のような関数で，微分を行うと式(3)のようになる．

$$\tanh(a) = \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}} \tag{2}$$

$$\tanh'(a) = 1 - (\tanh(a))^{2} \tag{3}$$

誤差を修正する際に活性化関数の微分が必要になるため，微分後が単純な式になる必要があり，そのような関数の中でも実績として tanh 関数は良い精度を出すことが知られているため，今回は tanh 関数を利用した．

この $h(s_{1j})$が次の層への入力となり，中間層のノード j から出力層のノード k へのリンクの重みを w_{jk} とする．入力層から中間層の受け渡しと同様にリンクの重みの値とリンクを掛けあわせ，ノードに入力される値はその総和である．出力層の活性化関数は中間層とは異なる関数を使う必要があり，それは目的とするタスクごとに異なっている．今回のタスクは，ニュースデータのカテゴリ分類という多クラス分類であるため，多クラス分類の際に利用する活性化関数であるソフトマックス関数のみ紹介を行う．式(1)の場合と同様に出力層のノード k への入力 s_{2k}，出力するクラス数を K とすると出力層のノード k の出力 y_k は式(4)のようになる．

$$\mathrm{y_k} = \frac{e^{s_{2k}}}{\sum_{k=1}^{K} e^{s_{2k}}} \tag{4}$$

この y_k の値が最も大きいノードをその入力データが属しているクラスと判断する．また $\boldsymbol{y}$ の値の和は 1 となるため，y_k は入力データがクラス k に属している確率とみなすことが出来る．多値分類をタスクとしたニューラルネットワークでは，このように入力データの分類を行う．

また，結果が正解と異なっていた場合は学習を行う必要がある．学習には，勾配降下法を利用し，誤差関数を最小化する．正解ラベル $\boldsymbol{d}$ を正解のラベルの次元のみ 1 とする one-hot 表現を用いて表現する．5 個のクラスが有り，2 つ目のクラスが正解ラベルの場合，$\boldsymbol{d}$ は式(5)のようになる．

$$\boldsymbol{d} = [0\ 0\ 1\ 0\ 0]^{\mathrm{T}} \tag{5}$$

多値分類のとき，最小化する誤差関数 $E(\boldsymbol{w})$は一件の正解ラベル $\boldsymbol{d}$ と出力 y_k,出力クラス数 K を用いると式(6)となる．

$$E(\boldsymbol{w}) = -\sum_{k=1}^{K} d_k \log y_k \tag{6}$$

この誤差関数を，勾配降下法を用いて最小化を行い，ネットワークの各リンクの重み $\boldsymbol{w}$ を最適な値に近づける．このとき，誤差関数の勾配$\nabla E(\boldsymbol{w})$は，ネットワークのリンク数 M を用いると, 式(7)のように表される. また, 式(7)中の∂は偏微分を表している．

$$\nabla E = \frac{\partial E}{\partial \boldsymbol{w}} = \frac{\partial E}{\partial \boldsymbol{w}} = \left[\frac{\partial E}{\partial w_1} \quad \frac{\partial E}{\partial w_1} \quad \cdots \frac{\partial E}{\partial w_M}\right]^{\mathrm{T}} \tag{7}$$

またこの勾配$\nabla E(\boldsymbol{w})$を用いて，t+1 回目の重み $\boldsymbol{w}^{(t+1)}$を式(8)とする．

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} - \varepsilon \nabla E \tag{8}$$

式(8)中の ε は，学習率と呼ばれ大きいほど早く学習が進むが，学習が進まなくなる場合があり，小さすぎると計算回数が増加してしまう．学習率は目的のタスクやネットワークの形状によって異なるため，各自で調整する必要がある．

また，今回は勾配降下法の中でも確率的勾配降下法を利用する．通常の勾配降下法ではサンプルのデータ全ての勾配を同時に計算し，その勾配を利用して，ネットワークのリンクの重みを修正するが，確率的勾配降下法では，任意の数のサンプルを全体からランダムで選択し，その勾配で重みを変化させる．確率的勾配降下法を行うことにより，毎回の勾配が変化することによって，良くない局所解に入るリスクが軽減できる．また，その選ぶサンプルの個数もタスクやネットワーク依存であるため，各自で調整する必要がある．

本研究では, 入力を Web ニュースデータの本文とし, 出力をそのニュースデータが属するカテゴリと設定した多値分類のタスクを階層型ニューラルネットワークで解く．

2.1.2.深層学習

深層学習とは，上記で説明した階層型ニューラルネットワークの中間層を増やしたニューラルネットワークの総称である．通常のニューラルネットワークでは，中間層を単純に増やすと入力層に近いリンクの重みほど，誤差を修正するための勾配が消失してしまい，学習が上手くいかなくなるという問題がある．しかし，近年事前学習を行うことで，この問題を解決できる場合があることがわかった．事前学習に関する詳しい説明は次項で行う．

事前学習により，層を重ねることができるようになり，深層学習では，深い学習が可能になったと言われている．深い学習とは，層を増やすことによりネットワークの表現力が上がったことで，従来学習が行えなかった複雑な問題を扱えるようになったことや，特徴を人力で入力する必要がなく，自動で学習することができることを指している．

また，事前学習を行うと，学習が上手くいく理由は数学的には解明されていないが，一般的には，事前学習を行うことで，ネットワークの初期値が従来ランダムに与えられていた場合よりも改善されることで，各リンクの変更量が小さく，入力層へ近い層でも十分学習が行えるからであるとされている．また，事前学習のあとに，通常のニューラルネットワークと同様に教師ありデータを用いて学習を行う．また，そのことをファインチューニングという．このように深層学習は，事前学習とファインチューニングの二段階で学習が行われる．

今回，提案手法で深層学習を利用した理由は，深い学習を行うためではなく，大量の教師なしデータを用いて事前学習を行い，良い初期値を手に入れることで，少量の教師ありデータで効率のよい学習が出来ると考えるため，深層学習を利用した．

2.1.3.事前学習

事前学習とは，教師ありデータを用いたファインチューニングの前に教師なしデータを利用して，層の初期値を定める学習法である．事前学習を行わないニューラルネットワークでは，層の初期値を乱数で定めることが一般的である．しかし，中間層が多層になるにつれ，深層学習の項で説明した通り，勾配が消失してしまう問題がおこる．また，事前学習で良い初期値を手に入れることで，この勾配消失問題が解決すると考えられている．事前学習には，主に制約付きボルツマンマシンと Autoencoder(自己符号化器)の二種類がある．本研究では，ネットワークの事前学習に Autoencoder

を使用したため，ここでは Autoencoder の説明を行う．

事前学習には, 主に制約付きボルツマンマシンと Autoencoder(自己符号化器)の二種類がある．本研究では，ネットワークの事前学習に Autoencoder を使用したため，ここでは Autoencoder の説明を行う．

簡潔に説明すると，Autoencoder とは各層ごとで入力を再現する値を出力できるように，各層へのリンクの重みを調節する手法である[9].

Autoenoder は学習を行う層への入力である encode とその入力を再現した出力である decode の二つの段階に分けられる．深層学習の事前学習として利用する場合は，encode の部分のリンクの重みのみを取り入れることになる．入力を $\boldsymbol{x}$，中間層への重みを $\boldsymbol{W}$，活性化関数を $f(x)$，バイアスを $\boldsymbol{b}$ とすると，encode は通常のニューラルネットワークの学習時と同様に中間層からの入力 $\boldsymbol{y}$ は次のように表される．

$$\boldsymbol{y} = f(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) \tag{9}$$

また, decode は式(9)から入力 $\boldsymbol{x}$ を再現する式である. enocde の $\boldsymbol{y}$, decode 時の重み $\boldsymbol{W'}$, バイアス $\boldsymbol{b'}$, 活性化関数 $f'(x)$ を用いて, 入力 $\boldsymbol{x}$ を再現した値 $\boldsymbol{x'}$ は次のように表される．

$$\boldsymbol{x}' = f'(\boldsymbol{W}'\boldsymbol{y} + \boldsymbol{b}) \tag{10}$$

Autoenocder は encode 時の式(9)中の入力 $\boldsymbol{x}$ と decode 時の式(10)中の $\boldsymbol{x'}$ との誤差を最小化するように学習を行う．本研究での学習方法は，通常のニューラルネットワークでの学習方法と同様に確率的勾配降下法で最適化を行う．

入力データより中間層のノード数が少ない場合は，次元圧縮が行われ，主成分分析のような働きをする．本研究では，中間層のノード数を変更するため，ノード数が入力層より少ない場合も多い場合も両方扱う．

また，本実験で取り扱う Autoencoder は denoising-Autoencoder と言われる特殊な Autoencoder である．通常の Autoencoder は先述の通り，入力を与え，その入力をそのまま再現するような出力を encode と decode の二段階で学習する．一方，denoising-Autoencoder では本来の入力に一定の確率でノイズを与え，そのノイズを与えた値を入力とする．そして，その入力を元に Autoencoder と同様の学習を行うが，

目的とする出力は，受け取った入力ではなく，ノイズが発生する前の入力であるという点が通常の Autoencoder とは異なっている．

2.1.4.Word2Vec

Word2Vec とは, Google の Mikolov が提案した単語の分散表現を学習し，単語をベクトル化する手法である．テキストデータをコンピュータで扱うためには，単語を単語そのものではなく，何らかの数値に置き換える必要がある．従来は，bag-of-words や bag-of-words によるベクトル空間を LSI(Latent Semantic Indexing)で次元圧縮した単語ベクトル等が用いられてきた．

しかし, bag-of-words は第一章でも紹介した通り，ベクトルの次元数が，全コーパス中の語彙数と等しくなってしまう欠点があり，本研究で用いたコーパスでは 230,000 語以上あるため，次元数も 230,000 次元にものぼる．

LSI では，特異値分解により次元圧縮を行っている．どの程度圧縮を行うかは任意であるが，次元圧縮を行うほど，精度が悪化してしまうと言われている．実際に bag-of-words によるベクトル空間で 60,000 次元のコーパスを 1,000 次元に圧縮すると，二値分類の精度が 95%から 55%まで下がってしまったという例[4]もある．

しかし, Word2Vec では，単語を 200 次元程度のベクトル空間に変換することができるとされており，このような密なベクトル表現のことを分散表現という．また，Word2Vec で学習したベクトル表現は，意味を学習していると言われており，

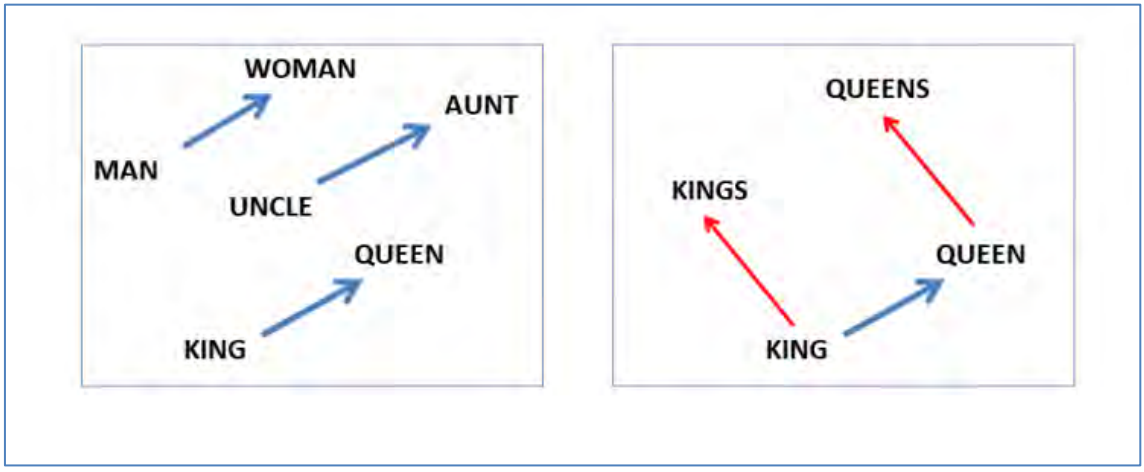

図 5　Word2Vec で学習した単語のベクトルの足し引き　出展 Mikolov(2013)[6]より

Word2Vec を紹介した Mikolov の論文[6]では，図 5 のように WOMAN のベクトル表現から MAN のベクトル表現を引いたベクトルを KING に足すと，最も近くに存在する単語が QUEEN であるという意味の足し引きが可能にみえる結果が得られている．このように意味の足し引きができるため，本研究では，Word2Vec の単語表現を用いて，文章のベクトルを構築し，その値で分類を行う．

Word2Vec は分布仮説に基づいて考えられたツールである．分布仮説とは，単語の意味は単語の周辺の言葉によって与えられるという仮説である．例を出すと，「○○は赤くて美味しい」という文章があったとき，○○に入る選択肢として，りんご，みかん，ぶどうがあった場合，多くの人間がりんごを選ぶ．それは「赤い」や「美味しい」がりんごを表す言葉であるためである．このように単語の周辺の言葉がわかれば，その単語の意味もわかってくるという仮説が分布仮説である．Word2Vec は上述の分布仮説に基づき，以下の仕組みで成り立っている．

Word2Vec には二つの工程がある．第一の工程として，CBOW(Continuous Bag-of-Words)モデルの学習を行う．CBOW とは，表現したい単語の周囲の単語を入力として与え，表現したい単語を出力させるニューラルネットワークを利用した学習モデルである．考慮する周囲の単語数は任意で決めることが出来る．これにより，単語の語彙数×周囲の単語数の重み行列を学習させることができる．また，計算を軽くするために，語順を考慮せず，活性化関数を恒等写像とするという工夫が行われていることが特徴である．

第二の工程として，Skip-gram モデルの学習を行う．Skip-gram モデルとは，CBOW の逆の学習を行う学習モデルである．CBOW では周辺単語から，目的の単語を予測したが，Skip-gram では一つの単語からその周辺の単語を予測するという学習を行う．

この二つの工程を同じ中間層で結び，入力として与えた周辺単語から，単語を予測し，その単語から周辺単語を予測するという学習を行うと，その中間層の重み行列が単語の分散表現を獲得した行列となる．この中間層の次元数を 200 とすると，各単語を 200 次元のベクトルで表記することが可能になる．

2.2. 関連研究

この節では，本研究の関連研究を紹介し，本研究との違いを述べる．

2.2.1.文書分類に関する研究

文書分類の研究は古くから行われており，第一章で紹介したように，ナイーブベイズを利用した研究[2]や SVM を利用した研究[3]などの機械学習の手法は一通り試されている．ナイーブベイズは，文書分類以外の分野では，それほど精度が良くない手法とされているが，[2]では，十分な教師データがあれば，85%の精度も出すことができたという報告があり，文書分類では有用であることが知られている．しかし，このような教師あり学習は教師ありデータが十分にあるという条件以外では，優れた精度を出すことは困難である．ナイーブベイズや SVM を両方実験している研究[7]でも，教師データ数が各クラス 30 件の多値分類の場合は分類精度 55.5%であったが，各クラスの教師データ数が 800 件のときは分類精度 85.4%という結果を出している．このように，精度の高い分類結果を出すためには，多くの教師ありデータが必要であると考えられる．

そこで，少数の教師ありデータで分類精度をあげるための研究[10]も行われている．この研究では，主に半教師あり学習を利用することで，学習に必要な教師ありデータの量を減らすことを提案している．半教師あり学習とは，教師ありデータを利用し，教師なしデータに情報を付与する学習方法である．具体的には，A というカテゴリの属性が付いている教師ありデータに，ベクトル上で近い教師なしデータをカテゴリ A であると確率的に判断する手法である．この半教師あり学習を利用すると，通常の学習方法よりも精度があがるという結果が得られている．しかし，これは教師ありデータが少ない場合であり，教師ありデータが増えた場合，半教師あり学習を利用した方が精度を落とす場合がある．

そこで，本研究では教師なしデータを利用する部分を半教師あり学習で行うアプローチではなく，事前学習にするアプローチを行うことで教師なしデータの利用を実現している．事前学習を行うことで精度が下がった例はないため，事前学習によるアプローチでは，教師データが増えた場合も精度を落とすことなく学習を行うことができると考えられる．

2.2.2.Word2Vec に関する研究

Word2Vec は 2.1.4 項で紹介したように 2013 年に公開された新しいツールであるため，その用途は現在も様々な研究が行われている．Word2Vec の特徴として，低次元の密なベクトルで精度の高い単語の表現が行えることや，単語のベクトルの関係が意

味の関係を表しているということがあげられる．

日本語が対象の研究では，単語の意味を取り扱う研究が多い[5][11]．[5]では，Word2Vec を用いて語義の曖昧性の解消を目指している．語義の曖昧性というのは，同じ単語でも場面によって意味が異なることをいう．例えば cool という単語は文脈によって，涼しいという意味とカッコ良いという意味の完全に別の意味を持っている．[5]では，このような単語を文脈から判断する際に，Word2Vec のベクトル表現を利用している．結果として，Word2Vec を利用した分散表現を利用した手法が最も精度が良いことがわかった．[11]では，レシピのテキストデータをコーパスとして Word2Vec で学習させることにより，何らかの食材の代わりになる食材を Word2Vec の単語ベクトルを利用して，近い単語から見つけることを目的としている．

Word2Vec は，ベクトルで意味の学習を行うことが可能であるとされているため，そのような単語の意味を扱ったような研究が多く行われていると考えられる．しかし，新しいツールであるため，Word2Vec を利用して作成した単語ベクトルから，文章ベクトルを作成した日本語の研究は少ないのが現状である．

海外の論文では，中国語を対象にし，中国で利用されている SNS の Weibo のユーザの投稿を対象に，単語の感情分析を行っている[13]． この研究では，Word2Vec を利用し作成した単語ベクトルをクラスタリングし，感情ごとで各単語が近い距離に配置されているかどうかを調査し，その後その単語ベクトルから文章ベクトルを作成し，文章の感情の値を算出している．

また，この Word2Vec と深層学習を利用した研究として[12]がある．最も類似している[12]と本研究の違いは，分類器に事前学習を含めた深層学習を利用しているという点である．[12]では，深層学習を単純な多層ニューラルネットワークとして扱っており，事前学習や教師なしデータやの利用については触れられていない．本研究では，ネットワークの表現力をあげるために深層学習を行っているのではなく，事前学習で教師なしデータを扱うことで教師ありデータの数を減らすことが目的とし，深層学習を利用しているため，その部分が最も異なっていると考えられる．また，[12]では，二値分類のみしか扱っておらず，多値分類を行うとどのような精度になるかが不明であるため，その点も異なっている．

また，深層学習では，層の数やノードの数といったパラメータが精度を大きく左右するにも関わらず，Word2Vec を利用した入力を使ったニューラルネットワークにお

ける学習でそれらのパラメータの参考となる値が他の論文では示されていないため，その値を実験によって定める．

3.分散表現と事前学習を利用した分類手法

この章では，本研究で提案する分類手法について説明を行う．

3.1. 分類手法の概要

本研究で提案する分類手法の特徴は，入力データを低次元とし，計算時間を短縮することと，入力データに教師なしデータを利用可能にすることであり，その目的は，分類精度の向上に必要な教師ありデータを減らすことである．前者を Word2Vec により実現させ，後者を深層学習の事前学習により実現させる．手法の概要図を図 6 に示す．

はじめに，所持している全てのニュースデータを Word2Vec へ入力し，ニュースデータ中に出現した各単語を 200 次元のベクトル化した分散表現を出力させる．そして，作成した各単語の分散表現のベクトルを元に，教師なし Web ニュースデータとその

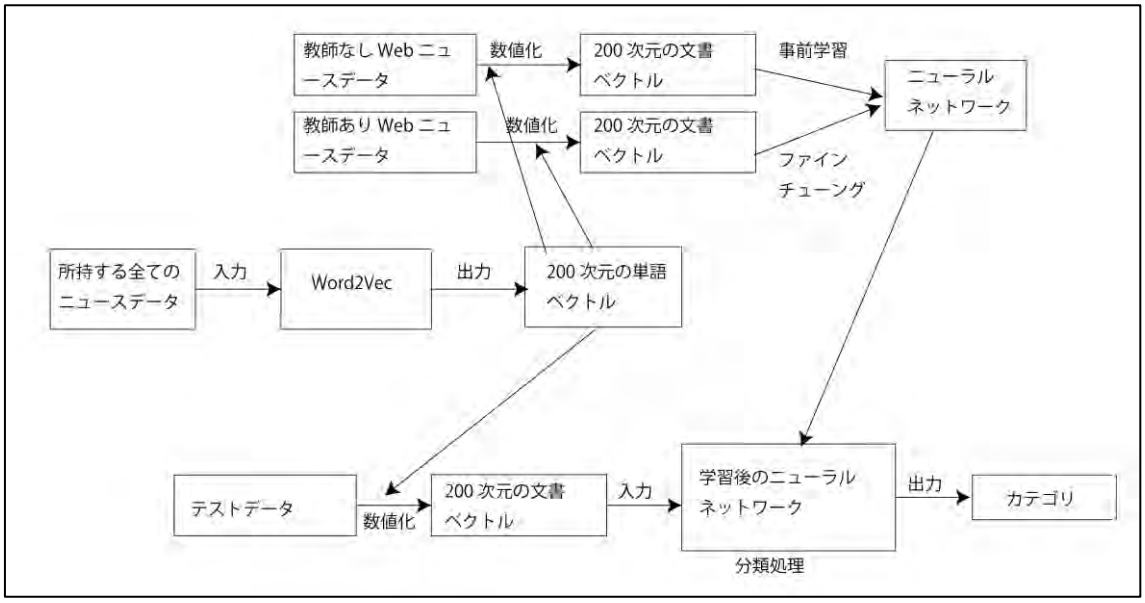

図 6 提案する分類手法の概要

ニュースデータがどのカテゴリに属しているのかのラベルを持った教師あり Web ニュースデータを 200 次元のベクトルに変換する．

次に，変換した教師なし Web ニュースデータのベクトルをニューラルネットワークの事前学習として入力する．これを AutoEncoder を利用して学習させることで，ニューラルネットワークの事前学習前の初期値を修正する．次に，変換した教師あり Web ニュースデータのベクトルをニューラルネットワークのファインチューニングの入力として与える．これにより，ニューラルネットワークの学習が完了する．

最後に，教師ありデータと同様にカテゴリの情報を付与したテストデータをニューラルネットワークの入力として与え，カテゴリを出力させ，正解ラベルと一致するかを調べ，評価を行う．

3.2. Word2Vec を用いた入力データの作成

この節では，本研究で利用する入力データの作成方法についての説明を行う．

本研究で扱うデータは，それぞれ長さや単語の数等が異なっている文書データである．コンピュータで文書を扱うには，文書や単語そのものをそのまま扱うことはできず，何らかの数値に置き換える必要がある．また，2.1.1 のニューラルネットワークの項で説明した通り，ニューラルネットワークの入力層のノード数は事前に定める必要があるため，文書の数値化を行う際には，何次元かの定められた次元数に変換する必要がある．今回提案する手法では，ニューラルネットワークを全連結の階層型のニューラルネットワークとしており，このニューラルネットワークの形では，入力が n 次元で，中間層のノード数が m の場合，修正するリンクの数が入力層から中間層へのリンクの数だけで n^m となり，一回の修正にかかる計算量は入力の大きさによって大きく変化してしまう．そのため，高速で計算を行うためには入力次元を低くする必要がある．

そこで本研究では，従来の次元数が膨大になる bag-of-words などの手法ではなく，Word2Vec を利用した文書のベクトル化を提案する．Word2Vec とは関連知識の項でも述べたように文書を入力とし，入力した文章中に出現した単語のベクトル表現を出力するツールである．

Word2Vec を利用することで，単語の分散表現を獲得することが可能になり，低次元のベクトルに圧縮できるためである．分散表現とは，bag-of-words のような一つの

次元のみが値を持ち，その他の次元の値が 0 になる疎な表現方法ではなく，全ての次元に何らかの値が与えられている密なベクトル表現である．

Word2Vec は 2015 年 12 月現在，Web 上でソースコードが公開されており，誰でも利用することが可能である．新しいツールであるため，その利用手順を記載する．Linux 系の OS や MacOSX 系の OS における利用手順は以下の通りである．

●ターミナルや端末上でファイルをダウンロードしたいディレクトリまで移動し，「svn checkout http://word2vec.googlecode.com/svn/trunk/」と入力する．

●現在いるディレクトリに trunk というフォルダが作成されている．そのフォルダに移動し，「make」コマンドを実行する．MacOSX 系の場合は「word-analogy.c」，「compute-accuracy.c」,「distance.c」の#include <malloc.h>を#include<stdlib.h>に変更する必要がある．

●Word2Vec で利用したいテキストデータを同フォルダに移動させる．

●ターミナルや端末上で「time ./word2vec -train 入力するテキストデータ名 -output 出力名 -cbow 1 -size 次元数 -window 考慮する周辺単語の数 -negative 25 -hs 0 -sample 1e-4 -threads 20 -binary 出力形式 -iter 15」と入力し，学習を行う．

●出力されるファイルに，単語とベクトルのデータが保存されている．

今回は，-size オプションで指定する Word2Vec で変換する際の次元数を 200 次元とし, -window オプションで指定する周辺単語の数を 5 とした. 200 次元とした根拠は，公式の論文での実験が 200 次元で行われていたためである．周辺単語の数は，実際に単語の表現を作成した際に，この数が単語の関係が最も適切に表れていたため，今回はこの数を採用した．

また，Word2Vec は本来英語を対象にしているため，単語同士が半角スペースで句切られていることが前提になっている．本研究で扱う Web ニュースデータは日本語であるため，事前に MeCab 等の形態素解析ツールを利用して，文章を単語ごとにスペースで区切る必要がある．本研究では，MeCab を利用し，事前に全てのニュースデータの単語を基本形で分割し，半角スペースで区切ったテキストデータを Word2Vec の入力として与えた．

本研究で扱うデータはニュースデータであり, 近年新しく出現した単語も考慮する必要があるが, MeCab の元々備わっている辞書にはそのような単語は登録されていない．そこで，本研究ではユーザ辞書として，Wikipedia のタイトルを名詞として辞書

登録することで，近年出現した単語を考慮した．このことにより，芸能カテゴリやスポーツカテゴリの固有名詞を正しい形で分かち書きすることが可能になる．

ここまでで，Word2Vec により，各単語を同一の 200 次元のベクトルで分散表現することが可能になった．ここで，今までの手順と通して実際に単語の表現を学習できているかを検証する．そこで，指定の単語から距離が一番近い単語を調べる．図 7 では，「衆議院」からコサイン類似度が近い単語を上位から順に表示した結果を示している．

図 7 では，「衆議院」と最も意味が近い単語は「参議院」となっている．これは現実と照らしあわせても，概ね正しい結果であると考えられる．それ以下の単語も政治関連の単語が多く，この「衆議院」の単語が，政治に関する意味をもつベクトルになっていると推測されるため，Word2Vec で正しいベクトル表現を学習することができたと考えられる．

次に，Word2Vec を利用して学習した単語ベクトルを利用し，事前学習とファインチューニングとテストデータで利用する文書のベクトル化を行う．この際にも，各単

図 7　衆議院と近いベクトルを持つ単語

語のベクトルを利用するために各文書を単語ごとに区切り，文書を単語の集合にする必要がある．そして，文書中の単語とその単語の分散表現を利用することで，文書をベクトル化する．その具体例を図 8 に示す．図 8 では，「昨日，衆議院選挙が行われた.」という文書をベクトル化する場合の手順を示している．文章で表すと以下の手ようにベクトル化を行っている．

●文章を「昨日」，「,」，「衆議院選挙」，「が」，「行う」，「れる」，「た」という各単語の基本形に分かち書きを行う．

●各単語のベクトルを，Word2Vec で出力したファイルから探す.

●「昨日」の 1 次元目の値，「,」の 1 次元目の値…「た」の 1 次元目の値を足していき，各単語のベクトルの要素の 1 次元目の和を計算する．

●上の手順を 200 次元分行い，200 次元分の値を作成する．

●文書の単語数がそれぞれの文書で異なっているため，その値を単語数で割り平均化し，これを文書のベクトルとする．

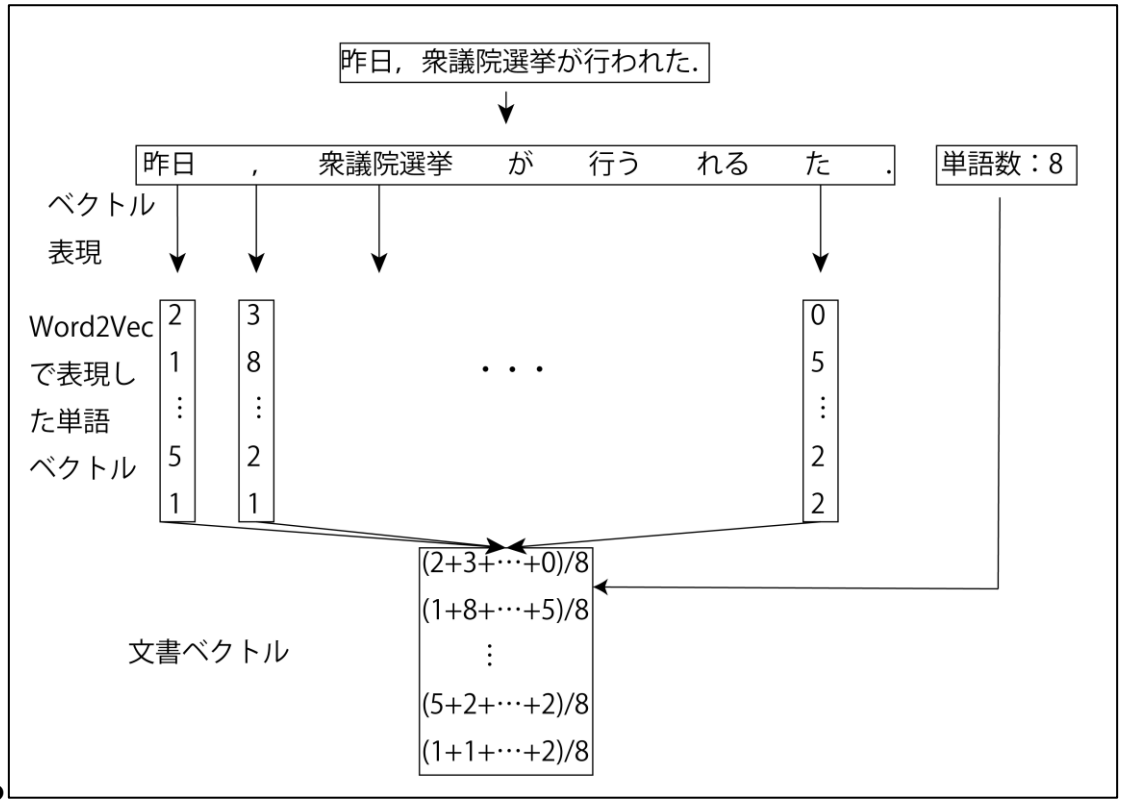

図 8 Word2Vec の単語ベクトルを利用した文書のベクトル化手順

つまり，文書中の単語の数を N，各単語のベクトル数を 200，i 個目の単語の j 次元の要素を w_{ij} とすると，文書ベクトル $\boldsymbol{d}$ は式(11)のように表すことが出来る．

$$\boldsymbol{d} = \left[\frac{\sum_{i=1}^{N} w_{i1}}{N} \cdots \frac{\sum_{i=1}^{N} w_{i200}}{N} \right] \tag{11}$$

式(11)は, 単純な各要素の和の平均であるが, Word2Vec で獲得できる分散表現では, ベクトルの数値は正も負も含んでいるため, 二乗平均等の負を考慮できない平均は扱えず, また Word2Vec の開発者である Mikolov が Word2Vec を用いて分類を [6]でも同様の文書ベクトルの算出方法を利用しているため，本研究でも式(11)の方法で文書をベクトル化した．

こうして，ニューラルネットワークに入力として与える教師ありデータ，教師なしデータ，テストデータを全てこの 200 次元のベクトルに変換する．

3.3. 事前学習を利用した学習

この節では，前節で作成した入力ベクトルを利用して，どのように学習と分類を行うかを説明する．

本研究では，学習時の教師ありデータを削減することを目的としている．そのために，深層学習の事前学習を利用することを提案している．事前学習は通常，深層学習で学習を行う際に，教師なしデータを用いて学習するデータの構造を理解させ，各重みの初期値を改善する補助的な役割で利用されている．しかし，提案する分類手法ではこの事前学習の件数をファインチューニングで扱う教師データの何倍も行うことで, ファインチューニングで分類精度を向上させるために必要な教師ありデータを削減させられるのではないかと考え提案している．また，ニューラルネットワークで文章を扱う際に，bag-of-words のような単語を直接扱っている表現方法では，次元数が膨大になり，それによりネットワーク型の学習記では，計算量が膨大になってしまうという問題がある．また，bag-of-words では似ている単語であっても，全く別の単語

として認識してしまい，学習する際にデータの構造を正しく理解できない，もしくは理解するために必要なデータが膨大になってしまう．事前学習で出来ることは，このベクトルの次元とこのベクトルの次元は同時に表れるため，重みを重くするといったことである．しかし，bag-of-words のような単語を単語として扱う表現方法では，「赤いりんご」と「赤いリンゴ」のようにいくら似た文章でも，全く異なる重みが修正されてしまう．そのため，それをカバーできるような膨大な言葉の使われ方を持つコーパスを学習しない限りは，正しいデータの構造は獲得できず，事前学習の意味がなくなってしまう．

しかし，本研究では Word2Vec を利用することで，この問題を解決出来ると考える．Word2Vec を利用して作成した入力は，全てのベクトルに共通して値が入っているため，変更されるベクトルも共通している．そのため，毎回の学習で全ての要素の重みを学習することが可能であり，事前学習の効果が bag-of-words を利用するときよりも増すと考えられる．さらに，大量のデータで事前学習を行うことで，どのベクトルの要素とどのベクトルの要素が共起しやすいかという特徴を学習することが出来るため，ファインチューニングを行う際に，通常よりも少量の教師ありデータでも，十分な学習を行うことができると考えられる．

ニューラルネットワークのモデル自体はシンプルなモデルで，入力層と出力層があり，その間に複数の中間層を挟む多層ニューラルネットワークである．入力層は，Word2Vec で変換した文書ベクトルの値を受け取るため，ノード数は 200 次元で固定され，次の層へのリンクがつなげられている．出力層は，カテゴリ数をノード数とし，前の層から全てのノードにリンクがつなげられており，各ノードからソフトマックス関数で変換された値一つを出力し，その出力が最も大きなカテゴリを，そのデータが所属するカテゴリであると判断する．また，中間層の形は可変なパラメータとしており，この形によって大きく精度が変わるため，後述する実験によってパラメータを最適なパラメータを決める必要がある．学習は，確率的勾配降下法とバックプロパゲーションを用いて行っている．確率的勾配降下法を行う際に必要な一回ごとのサンプル数のパラメータも実験によって定める．

学習の手順は，はじめに教師なしデータを用いて事前学習を行い，ネットワークの適切な初期値を定め，教師ありデータを用いたファインチューニングを行う．その後，テストデータを用いて分類を行う．

3.4. 実験内容

本研究では, 提案した分類手法の精度を高めるためにパラメータを最適化する実験と，最適化されたパラメータにおける精度評価の 2 つの実験工程を行う.

3.4.1.提案手法の精度向上

分類手法の精度を高めるための実験工程について説明する．深層学習では，ネットワークの形等の可変なパラメータが精度を大きく左右する．しかし，パラメータの決め方が確立されておらず，またタスクごとに最適なパラメータは異なっているため，実験でパラメータを調整していく必要がある.

今回，変更するパラメータは以下の通りである.

●中間層の数

入力層と出力層の間の中間層の数，2 層から 5 層まで変更する.

●ノード数

各中間層のノード数，50,100,200,500 と変更する.

●ノイズの発生確率

Denoising AutoEncoder のノイズの発生確率, 0.1 から 0.9 まで 0.1 刻みで変更する.

●エポック数

ファインチューニング中で同じデータを利用した学習回数である．少なすぎると，学習が不十分で終了し，多すぎると過学習を起こす可能性があるため，多いとよいものではない．10,50,100 と変更する.

●バッチ数

確率的勾配降下法で学習を行う際に，同時に勾配を計算する数，10,50,100 と変更する.

●打ち切り条件

ファインチューニングを行いすぎると，過学習が発生し，学習データ専用の学習器となってしまう可能性が高いため, 学習データのエラー率がどの程度になったら学習を打ち切るのかを表す条件，エラー率 0 から 30%まで 10%刻みで変化させる．また，エラー率が打ち切り条件で収束しない際は，学習を 100 回で打ち切る.

この 6 つのパラメータを変更していき, 最もテストデータの分類精度が良くなったときのパラメータを，本研究での最適なパラメータとする．しかし，全てのパラメー

タを検証していくと，そのパターンは約 5,200 通りとなり，今回は手動でパラメータの変更を行っており，さらに事前学習の件数が多いと一回の計算時間もかかってしまう．そこで本研究では，上記のパラメータの上から順に最適な値を求め，そのパラメータは他の値を変えた時も最適であると仮定して実験を行う．

3.4.2.提案手法の精度評価

精度評価についての実験工程の説明を行う．この実験では，本研究で提案した分類手法の精度が既存の手法と比べて，優れていたのかどうかを検証する．比較する対象の手法について説明する．本実験で，提案手法と比較する対象として選ぶ手法は，bag-of-words を利用したブベイズ分類器である．ナイーブベイズ分類器は，古い手法ではあるが，自然言語処理の分野では現在でもスパムフィルタや SmartNews などの Web の分野で分類に使われている．本研究では，最も基本なナイーブベイズ分類器との比較を行う．

ナイーブベイズ分類器とは，ベイズの定理を応用した確率的な機械学習の手法である．ナイーブベイズ分類器は，実装が他の機械学習手法と比較すると容易で，速度も早く精度も他の手法に比べて見劣りしないという特徴がある．以下に具体的なナイーブベイズの手法を説明する．

bag-of-words は，1.1 節で説明した通り，文章中に該当の単語が出現した場合，文章のその単語のベクトルを 1 にするという文章のベクトル表現の手法である．自然言語処理におけるナイーブベイズ分類器では，このベクトル表現を利用して，分類を行う．

文書 d がカテゴリ c に属する確率 P(c|d)はベイズの定理を利用すると，式(12)のように表すことができる．この P(c|d)が最も大きいカテゴリ c を文書 d が属するカテゴリをみなすことができる．

$$\mathrm{P}(c|d) = \frac{\mathrm{P}(c)\mathrm{P}(d|c)}{\mathrm{P}(d)} \tag{12}$$

式(12)中の P(c)はカテゴリの出現確率であり，学習データ中でカテゴリ c が出現した割合と等しい．つまり，学習データが 100 件で，カテゴリ c の出現回数が 20 である場合，P(c)は 0.2 となる．また，P(d)は比較する全ての P(c|d)で共通しているため，考慮する必要がなくなる．あとは P(d|c)を求めることができれば，P(c|d)が求まり，文書を入力として与えればカテゴリを出力として受け取ることができる．

P(d|c)はカテゴリ c 中で文書 d が出現する確率とみなすことができるが，文書 d と

全く同様の文書は基本的に出現しないため，通常では計算不可能である．そこで，文書 d を bag-of-words の考えと同様に単語の集合とみなすと，bag-of-words 中の i 番目の単語を w_i，語の数を N とすると，P(d|c)は式(13)のように表すことが出来る．

$$\mathrm{P}(d|c) = \prod_{i=1}^{N} \mathrm{P}(w_i|c) \tag{13}$$

P(w_i|c)は，学習データ中で，カテゴリ c が出現した時に，単語 w_i が出現した割合であるため，学習データから容易に計算することが出来る．しかし，この表現方法は単語同士がそれぞれ同時に発生する確率に偏りがないと仮定した場合にのみ，利用できる．また，カテゴリ c 中で単語 w_i が出現しない場合もあり，そのまま計算を行うと P(d|c)の値が 0 になってしまう．そのような問題をゼロ頻度問題と呼び，カテゴリ中の単語の頻度が 0 の場合にも何らかの値を与える必要がある．そのことをスムージングと呼び，本研究ではラプラススムージングというスムージング方法を採用した．この手法は上記のゼロ頻度問題を解決するための手法で，初期値を 0 ではなく，1 とすることで P(d|c)の値が 0 になることを回避できる．

実際に計算する際は，このまま行うと極端に値が小さくなってしまい，値がアンダーフローしてしまうため，対数をとって計算を行う．

このナイーブベイズ分類器と，本研究で提案する手法の比較を行う．手法の精度の比較方法の説明を行う．今回は４つの評価指標を利用して，精度の評価を行う．以下で 4 つの指標の説明を行う．

●Accuracy

テストデータ全てのうち，正しいカテゴリを出力できた割合であり，一つの手法につき一つ出力される．正解ラベルと同じ出力をした数を T，正解ラベルとは異なった出力をした数を F とすると Accuracy は式(14)のように表される．

$$\mathrm{Accuracy} = \frac{T}{T+F} \tag{14}$$

これは正解率を表しており，本研究で分類精度という言葉は Accuracy を指す．

●Precision

適合率と呼ばれるカテゴリごとに出力される値である．

出力カテゴリが i の場合の Precision は，分類器で i を出力して正解ラベルが i である場合の数を TT，実際の正解が i 以外である場合の数を TF とすると，カテゴリ i の

Precision は式(15)のように表される．

$$\text{Precision} = \frac{TT}{TT + TF} \tag{15}$$

つまり，Precision はカテゴリ i を出力したときに正しいカテゴリを出力できた割合であり，この値が高いと出力値の正確度が高いと言え，出力した値の信頼度が増すといえる．

●Recall

再現率と呼ばれるカテゴリごとに出力される値である．

出力カテゴリが i の場合の Recall は，Precision と同様の TT と，カテゴリ i 以外を出力し，カテゴリ i 以外であった場合の数を FT とすると，カテゴリ i の Recall は式(16)のように表される．

$$\text{Recall} = \frac{TT}{TT + FT} \tag{16}$$

つまり, Recall はカテゴリ i を出力したときに正解ラベルを再現できた割合である．この値が高いと正解に基づいた結果が出力できているといえる．

●F 値

F 値と呼ばれるカテゴリごとに出力される値である．

出力カテゴリが i の場合の F 値は, カテゴリ i を出力したときの Precision と Recall を使い，式(17)のように表される．

$$\text{F 値} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{17}$$

これは，Recall と Precision の調和平均を表している．通常，Recall と Precision はトレードオフの関係にあり，Recall を上げ過ぎると Precision が下がり，Precision を上げ過ぎると Precision が下がる．そこで，その調和平均である F 値が最も高いときが最もバランスがよく分類できていると言われている．

このような精度評価の実験を行い，提案手法とナイーブベイズ分類器の差を調べ，提案手法がナイーブベイズ分類器と比べ優れているかを評価する．また，提案手法とナイーブベイズ分類器で正解である文書を見比べ，ナイーブベイズ分類器と比較して，提案手法では，どのような文書が正しく分類でき，どのような文書が正しく分類できないのかを調べる．

3.5. 実験環境

本実験では分類と学習にかかる時間も計測するため，実験を行うコンピュータの環境や，使ったライブラリに関する記載を行う．

実験で利用したコンピュータの環境は表 1 に示す．

深層学習のために利用したライブラリの説明を行う．本研究では，深層学習を行うために，既存の深層学習用のライブラリである Pylearn2 を利用した．Pylearn2 は Python で記述された深層学習用のライブラリであり，本研究で利用した denoisingAutoencoder や確率的勾配降下法が実装されており，本研究ではコンピュータの環境上利用できなかったが，GPU の利用による高速化にも対応している．また，言語による速度の格差をなくすために，ナイーブベイズ分類器の実装も Python で行う．

表 1　実験環境

実験環境	
OS	OSX(10.10.5)
CPU	Core i5 (I5-4258U)
メモリ	8GB
プログラム言語	Python(ver.2.7.10)
深層学習ライブラリ	Pylearn2

また本研究で利用したデータについても説明を行う．

本研究で利用したデータは，Web ニュースサイトから収集したテキストデータであり，Word2Vec へ入力したテキストコーパスのニュースの件数は，1,600,000 件である．また，コーパス中に出現する単語の語彙数は 233.245 個である．

分類を行うカテゴリは”国内”，”国外”，”経済”，”芸能”，”スポーツ”，”IT・科学”の 6 カテゴリである．それぞれの分類基準は，まず該当のニュースデータが経済，芸能，スポーツ，IT・科学のどれかに分類されないかを考え，分類されるようであれば，そのカテゴリを適用させ，その他の事件や政治などは国内か国外に分類する．また，芸能人やスポーツ選手の不祥事などは，国内ではなく芸能やスポーツのカテゴリに分類し，社名が入っており株価に影響を及ぼすと判断したニュースは経済のカテゴリに分類する．

このような基準を定め，それぞれ教師データ 500 件とテストデータ 100 件を筆者以

外の第三者に分類を行ってもらうこととする．教師データ内の語彙数は 9,341 であった．

4.実験結果

この章では，今回行った二つの実験工程の結果を掲載する．

4.1. 精度向上のためのパラメータ調整

各パラメータの初期値は表 2 の値とし，実験内容の節で述べたようにパラメータを表 2 の左から 1 つずつ分類精度が高くなるよう調整していく．今回使う分類精度は，3.4.2 項における Accuracy である．また，精度を比較する際に分散分析を用い，有意差があると認められた場合は，平均値が最も良かったパラメータを最適とし，有意差があると認められなかった場合は最も計算時間が短かったパラメータを最適とする．

表 2 パラメータの初期値

	層の数	ノード数	ノイズの発生確率	エポック数	バッチ数	打ち切り条件
値	2	200	0.5	50	50	0.2

●層の数

2 層から 5 層の間で最適なパラメータを探す．各層の精度を五回出力した値とその平均を表 3 に示す．

表 3 層の数を変化させたときの精度の変化

層の数	2 層	3 層	4 層	5 層
精度[%]	79	65	64	66
	78	73	65	68
	74	66	67	69
	75	64	66	67
	73	69	65	68
平均[%]	75.8	67.4	65.4	67.6

この結果を元に層の数と各精度の関係に有意差がないという仮定の元，危険率を 5%とし，一元分散分析を行った結果，p 値が 1.71E-05 となった．そのため，層ごとに有意差が存在しているといえる．また，2 層の場合が最も平均精度が優れていたため，2 層が最も適したパラメータであると判断し，以下の実験における中間層の個数

パラメータは 2 層とする.

●ノード数

中間層のノード数を 50,100,200,500 の 4 パターンで，最適なパラメータの値がいくつかを探す．各ノード数の精度を五回出力した値とその平均を表 4 に示す.

表 4　中間層のノード数を変化させたときの精度の変化

ノード数	50	100	200	500
	62	72	74	78
	69	69	75	78
精度[%]	69	68	77	76
	69	77	77	77
	69	76	78	78
平均[%]	67.6	72.4	76.2	77.4

この結果を元にノード数と各精度の関係に有意差ないという仮定の元，危険率を 5%とし，一元分散分析を行った結果，p 値が 1.37E-04 となった．そのため，ノード数ごとに有意差が存在しているといえる．また，ノード数 500 の場合が最も平均精度が優れていたため，ノード数 500 が最も適したパラメータであると判断し，以下の実験におけるノード数のパラメータは 500 とする.

●ノイズの発生確率

denoisingAutoencoder におけるノイズの発生確率を 0.1 から 0.9 まで 0.1 刻みで変更し，最適なパラメータを探す．各ノイズの発生確率の精度を五回出力した値とその平均を表 5 に示す.

この結果を元に各ノイズの発生確率と各精度の関係に有意差がないという仮定の元，危険率を 5%とし，一元分散分析を行った結果，p 値が 0.70 となった．そのため，ノイズの発生確率ごとに有意差が存在しているとは言えない．また，計算時間はノイズの発生確率では有意差が存在しているとは言えないため，精度が最も優れていた 0.2 を最適なノイズの発生確率のパラメータとする.

表 5　ノイズの発生確率を変化させたときの精度の変化

発生確率	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
	77	78	76	77	78	78	75	77	78
	78	78	77	77	78	77	77	77	77
精度[%]	75	77	77	78	76	78	74	78	76
	77	77	78	76	77	77	79	77	78
	78	78	78	77	78	76	76	75	76
平均[%]	77	77.6	77.2	77	77.4	77.2	76.2	76.8	77

●エポック数

事前学習のエポック数を 10,50,100 の 3 パターンで，最適なパラメータの値がいくつかを探す．各エポック数の精度を五回出力した値とその平均を表 6 に示す.

表 6　エポック数を変化させたときの精度の変化

エポック数	10	50	100
	77	77	77
	76	78	76
精度[%]	78	77	78
	77	77	79
	76	77	78
平均[%]	76.8	77.2	77.6

この結果を元にエポック数と各精度の関係に有意差がないという仮定の元，危険率を 5%とし，一元分散分析を行った結果，p 値が 0.36 となった．そのため，ノード数ごとに有意差が存在しているとは言えない．また，計算時間はエポック数 10 が最も短いため，エポック数の最適なパラメータは 10 とする.

●バッチ数

確率的勾配降下法で同時に学習を行うバッチ数を 10,50,100 の 3 パターンで，最適なパラメータの値がいくつかを探す. 各ノード数の精度を五回出力した値とその平均を表 7 に示す.

表 7　バッチ数を変化させたときの精度の変化

バッチ数	10	50	100
精度[%]	78	77	77
	78	76	74
	78	78	77
	76	77	77
	78	76	77
平均[%]	77.6	76.8	76.4

この結果を元にバッチ数と各精度の関係に有意差がないという仮定の元，危険率を 5%とし，一元分散分析を行った結果，p 値が 0.22 となった．そのため，バッチ数ごとに有意差が存在しているとは言えない．また，計算時間はバッチ数 100 の場合が最も短かったため，バッチ数の最適な値は 100 とする．

●打ち切り条件

ファインチューニングにおいて，学習を打ち切る際の条件の値を決める．条件の対象は，教師データを今の学習器で分類したときのエラー率である．0 から 0.3 までを 0.1 刻みで 4 パターン検証する．各打ち切り条件の精度を五回出力した値とその平均を表 8 に示す．また，今回打ち切り条件 0 の場合はファインチューニングを 50 回行っても学習が終了しなかったため，そこで学習を打ち切った．

表 8　打ち切り条件を変化させたときの精度の変化

打ち切り条件	0	0.1	0.2	0.3
精度[%]	81	78	78	67
	81	78	76	68
	82	79	77	63
	80	78	76	68
	81	78	78	66
平均[%]	81	78.2	77	66.4

この結果を元に打ち切り条件と各精度の関係に有意差がないという仮定の元，危険

率を 5%とし，一元分散分析を行った結果，p 値が 1.34E-11 となった．そのため，打ち切り条件によって有意差があることがわかった．また，最も精度が良かった 0 を打ち切り条件の最適な値とする．

パラメータ調整を行った結果，表 9 が本提案手法でのパラメータの最適値となった．

表 9　提案手法のパラメータの最適値

	層の数	ノード数	ノイズの発生確率	エポック数	バッチ数	打ち切り条件
値	2	500	0.2	10	10	0

4.2. 提案手法の精度評価

4.1 節の実験で獲得した最適なパラメータを利用して，提案手法の精度評価を行う．評価方法に関しては，3.4.2 項に記載した．

表 9 のパラメータで提案手法を五回実行した際の，精度が最も良かった場合の各カテゴリの precision，recall，f 値の結果を表 10 に示す．

表 10　提案手法の精度

提案手法	precision	recall	f 値	件数
国内	0.91	0.88	0.89	33
国外	0.64	1	0.78	9
経済	0.67	0.86	0.75	14
エンタメ	0.9	0.82	0.86	11
スポーツ	0.86	0.95	0.9	19
IT・科学	1	0.36	0.53	14
平均/合計	0.85	0.82	0.81	100
分類精度	0.82			
平均計算時間	106.51			

また，比較対象としてナイーブベイズ分類器で同様の学習データとテストデータで実験を行った場合の精度も記載する．ナイーブベイズ分類器では乱数は使用されないため，出力

される精度は一通りである. ナイーブベイズ分類器で各カテゴリの precision, recall, f 値の結果を表 11 に示す.

表 **11** ナイーブベイズ分類器の精度

ナイーブベイズ	precision	recall	f 値	件数
国内	0.65	0.85	0.74	33
国外	0.33	0.44	0.38	9
経済	0.53	0.64	0.58	14
エンタメ	0.86	0.55	0.67	11
スポーツ	0.84	0.84	0.84	19
IT・科学	1.00	0.14	0.25	14
平均/合計	0.71	0.65	0.63	100
分類精度	0.65			
平均計算時間	6.31			

5.考察

本章では，4 章の実験結果に関する考察を行う．

5.1. パラメータに関する考察

表 3 から表 9 のパラメータの最適化についての実験に関する考察を行う．

今回検証したパラメータの中で，分散分析の結果で有意差があると判断されたパラメータは，中間層の数，中間層のノード数，打ち切り条件の三つであった．

●中間層の数

中間層の数は，今回実験を行った中で最小の 2 層が，最も優れた精度をあげた．また，3 層から 4 層へは精度が悪化し，4 層から 5 層へは精度が改善されたが，どれも 2 層の場合と比較すると，8%程度の差が出ている．通常，深層学習を行えば中間層の数が増えれば増えるほど，ネットワークの表現力が上がり，精度が上がると言われている．また，類似研究[]でも，中間層が 0 から 2 の場合でしか検証は行われていないが，中間層の数を増やすと精度があがるという結果が出ている．しかし，今回の実験では，通常の見解とは異なる結果が得られた．これは，層を増やしたことによるネットワーク上のリンクの増加に対応出来るだけの教師データが存在していなかったためであると考えられる．近年の深層学習の進歩は，教師データの増加も理由の一つとして挙げられているため，層を重ねる場合は教師データを大量に用意する必要が有ることがわかる．そのため，本研究のような教師データを少量に抑えたい場合は事前学習を使った深層学習ではあるものの，中間層の数は増やし過ぎるべきではないと考えられる．

また，中間層二層で十分な精度をだせた理由としては，入力に Word2Vec を利用したためということが考えられる．Word2Vec 自体が Autoencoder に似た仕組みのニューラルネットワークであるため，Word2Vec で作成された入力が既に文章の特徴を表現したベクトルであり，深い表現力を持ったネットワークでなくても，十分な学習を行うことが可能になったのではないかと考えられる．

●ノード数

中間層のノード数は，今回の実験では実験中の最大ノード数である 500 が最も精度が高いという結果になった．また，ノード数を小さくすればするほど，分類精度は下

がっていくという結果が得られた．ノード数が 200 より低い場合は，次元圧縮を行うことと同値であり，表現力が下がるため精度が落ちることはわかる．特に本実験では，既に Word2Vec で入力を次元圧縮したあとであるため，更にそれ以上の圧縮を行うと，文章を分類するために必要な情報が消えていってしまうのではないかと考えられる．

また，入力層と同じ場合とそれ以上である 200 と 500 では，ほぼ同値の精度となり，共に精度が高かったため Word2Vec で入力を作成した際には，中間層のノード数は入力層のノード数と同数，もしくはそれ以上にすることにより，十分な表現力のまま学習を行うことができ，精度が向上すると考えられる．

●打ち切り条件

ファインチューニングの打切り条件は，今回の実験では最も低い 0 が最も良い精度を出した．また，打ち切り誤差を 0 としたが，実際はファインチューニングを 50 回繰り返したところで，教師データのエラー率は 0 にならなかったため，1%から 5%程度の部分で学習は打ち切られている．通常，学習データに対する誤差である訓練誤差を低くし過ぎると，テストデータに対する誤差である汎化誤差があがってしまう．そのような状態を過学習と呼び，図 9 のような形状になることが多い．しかし本実験では，打ち切り条件が下がるほど，つまり訓練誤差を下げるにつれ汎化誤差も下がっていった．これは今回の実験では図 9 のような過学習を起こさずに学習を行えていると

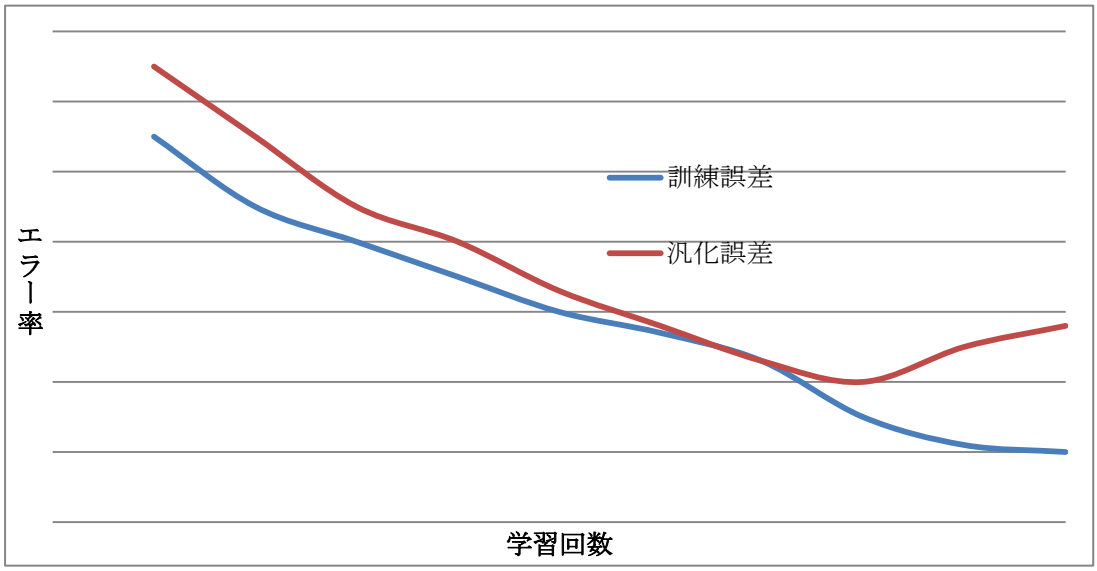

図 9　過学習のイメージ図

いうことである．

このことからテストデータで正しいと判定された八割のデータも訓練データ中のデータと極度に似ていたと考えられる．つまり，今回のようなニュースデータの分類というタスクでは，Word2Vec を用いた文書の表現は二割の例外を除けば，芸能カテゴリは，このベクトルとこのベクトルの数値が大きいというようなカテゴリごとに特徴があるベクトルになっていると考えられる．

今回のような教師ありデータを少数としたニュースデータの多値分類というタスクでは，これらのパラメータによって，精度が大きく変わり，最適なパラメータは上記のようになることがわかった．しかし，このパラメータは文書分類全体に適用できるものかは定かではないため，他のタスクで実行する場合には，参考程度にし，他のパラメータも考慮するべきである．

5.2. 精度評価の実験に関する考察

本節では，4.2 節で行った提案手法の精度評価に関する考察を行う．精度評価の実験では，提案手法とナイーブベイズ学習器の精度の比較を行った．

結果として，提案手法で最も精度がよかった場合では，ナイーブベイズ学習器の精度と比べて，17%の差があり，平均でも 15%以上良い結果が得られた．これは教師データが少ないためであると考えられる．学習データが少ない場合，ナイーブベイズ分類器で用いた bag-of-words では，学習できる語彙が少なくなるため，テストデータ中の単語で教師データ中の頻度がゼロである単語が増えてしまう．また，bag-of-words では，出現した単語そのものしか扱えず，教師データの中に類似語があっても学習することはできない．しかし，Word2Vec では教師なしデータから語彙を獲得し，数値化を行うため，教師データに存在しない語彙も扱うことが可能であり，図 7 の衆議院の例のように似た単語は似たベクトルになっているため，単語同士の関係も考慮した学習を行うことが出来る．そのような，教師データと教師なしデータに含まれる語彙や情報量の差が精度に影響していると考えられる．

また，教師なしデータによる事前学習を利用してデータの構造を学習したあとに，ファインチューニングを行ったことも，その差に貢献していると考えられる．その根拠は事前学習をせずに，ファインチューニングを行った際の精度はナイーブベイズ分類器の精度よりも低かったためである．事前学習時の教師なしデータの件数と精度の

推移を表した図を図 10 に示す．図 10 の横軸は対数を取っており，事前学習を行わなかった場合の 0 件の場合を，対数をとる関係上データ件数を 1 として図にあらわした．

図 10 からわかるように，事前学習を行わなかった場合の精度はナイーブベイズ分類器の精度よりも悪く，1,000 件の時点でナイーブベイズ分類器の 65%という精度を超えている．また，教師なしデータの件数が一万件を超えてからは，あまり精度に差がでていない事がわかる．本研究では，大量の教師なしデータを事前学習させ，そのデータが多いほど精度を向上させることが可能であるのではと考えていたため，10,000 件以上はあまり効果が出ない点は望んだ結果を得ることが出来なかった．しかし，この結果のように事前学習を行わない場合より，行う場合のほうが明らかに良い結果が得られたため，教師なし学習の効果はあったと考えられる．また，本当に有意差があったのかを確認するために，教師なしデータの件数とその分類精度の関係に有意差がないという仮定で一元分散分析を行った結果，p 値が 1.95E-45 となり，

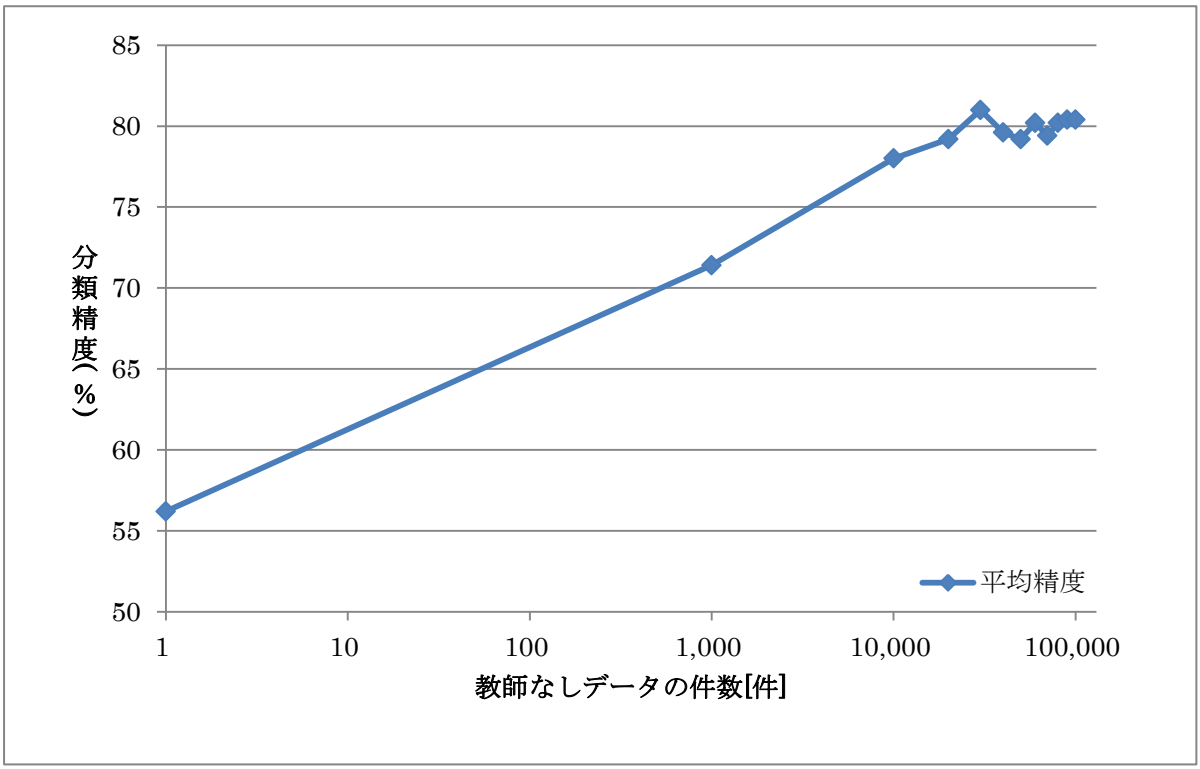

図 10　分類精度と教師なしデータの件数の関係

有意差があるという結果が得られたため，教師なしデータの件数と分類精度は関係があるという結果が得られた．

また，精度のみではなく提案手法とナイーブベイズ分類器における正解のカテゴリを出力できた文書の違いに関する考察も行う．

はじめに，テストデータ中の文書における単語数と正解率との相関について考察を行う．図 11 はテストデータの単語数ごとのヒストグラムである．階級数はスタージェスの公式によって定めた．また，テストデータ中の最小単語数は 31，最大単語数は 665 であり，平均単語数は，211.41 であった．この数は通常の新聞等のニュースと比較すると，著しく少ない．また，ナイーブベイズ分類器で分類した結果の正解と不正解の二値の値と，その文章の単語数の相関をとったところ，相関係数は 0.072 となりほぼ相関がないと考えられる結果が得られた．提案手法の場合の相関係数は-0.023 となり，こちらもほぼ相関がないと考えられる結果が得られた．実際にデータを見ても，単語数が多い文書が正解できていたり，その逆であったりという傾向は全く見られなかった．

つぎに，提案手法では正解することができ，ナイーブベイズ分類器では正解することができなかった文章に関する考察を行う．提案手法で正解が可能で，ナイーブベイズ分類器で正解が不可能であったテストデータの件数は 19 件であった．例として，この文章が提案手法のみで正解をとることができていた．

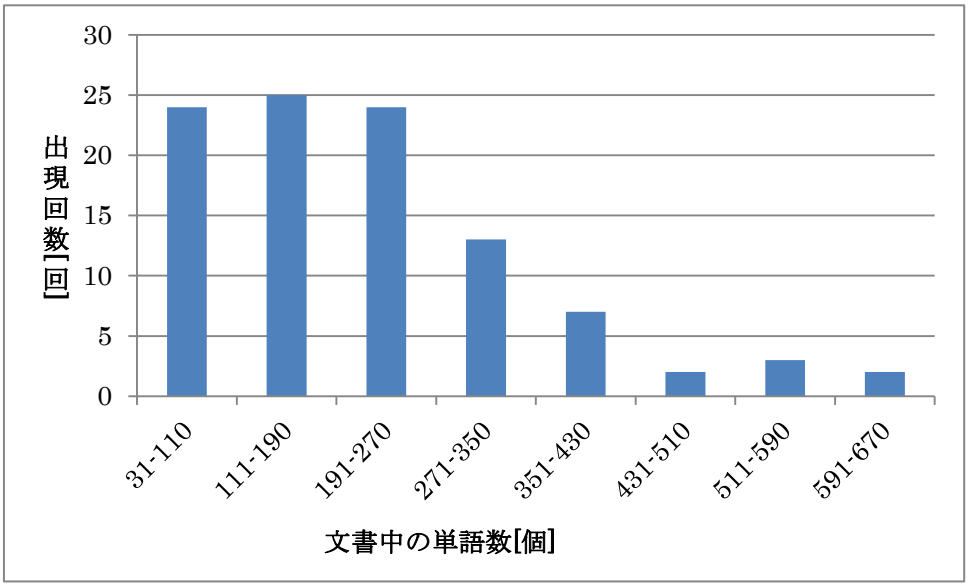

図 11　テストデータ中の文書の単語数に関するヒストグラム

「政府は 20 日午前，宇宙飛行士の若田光一さんに内閣総理大臣顕彰を授与することを決めた．授与式は 25 日午前 10 時から首相官邸で行う．」

この文章は，正解ラベルを IT・科学としていたにもかかわらず，ナイーブベイズ分類器では国内というカテゴリが出力された．これは，教師データ中に宇宙に関するデータが存在しなかったため，「宇宙飛行士」や「岩田光一」という単語の意味を取ることができず，「内閣総理大臣」や「首相」といった政治の単語によってカテゴリが定まってしまったため，政治ニュースが割り当てられる国内というカテゴリに分類されてしまったと考えられる．また，もう一つの例をあげる．

「母親の育児放棄のために飼育係に人工飼育され，その愛くるしさから世界中で人気者となったベルリン動物園のシロクマ「クヌート」が，同じ飼育場の年上のメスグマたちから攻撃されている．「新入り」に対する先輩グマの洗礼の可能性もあるが，ファンらは「いじめでは？」と心配している．３歳になったクヌートはすっかり大きくなったが，今でも同動物園随一の人気者だ．独ビルト紙などによると，クヌートは先月，母親のトスカ（２４）といずれもメスのナンシー（２１），カチューシャ（２４）と同じ飼育場に移されたが，３頭から脅されてうずくまっている様子やカチューシャから池に突き落とされるシーンが目撃された．動物園は「クヌートは年上の３頭と比べるとまだ筋力が足りない」としているが，ファンの心配をやわらげるためか，クヌートのお相手となるような若いメスグマを探しているという．」

この文章は，正解カテゴリが海外であったが，ナイーブベイズ分類器の出力は，国内であった．このニュースが海外のニュースであるとわかるのは，「ベルリン」や「独」の部分の単語からのみである．先ほどの宇宙飛行士の例も「宇宙飛行士」や「岩田光一」という特定の単語からのみで，カテゴリが定められる．その他のニュースデータも，芸能人が事件を起こしたというニュースや，スポーツ界の不祥事などでこのような，提案手法とナイーブベイズ分類器の差が出ている．

教師データが十分あり，宇宙飛行士が IT・科学のカテゴリにのみ多く出現しているということがわかっていれば，ナイーブベイズ分類器でも分類が可能であると考えられるが，今回のような少量の教師データでは，それは不可能であると考えられる．しかし，提案手法ではこれらの分類を可能にしている．これは，Word2Vec を利用することで，教師なしデータの単語も学習することが可能であるため，教師データの文書中に出現しなかった単語も扱えるためではないかと考えられる．また，そのような特

徴的な単語が含まれていた場合に，その単語のベクトルによって文書のベクトルのカテゴリが変化するように文書ベクトルの作成が上手くできており，またその少数の単語のベクトルによる文書ベクトルのカテゴリごとの差をニューラルネットワークが上手く学習できていると考えられる．このように教師データが少量の場合に，ナイーブベイズ分類器で学習が不可能で，提案手法では分類が可能というニュースデータには，その文章のカテゴリを定める何らかの特徴的な単語が含まれていることが多いことがわかった．

つぎに，ナイーブベイズ分類器で正解の出力が可能で，提案手法で出力が不可能であった場合のニュースデータについて考察を行う．そのようなニュースデータはテストデータ中に 3 件存在した．その例として以下のニュースデータを参考にする．

「４日午後２時半ごろ，東京都港区元麻布３丁目の中国大使館正門前で，男が鉄柵（さく＝高さ９７センチ，幅１６１センチ）を乗り越え，敷地内に侵入した．警備中の警視庁機動隊員がその場で取り押さえ，建造物侵入容疑で現行犯逮捕した．同庁によると，男は，１９８９年の中国民主化運動の学生リーダーで，台湾籍の投資家ウアルカイシ容疑者（４２）とみられる．４日は天安門事件から２１年にあたる．同庁によると，ウアルカイシ容疑者は同日午後１時半ごろから仲間数人と正門前で抗議活動をしていた．現場にいた活動家らによると，「中国への帰国を求めて大使館と交渉する」と言って柵を越えたという．ウアルカイシ容疑者は天安門事件後海外に逃れ，９６年から台湾に拠点を移し「中華民国籍」を取得した．事件２０周年の昨年６月４日にも，中国帰国を目指し台湾からマカオに向かったが強制送還された.」

このニュースデータは提案手法では，海外と判断された．しかし，実際は国内のニュースであるため，正解カテゴリは国内となっている．これは先程の例とは逆に，「中国」や「天安門事件」等の海外の特徴的な単語の影響を受け，国内ではなく海外のニュースであると判断されたと考えられる．また，このニュースデータは見方によっては海外のニュースでもあるため，そのような線引きの難しいニュースデータを分類することは提案手法では困難であると考えられる．

また，3 件のうち，ナイーブベイズ分類器で分類が可能であったが，提案手法で分類が失敗した理由が不明であったニュースデータも存在した．それが以下のニュースである．

「大人気アニメ「進撃の巨人」ＯＰが快進撃！人気アニメ「進撃の巨人」（ＴＯＫ

ＹＯＭＸなど）のオープニング曲が収録されたマキシシングル「自由への進撃」が，２２日付オリコン週間シングルランキングで初登場２位に入ったことが１５日，分かった．音楽クリエーター，Ｒｅｖｏ（年齢非公表）のソロプロジェクト「ＬｉｎｋｅｄＨｏｒｉｚｏｎ（リンクトホライズン)」の楽曲で，初週売り上げ１２．９万枚を記録．今年発売のアニメソングでは１位の“快進撃”だ．聞き慣れぬアーティストが，オリコンランキングの上位に名を連ねた.」

このニュースの正解カテゴリは，芸能であり，「オリコン」や「音楽」や「アーティスト」といった芸能関連の単語が文章中に多く出現しているにも関わらず，提案手法での出力カテゴリは経済であった．経済関連の単語がほとんど含まれていないにも関わらず，経済と出力してしまうのは単語のベクトル表現の次元を圧縮してしまったデメリットであると考えられる．そのため，時折全く出現単語に関係のない出力をしてしまうおそれがあると考えられる．

このように，ナイーブベイズ分類器における分類結果と提案手法における分類結果にはいくつかの差異があることがわかった．特に，対象データに特殊な単語を含み，その単語によってカテゴリが変化するような場合に提案手法は向いていると考えられる．

6.おわりに

本研究では，従来の分類手法の学習で大量に必要であった教師データの削減を目的とした研究を行った．また，教師データを削減する代わりに事前学習と文書のベクトル化において教師なしデータを利用することを提案した．

従来の類似研究との違いは，タスクが多値分類である点，対象がニュースデータである点，少量の教師データで行っている点等があげられる．また，深層学習はパラメータで大きく精度が変わるにもかかわらず，従来の論文内で触れられてこなかったパラメータによる精度の変化についての検証を行った．その検証の結果，パラメータ次第で精度が最大 10%程度変わることがわかり，今回の分類タスクでの重要なパラメータは中間層の数，ノード数，打ち切り条件であることがわかった．

また，従来スパムフィルタなどで利用されているナイーブベイズ分類器との精度比較も行った．その結果，平均精度はナイーブベイズ分類器を 15%程度上回る結果を得ることが出来た．また，その分類結果にナイーブベイズ分類器と提案手法での分類結果には，いくつかの特徴があるという考察を行った．

目的であった教師なしデータの利用は，Word2Vec の部分では十分に利用できたが，事前学習の部分では，10,000 件以上では効果が薄いことがわかった．この対策として，教師なしデータを事前学習のみではなく，半教師あり学習で役に立てた場合，更に教師なしデータを多く利用できるのではないかと考えられる．

この研究は，新たにデータ分析を組み込んだサービスを始めたいが，元となる教師データがないという状況の人や企業でも，少量の教師データから学習を行うことが可能となるため，そのような分野で役立てることができると考えられる．

参考文献

[1] 総務省情報通信国際戦略局情報通信経済室, “ビッグデータ時代における情報量の計測に係る調査研究”, 総務省, 2014.

[2] A. McCallum and K. Nigam, “A Comparison of Event Models for Naive Bayes Text Classification”, Workshop on Learning for Text Categorization, In the Fifteenth National Conference on Artificial Intelligence, 1998

[3] 平博順, 向内隆文, 春野雅彦, “Support Vector Machine によるテキスト分類”, 情報処理学会研究報告, NL—128, pp.173–180, 1998.

[4] 永田純平, 新納浩幸, 佐々木稔, 古宮嘉那子, "文書分類をタスクとした Pylearn の Maxout+Dropout の利用", 言語処理学会発表論文集, Vol.21, pp.900–903, 2015.

[5] 菅原 拓夢，笹野 遼平，高村 大也，奥村 学，"単語の分散表現を用いた語義曖昧性解消"， 言語処理学会発表論文集, Vol.21, pp.648–651, 2015

[6] T. Mikolov, W. Yih and G. Zweig, “Linguistic Regularities in Continuous Space Word Representations”, NACCL13, pp.746–751, 2013.

[7] J. Rennie, ” Improving multi-class text classification with naive bayes”, Master’s thesis, Massachusetts Institute of Technology, 2001

[8] 熊沢逸夫，”学習とニューラルネットワーク”，森北出版，1998

[9] Geoffrey E. Hinton, R. R. Salakhutdinov, ” Reducing the Dimensionality of Data with Neural Networks”, Science 313 (5786), pp.504–507

[10] Lee, K. H, “Text Categorization with a Small Number of Labeled Training Examples”, Unpublished Doctor of Philosophy, University of Sydney, 2002.

[11] 野沢健人，中岡義貴，山本修平，佐藤哲司，"word2vec を用いた代替食材の発見手法の提案"， 信学技報, Vol. 114, No. 204, pp. 41–46, 2014

[12] 加藤和平, 大島考範 , 二宮崇, ” word2vec と深層学習を用いた大規模評判分析”, 言語処理学会第 21 回年次大会発表論文集，pp.525–528，2015

[13] B. Xue, C. Fu, and Z. Shaobin, “A Study on Sentiment Computing and Classification of Sina Weibo with Word2vec,” 2014 IEEE Int. Congr. Big Data, pp. 358–363, 2014.