卒 業 研 究 報 告 題 目

文書間類似度計算のための
単語の情報量最大化クラスタリング

Invariant Information Clustering of Words for Calculating Similarity between Documents

指 導 教 員　　出口 利憲 教授

岐 阜 工 業 高 等 専 門 学 校　電気情報工学科

2019E18 相撲　美月

令 和 6 年（ 2 0 2 4 年 ）　　2 月　16　日 提 出

Abstract

In this study, an unsupervised learning clustering method is proposed, utilizing Word2vec and Invariant Information Clustering. This method is an unsupervised learning method in which a neural network is trained to maximize the mutual information of a word vector and a vector that is a slightly transformed from it at random. To verify the effectiveness of the proposed method, experiments were conducted to calculate the similarity between words using the following two clustering methods.

1. Invariant Information Clustering
2. Dimensionality reduction method using hierarchical clustering

The second method is the dimensionality reduction method proposed in previous studies. In order to improve the accuracy of this method, the over-clustering rate and the number of units in all combined layers were varied and verified. The text data used in this study are articles from livedoor news. The effectiveness of the method was verified using a WordCloud created from the results of the clustering. Experimental results confirmed the effectiveness of clustering by Invariant Information Clustering over hierarchical clustering, a conventional method.

目 次

第1章　序論

現代社会は急速な情報化により、日常生活のあらゆる側面がデータ化され、個人はタブレットやスマートフォンなどの端末を利用して日々の意思決定に活用している。これにより膨大な量のデータが蓄積され、それらのデータは生活の一環となり、社会の基盤を支える一方、今後もその量は増加し続けることが予想されている。結果、ソーシャルネットワーキングサービスの普及に伴い、信頼性に欠ける情報や虚偽の情報も広まっており、情報リテラシーが重要視されている。現代社会ではその膨大なデータから有益な情報を見極める難しさがあり、個人が信頼性のある情報を見極めることは困難である。この課題に対処する技術の一つがデータマイニングであり、統計学やパターン認識、人工知能などの手法を用いて有益なパターンやルールを発見するものである。特に、自然言語処理技術を応用したテキストマイニングは、テキストデータの分析に活用されている。しかしこのテキストマイニングは、対象が自然言語であり、単語や接続詞などの意味を捉える必要があるため、他の数値を対象とするようなデータマイニングより難易度が高く、未だに完成された技術ではない。

本研究では、文書の類似度計算に活用される、クラスタリングという手法において、2018年に新たに発表された情報量最大化クラスタリングを実装し、その有効性の確認や他手法との比較を目的として実験を行う。この手法は、元々画像分野にて、成果を上げている手法であるが、今回は単語の分散表現であるWord2vecと情報量最大化クラスタリングを用いた、教師なし学習によるクラスタリング手法を提案する。これによりこれまでの手法より、高い精度でのクラスタリングが可能になると考えられる。類似度を計算するテキストデータの対象には、株式会社ライブドアが提供するlivedoorニュースに現在掲載されている記事と、株式会社ロンウイットが公開しているlivedoorニュースコーパスを使用した。前者は「スポーツ」「食べ物」「教育」「産業」「気象」の5つのジャンルからいくつか記事を抽出したものを使用し、後者は9種類のニュース記事が計7367本収載されており、そのうち7ジャンルからニュースを取得し使用した。これらはジャンルに分かれているため、類似した単語の評価がしやすいと考えた。

実験では情報量最大化クラスタリングと、階層的クラスタ分析であるward法でのクラスタリング結果を比較する。また使用した文章によるクラスタリングの精度についても比較し、検証する。学習モデルの訓練回数、指定するクラスタ数、全結合層のユニット

数、オーバークラスタリング率によるクラスタリング結果を、比較対象とする。クラスタリングした結果は、Wordcloud によって可視化し、同一のクラスタにどのような単語が分類されたかによって、精度を検証する。

第2章　自然言語処理

2.1　テキストマイニング

2.1.1　データマイニング

データマイニングとは、データベースに蓄積された大量のデータから統計学や人工知能によって、情報の傾向を見出す技術である。企業のマーケティング戦略や顧客管理などで活用されており、一般に広く浸透している技術である。[1)]

2.1.2　テキストマイニング

テキストマイニングとはデータマイニングの一種であり、対象が文書データに限られている。自然言語解析の手法を使って、文章を単語（名詞、動詞、形容詞等）に分割し、それらの出現頻度や相関関係を分析することで有益な情報を抽出する。ビッグデータの活用においても、テキストマイニングは非常に重要な要素となる。

2.1.3　形態素解析

形態素解析とは、自然言語処理（NLP）の一部であり、自然言語で書かれている文を、文法や品詞の情報をもとに形態素に分解し、一つ一つの品詞や変化などを判別していくことである。これにより単語の出現頻度の計算や特定の品詞のみを抽出するといった処理が可能となる。「形態素」は言語学の用語であり、意味を持つ表現要素の最小単位のことである。

テキストマイニングにおいて形態素解析が行われる理由は、多くのテキストマイニングでは単語を入力値として与えて処理するため、日本語は英語とは異なり、文章がどこで区切れるか分かりにくいためである。形態素解析を行うためのツールは形態素解析器と呼ばれ、いくつかの形態素解析器はオープンソースで公開されている。本研究では MeCab というオープンソースのソフトウェアを使用した。

2.1.4　MeCab

MeCab とは、京都大学と日本電信電話株式会社 (NTT) が共同開発したオープンソースの形態素解析エンジンのことである。MeCab は日本語に対応しており、日本で使用される形態素解析エンジンの中でもメジャーである。言語、辞書、コーパスに依存しない

汎用的な設計方針を採用しており、C 言語、C++、Java、Python 等、数多くの言語で使用することが可能である。MeCab では、品詞等の情報が記録された辞書を用意し、形態素解析を行うことができる。例として以下の文を形態素解析し、形態素、品詞、標準形等を出力した結果を示す。[2)]

「すももももももものうち」

すもも　名詞, 一般,*,*,*,*, すもも, スモモ, スモモ

も　助詞, 係助詞,*,*,*,*, も, モ, モ

もも　名詞, 一般,*,*,*,*, もも, モモ, モモ

も　助詞, 係助詞,*,*,*,*, も, モ, モ

もも　名詞, 一般,*,*,*,*, もも, モモ, モモ

の　助詞, 連体化,*,*,*,*, の, ノ, ノ

うち　名詞, 非自立, 副詞可能,*,*,*, うち, ウチ, ウチ

2.2　自然言語

2.2.1　自然言語と人工言語

自然言語は、人間が日常的に使用する、意思疎通を行うための言語である。例として、日本語や英語、中国語などが挙げられる。対して人工言語は人間がコンピュータに処理させるための言語である。例として、C 言語や Python といったプログラミング言語などが挙げられる。自然言語においては、同一の文章でも読み手によって複数の解釈がある場合、伝える際に曖昧な表現を使用しても、相手が解釈することで伝わる場合がある。しかし人工言語では、解釈は一通りしか認められない。命令に対して複数の解釈があると、コンピュータの処理が毎回変化してしまうのを防ぐ為である。

自然言語は規則が曖昧なため、使用する単語を入れ替えたり、単語の順番を入れ替えたりすることができる。感情でも文章を制御しやすいため、形に縛られない自由な情景表現が可能である。

2.2.2　自然言語の曖昧性

自然言語の曖昧性（Ambiguity）とは、言語表現や文脈が複数の解釈や意味を持つ状態を指す。言葉や文章が一意に解釈できない状態が生じることで、曖昧性が発生する。

曖昧性には多義性と類義性の二つの種類がある。多義性とは、ある単語が複数の意味

を持っており、可能な解釈が広がること、すなわち単語の意味が完全に一つに定義できないことを指す。多義性の例として「潰れる」という単語を上げる。「箱が潰れる」という文では、外部からの力を受けて、もとの形が崩れることを表しているが、「あの店は潰れてしまった」では経営・生活などが成り立ってゆかなくなるという意味で使われている。類義性では異なる単語が同じ意味を示す性質のことを指す。「用意」と「準備」といったような、読み書きが異なる場合においても似た意味を持つ単語が例として挙げられる。

自然言語の曖昧性は、言語理解や機械翻訳、情報検索、テキストマイニングなどの分野で課題となっている。これを解決するためには、文脈や周囲の情報を考慮して意味を推測する手法やアルゴリズムが必要とされる。

第3章　学習

3.1　機械学習

3.1.1　機械学習とは

機械学習とは、Machine Learningとも呼ばれ、コンピュータに大量のデータを読み込ませ、一定のルールやパターンを見つけ出し学習させることで、同じような課題に直面した際に、以前学習したルールやパターンを用いることで、良い推測や判断を行うことができるデータ解析技術である。機械学習を支える技術の一つがニューラルネットワークである。またこのニューラルネットワークの学習能力を高める一つの手法がディープラーニングである。

3.1.2　教師あり学習

学習手法のうち最も代表的なものが教師あり学習である。人間が事前に正解のデータ（ラベル）を入力し、その正解のデータと比較して正しいかどうかを判断させる。コンピュータにとって判断基準が明確であるため、「正しいか、間違っているか」の問題を解決するのに適している。教師あり学習のアルゴリズムで代表的なものは、「回帰」と「分類」である。

3.1.3　教師なし学習

教師なし学習は、教師あり学習とは異なり、正解のデータを教えずに学習を行う。大量のデータを学習させることでデータの特徴やパターンを学習する。データ内に存在する未知のパターンを見つけたいときに適している。教師なし学習のアルゴリズムで代表的なものは、クラスタリングである。[3)]

3.1.4　特徴量

機械学習の活用のうえで重要な概念の一つが特徴量である。特徴量とは対象となるデータの特徴を数値にして表したものである。自然言語処理の場合、テキストデータにおける特徴量とは、ある単語の出現頻度や重要度の指数、単語を数値化したベクトルなどのことを指している。これらは非構造化データとも呼ばれ、エクセルといった「列」と「行」の概念を持つ構造化データに比べて、取得・解釈・利用が難しい。近年のインターネット

のさらなる普及により、これらの非構造化データを上手く活用する技術が重要度を増している。そこで非構造化データには、ニューラルネットワークを適用することが多い。[4)]

3.2 ニューラルネットワーク

ニューラルネットワークとは、人間の脳内にある神経細胞（ニューロン）とそのつながり、つまり神経回路網を人工ニューロンという数式的なモデルで表現したものである。ニューロンは信号を受け取った後、次へ情報を伝えるためにシナプスを作る。その際シナプスの結合強度によって、情報の伝わりやすさが変わる。

ニューラルネットワークで、伝達された情報は、Figure 3.1 に示す入力層、隠れ層、出力層の順に処理される。入力層は、人工ニューロンが最初に数値の情報を受け取る層のことである。その後受け取った情報を、次の隠れ層に転送する。隠れ層は中間層とも呼ばれ、入力層から情報を受け継ぎ、取り込んだ複雑なデータを選別し、学習によって扱いやすい状態に変換する層である。隠れ層の数は決まりがなく、層が多い程、複雑な分析が可能となり、隠れ層が 3 層以上あるニューラルネットワークを用いた機械学習の手法やその周辺の研究領域のことをディープラーニングと呼ぶ。出力層では、隠れ層で処理された信号が送られ、隠れ層での計算の最終結果を伝達する。あるニューロンから次のニューロンへの出力過程において、入力された数値を特定の方法で変換し、その結果を出力する関数を活性化関数と呼ぶ。ニューロンの数や中間層が増えるほど、分析の柔軟性や結果の表現力は向上する反面、データやメモリ、演算量は増加する [5)]。

活性化関数にはシグモイド関数、ソフトマックス関数、ReLU 関数、Leaky ReLU 関数などがある。例としてソフトマックス関数を挙げる。これは入力データ内の複数の値を 0.0〜1.0 の範囲の確率値に変換する関数であり、データがどのクラスに属するか判断するような分類問題に用いられる。この関数によって出力される複数の値の合計は常に 1.0 となる。ソフトマックス関数はニューラルネットワークにおいて、入力値ベクトルの各ベクトルに対する確率値に相当する出力値ベクトルに変換する役割を持っている。ここでソフトマックス関数は以下に示す式で表すことができる。y_i は i 番目の出力、x_i は i 番目の入力のことである。分母はすべての入力信号の指数関数の和、n はデータ数を表す。[6),7)]

$$y_i = \frac{e^{x_i}}{\sum_{k=1}^{n} e^{x_k}} \tag{3.1}$$

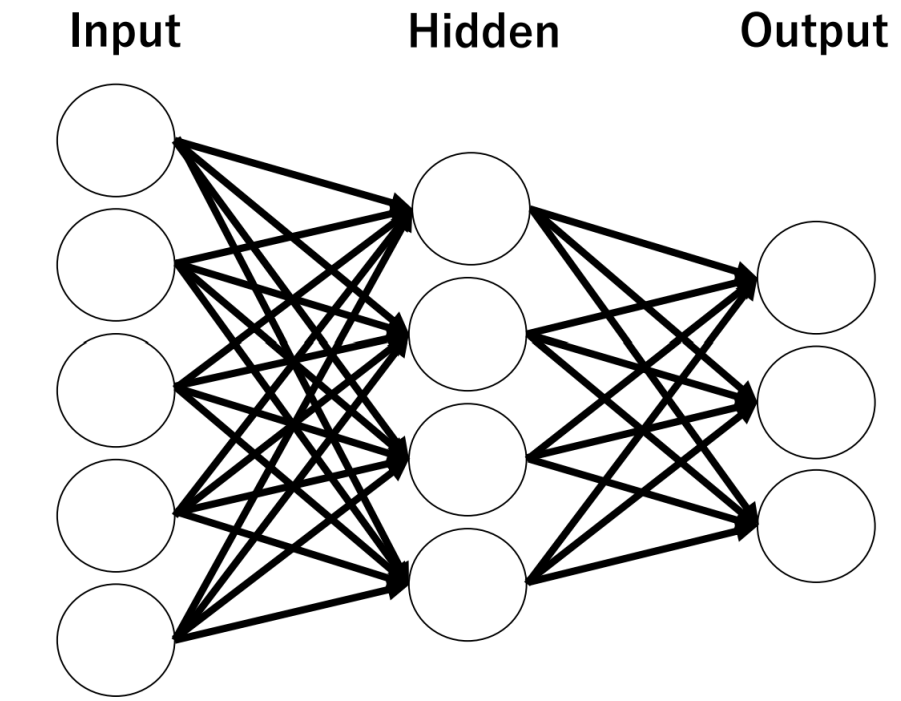


Figure 3.1: Network layer.

第4章　実験で使用した技術・手法

4.1　提案手法

4.1.1　背景

近年、多くの場面で非常に優れた性能を発揮する深層学習手法が使用され始めている。しかし深層学習モデルを実際に学習する場合、大量のラベル付きデータが必要となる。これが深層学習の実応用を制限している。そこでラベルを必要としない教師なし学習手法が注目されている。ここで教師なし学習はデータのみから特徴量を上手く得ることが重要である。教師なし学習の代表的なものとしてK-means法などのクラスタリングが挙げられる。しかし従来の教師なし学習では、クラスタの特徴量が他のクラスタと似通ること、学習データのノイズに左右されることにより、クラスタが一つにまとまってしまう、本来あるべきクラスタが消失するという問題が指摘されている。上記に述べた問題点を解消する手法として情報量最大化クラスタリング（IIC）が2018年に提案された。半教師あり学習にIICを適応した場合、教師あり学習の精度を超えるという結果も報告されている。[8)]

4.1.2　情報量最大化クラスタリング（IIC）

情報量最大化クラスタリング（IIC）は、正解ラベルを必要としない教師なし学習手法である。IICは主に画像分類の分野で成果を上げており、ある元画像に一般的なランダム変換を加えたペアとなる画像を作成し、このペアの相互情報量を最大化するようにネットワークを学習させる。IICは汎用的な手法であり、データの相互情報量が計算できれば、様々なタスクに使用することができる。

IICの大きな特徴は、相互情報量の最大化とオーバークラスタリングの二つである。[8)]

4.1.3　相互情報量

相互情報量とは2つの確率変数の相互依存の尺度である。数字が大きければ大きいほど、2つの情報が影響を及ぼしているため、1つの情報を見ただけでもう1つの情報を判別しやすい。元の画像とノイズを加えた画像は、同じオブジェクトを含む異なる画像である。元の画像と、ペアとなる画像の共通点を探すことで、オブジェクトの表現できる特徴量が得られる。

ここで相互情報量は以下に示す式で定義することができる。

$$I(X;Y) = \sum_{x \in D_X} \sum_{y \in D_Y} P_{X,Y}(x,y) \log \frac{P_{X,Y}(x,y)}{P_X(x) P_Y(y)} \tag{4.1}$$

ここで $P_{X,Y}$ は同時分布確率、$P_X(x)$ と $P_Y(y)$ はそれぞれ X と Y 周辺確率分布関数である。

本研究では、ニューラルネットワークに単語ベクトルを入力して出てくる出力ベクトルと単語ベクトルをランダムに少し変換したベクトルを入力して出てくる出力ベクトルの相互情報量を計算して、それが最大となるように学習を行う。[9)]

4.1.4 オーバークラスタリング

オーバークラスタリングは学習時のみに全結合層の最終層の数を増やす手法のことである。テスト時には無視されるが、正解クラスタ数よりもクラスタ数が多くなるように学習することで、ノイズの影響と未知のクラスタに対処できるようにする。

4.1.5 COS 類似度

COS 類似度とは、2 つのベクトルの類似性を表す指標である。ベクトル間の COS 値を求めることで、ベクトル同士がどの程度同じ方向を向いているかが求められる。このため、三角関数で用いるコサインの通り、1 に近ければ類似しており、0 に近ければ類似していないということになる。本研究では、ある二つの単語ベクトルのなす角度の近さ、すなわち単語の類似度を求めるために使用した。

2 つのベクトルを $\vec{a}$ と $\vec{b}$ とすると、COS 類似度をもとにした距離は以下の式で導出される。[3)]

$$\cos(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|} \tag{4.2}$$

4.1.6 WordCloud による単語集合の表現

WordCloud とは、文章中の単語の出現頻度に応じて単語を視覚的に図示する手法である。本来この手法は、文章中の単語の出現回数が多い程、その文字が強調されて表現されるため、その文章内の単語構成を視覚的に表現できる手法である。また大きさや色彩、角度などで単語を強調するため、文章を構成する単語が理解しやすい。WordCloud の例

を Figure 4.1 に示す。[10]

Figure 4.1: WordCloud.

WordCloud を作る際、generate 関数にある文章を渡すと、その中のある単語の出現回数でサイズを決める。しかし実際には、出現回数ではなく、他の別の数値で文字のサイズを調整したい場合がある。例として、TF-IDF で重みを付けた値を使いたい場合や、トピックモデルのトピック別出現確率のようなもともと割合で与えられたデータを使いたい場合などである。このような場合、"単語:頻度"の辞書（dict）を作成し、generate_from_frequencies 関数（もしくは fit_words）に渡すと実行することができる。本研究では、情報量最大化クラスタリングによって指定したクラスタ数へ単語を分類している。このクラスタリングの際にそのクラスタになる確率（推定確率）を単語頻度として使用し、" 単語：推定確率 " という辞書を作成することによって、分類された単語が、同一クラスタ内の他の単語とどれほど類似しているかを表す指標としている。

また比較対象として、階層的クラスタ分析によって得たクラスタでも WordCloud を作成している。この場合は、単語ベクトルとその単語が属するクラスタの中心座標の COS 類似度で数値化し、" 単語：COS 類似度 " という辞書を作成し、より類似している単語を強調している。これらによってクラスタの生成結果を可視化し、検証している。

4.2 Python

4.2.1 Python とは

Python はグイド・ヴァン・ロッサムによって創られたインタープリタ型の高水準汎用プログラミング言語である。動的な型付けやガベージコレクションなどの機能を持ち、手続き型、オブジェクト指向型、関数型プログラミングなどの幅広いプログラミングパラダイムをサポートしている。読みやすく、それでいて効率もよいコードをなるべく簡単に書けるようにするという思想が浸透しており、その単純さから初心者や非技術者に利用される。

4.2.2 Google Colaboratory

Google Colaboratory とは Google が提供するサービスであり、ブラウザから Python を実行できる。機械学習で必要な外部ライブラリ（NumPy など）もインストール済みであるため、簡単に実行環境を構築できる。また環境構築なしで、GPU を使用することができる。GPU とは Graphics Processing Unit の略称である。GPU は大量の演算を並列かつ高速に処理することができる性質を持っている。

4.2.3 MeCab

本研究では、Python から MeCab を呼ぶ出すことで形態素解析を行った。MeCab の辞書には、標準の IPA を導入した。

4.2.4 PyTorch

PyTorch は Facebook 社が開発した Python 向けのオープンソース機械学習ライブラリである。PyTorch は可読性の高さやデバックのしやすさで近年大きく人気を伸ばしているライブラリである。PyTorch は NumPy に近い操作性を持ち、「Define-by-Run」型の動的な計算グラフで設計されているため、柔軟性が高く、複雑なネットワークであっても比較的容易に実装することが可能である。また GPU を使用できるため、大規模なシステムやデータセット、複雑なモデル構築にも対応することができる。本研究では、GPU での計算、多次元配列を扱うためのデータ構造である Tensor 型への変換、ニューラルネットワーク構築の際のデータ構造やレイヤーを定義する活性化関数や損失関数の定義、パラメータ最適化アルゴリズムの実装などに使用した。[11)]

4.2.5 Gensim

Gensim は自然言語処理に用いられる様々なトピックモデルを実装した Python のオープンソースライブラリである。主に潜在意味解析のようなトピックモデルを扱いやすく、他に Word2vec のような Word embedding 手法を扱うこともできる。本研究ではトピックモデルを扱う用途ではなく、Word2vec を実装するために使用した。

4.2.6 SciPy

Scipy は Python のための数値解析ソフトウェアであり、Numpy に基づいた機能を有している。今回の実験においてクラスタ分析の過程で使用する階層的クラスタリングはこの SciPy の ward 法の linkage を用いて行う。

4.3 単語のベクトル化

4.3.1 分散表現

分散表現とは単語を高次元のベクトルとして表す技術である。コンピュータが演算できるのは数値のみであるため、単語の意味という概念的な要素を数値に置換することで、意味概念を計算することが可能になる。分散表現を効率的に活用するためのツールとして Word2vec が挙げられる。

4.3.2 Word2vec

Word2vec とは、ニューラルネットワークの重み学習を利用した単語の意味をベクトル表現化する手法である。2013 年に Google のトマス・ミコロフ氏らによって開発・公開された。Word2vec を利用して単語をベクトル化すると、次のような計算ができる。

- 単語同士の類似度計算
- 単語同士の加算・減算

具体例について以下の式を用いて説明する。Word2vec によって生成されたベクトル空間上には「king」、「man」、「queen」、「woman」という単語が存在するとする。これらの単語はベクトルとして実際に値を持っているため、単語同士で以下のような計算をすることができる。

$$「\text{king}」 - 「\text{mam}」 + 「\text{woman}」 = 「\text{queen}」 \tag{4.3}$$

式 (4.3) は空間上にある単語の足し引きによって導出されているため、ほかにも近い意味

のものが存在すればいくつか導出することも可能となる。こうした操作は、Word2vec による学習を済ませたモデルを用いることで、実際に行うことができる。[3)]

4.3.3 Word2vec による単語のベクトル化

Word2vec では、学習済みモデルに対して単語を指定することで、指定した単語のベクトル表現を取得する事が出来る。これにより、単語がベクトル空間上のどこの位置しているのかを数値として受け取ることが可能となる。数値型であるため、COS 類似度を用いた単語間の類似度計算や、単語間の距離をもとにしたクラスタ分析が可能となる。

4.4 階層的クラスタリング

4.4.1 クラスタ分析

クラスタ分析とは、異なるものが混ざり合う集団から、似た性質を持つクラスタという集団に分類する手法のことである。教師なしの分類手法であり、データの傾向をつかむことができる。形や色などで分類する場合とは違い、クラスタ分析は分類の基準や評価があらかじめ決められていない。

クラスタ分析には分類が階層的になる階層的クラスタ分析と、あらかじめクラスタ数を指定して分類する非階層的クラスタ分析がある。

4.4.2 階層的クラスタ分析

階層的クラスタリングは似ている対象から順にクラスタに分類していく手法である。対象となる 2 つの距離を計算し、その距離が近い者同士から順にクラスタを形成していく。階層的クラスタ分析は、クラスタ間の距離測定の方法にいくつか種類があり、対象データに最も適したものを選択する。本研究では ward 法を採用した。

4.4.3 ward 法

ward 法は 2 つのクラスタを融合した際に、同クラスタ内の分散と他クラスタ間の分散の比を最大化するようにクラスタを形成していく方法である。

4.4.4　非階層的クラスタ分析

非階層的クラスタリングは集団全体から、似た対象が同じクラスタに集まるよう分割する手法である。あらかじめいくつのクラスタに分類するかきめておく必要があるが、最適なクラスタ数を自動的に計算する手法が存在しないため、分析者によって大きく結果が変わることがある。

第5章　実験

5.1　実験の概要

本実験では、livedoor ニュースの記事を対象として、以下の二種類のクラスタリング手法で、単語間の類似度計算を行う。

- Word2vec ＋情報量最大化クラスタリング
- Word2vec ＋階層的クラスタリング（ward 法）

２つ目の手法は従来の研究で用いられている次元削減手法で、ward 法を用いた階層的クラスタリングによる方法になっている。この手法と提案手法である情報量最大化クラスタリングによる結果を比較することで、提案手法の優位性を検証する。

実験では、まず情報量最大化クラスタリングを行う。オーバークラスタリング率と全結合層のユニット数、クラスタ数を変化させ、情報量最大化クラスタリングの精度を高め、有用性を検証する。また扱った文書データの違いよるクラスタリング結果も検証する。実験の際に変化させたものによって以下のパターンに分けることができる。

- 実験パターン１　オーバークラスタリング率による変化
- 実験パターン２　出力層による変化
- 実験パターン３　オーバークラスタリング率と出力層による変化
- 実験パターン４　クラスタ数による変化
- 実験パターン５　文書データによる変化

クラスタ数やオーバークラスタリング率、出力層を変化させ結果を出力した。これは手法の精度にどのような影響が及ぼされるか検証し、より精度の高いクラスタリングを行うためである。

実験の文書データは、livedoor ニュースに現在掲載されている記事と livedoor ニュースコーパスから取得した。文書データを２つから取得し、比較することで、文書による結果の大きな相違がない事を検証するためである。 実験では、分析対象であるニュースのジャンル数、記事数を変化させ、結果を出力した。これは、ジャンル数や作品数といったテキストデータの情報量に変化をつけることで結果にどのような影響が及ぼされるか確認するためである。扱った文書データは以下に示すとおりである。

- 文書データ 1

  livedoor ニュースに現在掲載されている記事のうち 5 つのジャンルから 2 つずつ

- 文書データ 2

  livedoor ニュースコーパスの 7 つのジャンルからニュースを 1 つずつ

- 文書データ 3

  livedoor ニュースコーパスの 7 つのジャンルからニュースを 2 つずつ

次に階層的クラスタリングである ward 法での単語集合のクラスタ分析を行い、各クラスタ内の単語の類似性を検証する。この際、指定するクラスタ数、使用する文書データを変化させて、結果を比較する。全結果を踏まえて、ward 法に対する情報量最大化クラスタリングの優位性を検証する。本実験では、行った実験パターンが非常に多いため、結果の比較が行いやすいデータに注目して検証する。

5.2 実験準備

5.2.1 実験環境の構築

本実験の環境を構築するために、以下に示す項目を行った。

- Python,MeCab の導入
- Word2vec の学習済みモデルの取得
- ニュース記事の収集
- livedoor ニュースコーパスからの文書データの取得

5.2.2 MeCab の導入

MeCab を Python で実装するため、MeCab-python3 を使用した。MeCab-python3 は Python で簡単に MeCab を利用できるラッパーである。

5.2.3 Word2vec の学習済みモデルの取得

Word2vec を Python で実装するためにはモデルの学習、または学習済みモデルが必要である。しかし学習には膨大な時間を必要とする。そのため、本実験では配布されている学習済みモデルを取得し、使用することとした。取得した Word2vec の学習済みモデルは東北大学の乾、岡崎研究室にて作られた「日本語 Wikipedia エンティティベクトル」というモデルである。このモデルは、人名や地名といった固有表現の情報も含めた上でモデルを作成するために、日本語 Wikipedia の全記事本文から学習が行われている。本実験では、2017 年に学習された 200 次元のモデルを gensim で読み込んで使用した。

5.2.4　テキストデータ取得

今回文章分類の対象となるデータは、livedoor ニュースに現在掲載されている記事と livedoor ニュースコーパスの 2 つから取得した。

一つ目は株式会社ライブドアが提供する liveddor ニュースで現在掲載されている文書であり、以下の 5 つのジャンルから複数のニュースを任意に集めた。

- 野球
- 食べ物
- 産業
- 教育
- 気象

二つ目は livedoor のニュースとして以前掲載されていた文書であり、株式会社ロンウィットが収集し配布したデータである。この文書データは可能な限り HTML タグを取り除いて作成したものであり、9 ジャンルに分類されている。そのうち、以下に示す 7 ジャンルからニュースを複数個取得し、テキストデータとしている。

- dokujo-tsushin
- it-life-hack
- kaden-channel
- livedoor-homme
- movie-enter
- smax
- sports-watch

今回の実験の目的は似ている単語集合のクラスタを生成することであるため、対象となる文書データにジャンルが存在すると類似度の高い単語が多く含まれていることで、クラスタリングが容易になり、評価しやすいと考え、これらの文書データを用いる。

5.3　データの前処理

5.3.1　テキストデータの形態素解析

取得・収集した文章に対して、MeCab を使用することで形態素解析を行った。形態素解析後は必要な品詞のみを残す作業を行うよう設定した。本実験では、いくつかのニュースから単語を大量に抽出し、その単語同士の類似度を調べたいため、動詞や形容詞など

を含めると、結果が評価しずらくなると考え、名詞のみを抽出するように設定し、それ以外の単語は削除した。

またMeCabでは数値も名詞として抽出されるが、本研究の特性上必要ないため、削除した。さらに文章中には重複する単語が多く存在する。重複する単語は同一ベクトルであるため削除した。

5.3.2 Word2vecによる単語のベクトル化

単語ベクトル行列を生成するために、第5.3.1節で取得した単語の配列をWord2vecを用いてベクトルに変換する。GensimでWord2vecのモデルを作成して、get_vectorの引数に単語を入力することで対応したベクトル配列を取得する。本実験では200次元のモデルを使用するため、200次元配列を取得する。この配列を列方向に結合していき、単語数× 200の配列を作成する。

Word2vecで変換する際に、入力した単語のベクトルが存在しないことがある。そのため本実験では単語ベクトルが存在する単語のみベクトル化を行った。

これらの作業を行った結果、今回実験に使用した各文書データに含まれる単語の種類は以下の数となることが分かった。

- 文書データ1（単語数）… 455語
- 文書データ2（単語数）… 698語
- 文書データ3（単語数）… 1164語

5.3.3 データセット生成

前項で生成した配列をもとに、学習用の自作のデータセットを作る。

まずデータセットをテンソル化する前処理を定義する。前項で作成された単語ベクトル配列はNumpyのndarray型であるため、torch.Tensor関数によりテンソル型へと変換するように記述する。

次にPyTorchのモジュールであるtorch.utils.data.Datasetを使用し、データをテンソル化してラベルと合わせて返すDatasetを定義する。親クラス「torch.utils.data.Dataset」を継承して子クラス「MyDataset」を定義する。__init__で引数の処理をしておき、__len__で引数の長さ、__getitem__でインデックス番号を指定した際に、対応するdataとlabelを返すように定義する。本研究では教師なし学習であるため、labelは必要ない。しかし

torch.utils.data.Dataset を使用するため、label は np.zeros() でどのデータに対しても 0 を設定している。

最後に学習のためにデータをバッチサイズに分割してイテレータを返す DataLoader を定義する。第 1 引数は先程取得した Dataset を渡し、第 2 引数「batch_size」に 1 回の訓練またはテスト時に一気に何個の data を使用するか、第 3 引数「shuffle」に data の参照の仕方をランダムにするかを指定する。本実験では、batch_size=128、shuffle=False を指定した。

5.4 情報量最大化クラスタリング

5.4.1 IIC モデル定義

IIC のモデル定義では第 4.1.3 節で示したオーバークラスタリングを使用する。最終出力層では分類したいクラスタ数のものと、オーバークラスタリングのものを使用し、損失関数もその両方の出力を計算して、総和を使用する。オーバークラスタリングするネットワークで微細な変化を捉えることで、通常の期待したいクラス分類の性能もアップさせたいためである。

実験では NUMBER_OF_CLUSTERS（クラスタ数）と OVER_CLUSTERING_RATE（オーバークラスタリング率）と OUTPUT_LINEAR（ユニット数）の、3 つの変数の値を変化させることによる結果を比較し、与えたデータに対して最も良い結果となる数値を検証した。モデルの構築には、PyTorch のメインパッケージである torch に定義されている nn パッケージを使用し、torch.nn.Module クラスを継承し、NetIIC クラスを作成した。クラス内では第 3.2 節で説明した層（レイヤー）の定義を__init__関数で行う。forward というメソッドで、引数としてデータ（x）を受け取り、出力層の値を出すまでのネットワーク（順伝播）を記述している。

PyTorch で全結合層を定義するためには、torch.nn.Linear を用いる。nn.Linear は入力データに線形変換を適用するクラスである。引数は（インプットされたユニット数、アウトプットするユニット数）とする。全結合層の役割は、隣接する 2 層間の全てのニューロンユニット間において、単純な「線形重みパラメータによる線形識別的な変換」である。今回全結合層は 2 層である。一層目の引数は（入力ベクトル数、ユニット数）として（200, OUTPUT_LINEAR）とした。入力ベクトル数は単語ベクトルの 200 次元ベクトルに対応させた。この OUTPUT_LINEAR は変数であり、実験の際に変化を与えることによってユ

ニット数を増加させる。活性化関数 (Activation) としてReLU関数を採用し、正の値はそのまま出力され、負の値は0となる線形変換を行う。ReLU関数は、torch.nn.functionsに含まれており、こちらは慣習的にFと読み込まれる。二層目の引数は（OUTPUT_LINEAR, NUMBER_OF_CLUSTERS）とした。NUMBER_OF_CLUSTERSはクラスタ数であり、実験の際に変化を与えることで、クラスタリング結果を比較する。ここでの活性化関数には第3.2節で例として挙げたsoftmax関数を用いて、入力値ベクトルの各要素を0.0〜1.0の範囲の確率値に相当する出力値ベクトルに変換する。この関数によってデータがどのクラスに属するか判断する。同様にオーバークラスタリング用に二層目の引数を（OUTPUT_LINEAR, NUMBER_OF_CLUSTERS × OVER_CLUSTERING_RATE）にしたものを定義する。OVER_CLUSTERING_RATEはオーバークラスタリング率であり、実験の際に変化を与えることによって全結合層の最終層の数を変化させている。[12)] 最終的に出力層では、クラスタ数のものと、クラスタ数にオーバークラスタリング率をかけたものの二つが出力される。

5.4.2 重み、損失関数、相互情報量

モデルの重みに初期値を設定する際はtorch.nn.initを使用した。torch.nn.init.normal_()は、平均 (mean) に1、標準偏差（std）に0.02の正規分布から初期化するように設定した。torch.nn.init.constant_()では定数（val）を0に設定し、初期化した。

予想データと正解データの出力の間にどのくらい誤差があるのかを評価する損失関数を定義する。作成した予測モデルの精度を評価する際に損失関数は使われ、値が小さければ小さいほど正確なモデルである。今回は（出力値、少し変えた単語ベクトル）のペアを入力として受け取り、出力が少し変えた単語ベクトルとどれだけ離れているのかを計算する。相互情報量を最大化したいが、損失にするために、マイナスをかけ算して、最小化問題に置き換える。また、相互情報量の計算に係数項を加え、よりクラスがバラつきやすいようにしている。最終的なlossはクラスタリングとOverclusteringの平均となり、この値で勾配計算を行う。

損失を最小限に抑えるための最適化関数を定義する。pytorchのoptimというモジュールには様々なパラメータの更新手法があり、簡単に誤差逆伝播法を行ないパラメータを更新していくことができる。今回は勾配を記憶する「Momentum」と、勾配の二乗を記憶する「Adagrad」の項により構成される、torch.optim.Adamを使用する。

5.4.3　ペアの生成

この手法は教師なし学習であるため、教師ラベルは使用しない。IIC における入力には、対象のデータとデータを適当に変換したもの 2 つを使用する。実験では、元のベクトルデータに、全ベクトルデータの標準偏差に基づくノイズを加えて、ペアのデータを生成した。ネットワークにはそれぞれを入力し、それぞれの出力を得る。

元のデータと、ペアのデータを入力した際のそれぞれの出力の第 4.1.3 節で示した相互情報量が最大となるように学習を行う。

5.4.4　学習

学習時には model.train() を実行し、ネットワークを学習モードにする。epoch とはデータを一通り使用する 1 試行のことを意味し、今回は 1000 を設定した。pytorch のスケジューラーは epoch ごとに学習率を更新する。学習率は、scheduler の CosineAnnealingWarmRestarts を用意し、変化させている。学習率を変化させ、小さい値から急激に大きくしたときに局所解から抜け出し、大域的な極小解へとパラメータを学習させやすくする工夫である。ミニバッチ学習のため、1epoch の間に少しずつデータを使用して学習を進め、全データを一通り使用したら 1epoch 終了となる。

5.4.5　テスト（分類）

テスト時には model.eval() を実行して、テストモードに切り替える。結果を把握しやすいように、ミニバッチサイズ 1 のテスト用の DataLoader を用意しなおして、1 つずつ推論し、結果を格納する。

5.4.6　類似度による単語表現

5.5　クラスタ分析

単語をベクトル化した行列に対してクラスタリングをし、単語を分類する。SciPy の linkage を用いた階層的クラスタリングを使用する。距離の測定はユークリッド距離 (euclidien) を使用した。クラスタリング方法には第 4.4.3 節で説明を行った ward 法を指定する。linkage 関数のパラメータ method を ward にすることで Ward 法でのクラスタリングとなる。また、fcluster の引数に linkage の結果とクラスタ数を入力することで、指定したクラスタ数に分類する。

5.6 WordCloud の作成

ワードクラウドの作成は以下の手順で行う。

1. 本研究手法で文書分類を行う。
2. 分類過程で生成されたクラスタを用いて、クラスタ数×単語数の行列を作成する。
3. 行列の各行を用いて、各クラスタの WordCloud を作成する。

行列の作成には、階層的クラスタリングではクラスタの中心座標と各単語間の COS 類似度を使用する。各単語の２００次元のベクトルを NumPy の mean 関数で平均化し、中心座標を求めた。ワードクラウドは本来単語の出現頻度を使うため、重要度が高いほど値が大きくなる COS 類似度を代用で使用する。情報量最大化クラスタリングでは各単語の各クラスタへ属する確率値が算出されるが、そのうち確率値が最も大きいクラスタに各単語を分類している。よってこの最大値を、各クラスタに分類した際のよりそのクラスタに類似している重要値として代用する。それぞれ“ 単語：COS 類似度 ”の辞書、“ 単語：推定確率 ”の辞書型変数を作る。

WordCloud の作成には、Python の WordCloud ライブラリ generate_from_frequencies を使用する。行列の列ベクトルに単語を結び付けた辞書型変数をこの関数の引数にすることで、各クラスタのワードクラウドを作成する。

第6章　実験結果と考察

6.1　実験結果

6.1.1　変数と評価方法

変化させた変数は、オーバークラスタリング率とユニット数、クラスタ数である。文書データ1において、60以上の変数の組み合わせによる結果を出力し、文書データ2,3を合わせて、約100通りの組み合わせについて実験を行った。これらのデータはWordCloudによって可視化し、各クラスタ内に分類されている単語によって結果を比較する。

6.1.2　オーバークラスタリング率による変化

実験パターン1について検証を行った。クラスタ数はジャンル数の5を基準とした。オーバークラスタリング率の変化による結果の相違を検証した。文書データ1では1〜750まで任意に変化させ、14個の値を設定した。

まずユニット数を400に固定し、オーバークラスタリング率を1, 10, 20, 25, 50, 100, 150, 200, 700と変化させた。ユニット数400、オーバークラスタリング率が10のWordCloudの結果をFigure 6.1(a)〜6.1(e)に示す。それぞれのWordCloudでのクラスタリング結果を見ると、Figure 6.1(a)は産業関連Figure 6.1(d)は国名、地名、学校関連の類似した意味の単語が強調されていることがわかる。しかし強調されていない単語には類似性の低い単語が多い。その他のクラスタでは全体的に類似性の高い単語が少ないことが分かる。

オーバークラスタリング率を100に変えた場合の結果をFigure 6.2(a)〜6.2(e)に示す。それぞれのワードクラウドを見ると、どれも類似した意味の単語が強調されていることがわかる。例えばFigure 6.2(a)では「二塁」「三振」「投手」といった野球関連の単語、Figure 6.2(b)では「気温」「気圧」「積雪」といった気象関連の単語といったように、それぞれのワードクラウドで特色を見ることができる。それぞれのワードクラウドの内容は下のようになる。

- 野球関係（投手、二塁、三振、投球、エース　など）
- 気象関係（気圧、気温、積雪、大雨、発達　など）
- 食べ物関係（醤油、野菜、レシピ、料理、炭酸　など）
- 教育関係（教育、指導、分野、育成、科学　など）
- 産業、経済関係（政府、生産、米中、対中、インテル　など）

(a) Cluster 0.

(b) Cluster 1.

(c) Cluster 2.

(d) Cluster 3.

(e) Cluster 4.

Figure 6.1: IIC,clusters:5,overclustering rate:10,unit number:400.

Figure 6.2(e) の産業、経済関係には「中頃」「舞台」「激動」などのそれほど類似性の高くない単語も強調されているが、強調されていない単語に「輸出」「倒産」「売り上げ」等があるため、このクラスタは産業や経済関連のクラスタであると言える。

オーバークラスタリング率 10、ユニット数 400 の際の Figure 6.1(a)〜 6.1(e) と比較すると、全体的に類似性の高い単語が同一クラスタ内に多く分類されており、5 つジャンルにまとまっているといえる。

オーバークラスタリング率を 200 に変えた場合の結果を Figure 6.3(a)〜6.3(e) に示す。オーバークラスタリング率 100、ユニット数 400 の際の Figure 6.2(a)〜 6.2(e) と比較すると、野球関連のクラスタは単語は異なるが関連度に大きな変化はない。また気象関連の単語クラスタは地名関連のクラスタに変わり、食べ物関連のクラスタには「商船」「インテル」「デンソー」といった産業関連の単語がみられるようになった。また教育と産業

(a) Cluster 0.

(b) Cluster 2.

(c) Cluster 2.

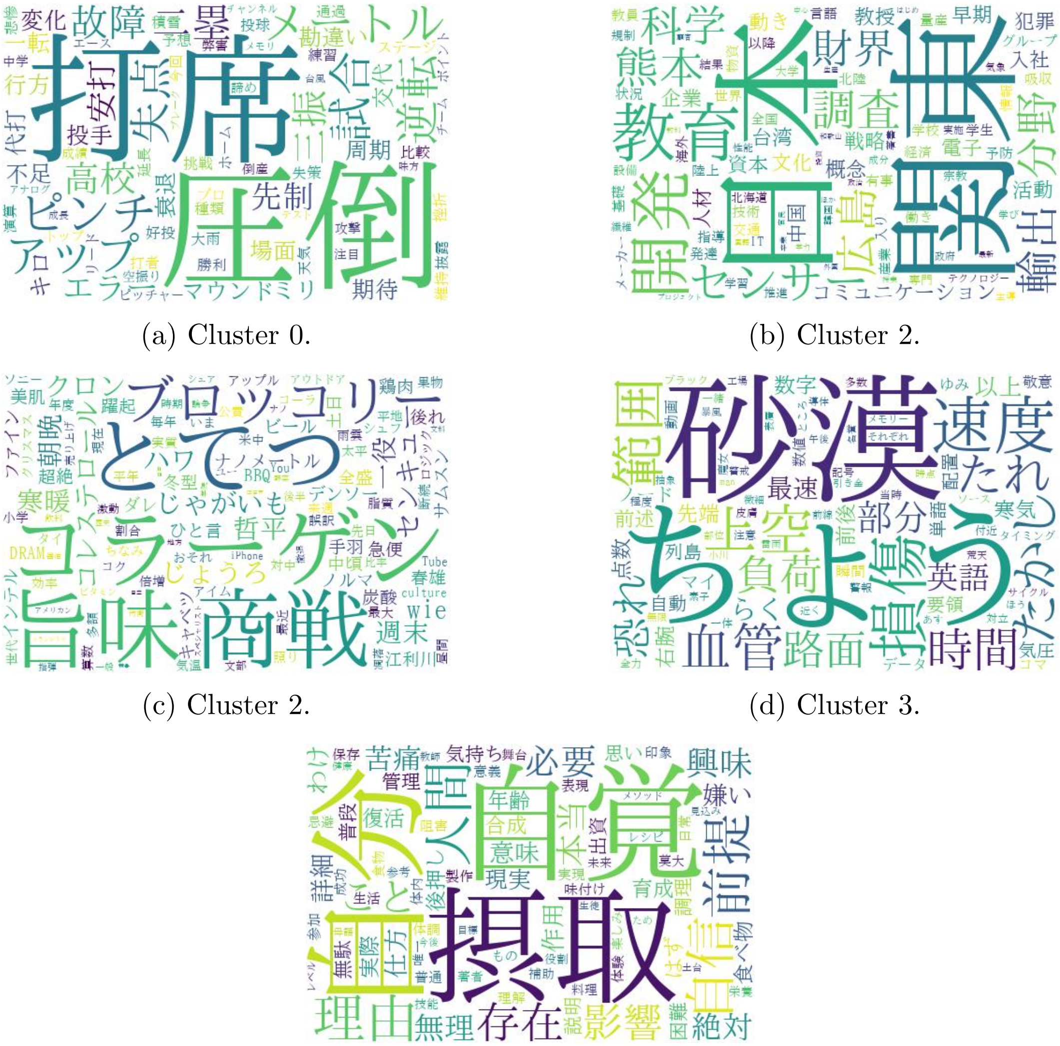
(d) Cluster 3.

(e) Cluster 4.

Figure 6.2: IIC,clusters:5,overclustering rate:100,unit number:400.

関連のクラスタは無くなり、「砂漠」「ちょう」「自覚」「摂取」といった関連性の低い単語が強調され、クラスタ内の他の単語の類似性は低くなった。

オーバークラスタリング率を 700 に変えた場合の結果を Figure 6.4(a)〜6.4(e) に示す。それぞれの WordCloud を見ると、Figure 6.4(b) では「勝利」「代打」「失点」「トップ」など野球関連の単語が多い。Figure 6.4(d) では「ナノ」「ミリ」「アナログ」など生産関連の単語が多く、全体的にカタカナが多いことが分かる。しかし強調されていない単語の多くは、これらとは関係のない単語ばかりである。またその他のクラスタでは、「企業」と「顧客」や、「苦痛」と「嫌い」など関連している単語もあるが、全体的に類似性の低い単語が多く、1 つのクラスタ内に複数のジャンルの単語が集まっていることが分かる。オーバークラスタリング率 200、ユニット数 400 の際の Figure 6.3(a)〜 6.3(e) と比較すると、さらに単語が分散しており、食べ物や地域などに関連するクラスタも無くなって



(a) Cluster 0.

(b) Cluster 2.

(c) Cluster 2.

(d) Cluster 3.

(e) Cluster 4.

Figure 6.3: IIC,clusters:5,overclustering rate:200,unit number:400.

いる。オーバークラスタリング率 100、ユニット数 400 の際の Figure 6.2(a)〜 6.2(e) と比較すると、食べ物や気象、教育などに関連するクラスタは見受けられず、類似性の高い単語を含め、全体的に単語が分散してしまっていることが分かる。

次にユニット数を 200 に固定し、オーバークラスタリング率を 1〜750 まで任意に変化させた。

ユニット数 200、オーバークラスタリング率が 10 の WordCloud の結果は、5 個のクラスタのうち類似性の高い単語が少なく単語の分散が激しいクラスタは省略する。そのうち関連性を判断することが出来た 2 つのクラスタを Figure 6.5(a),6.5(b) に示す。 それぞれの WordCloud でのクラスタリング結果を見ると、Figure 6.5(a) は「ポイント」「三振」「投手」「打席」など野球関連の単語が多い。Figure 6.5(b) は「倒産」「撤退」といった産業関連の単語が多い。しかし強調されていない単語には類似性の低い単語が多い。

(a) Cluster 0.

(b) Cluster 1.

(c) Cluster 2.

(d) Cluster 3.

(e) Cluster 4.

Figure 6.4: IIC,clusters:5,overclustering rate:700,unit number:400.

(a) Cluster 0.

(b) Cluster 2.

Figure 6.5: IIC,clusters:5,overclustering rate:10,unit number:200.

(a) Cluster 0.

(b) Cluster 1.

(c) Cluster 2.

(d) Cluster 3.

(e) Cluster 4.

Figure 6.6: IIC,clusters:5,overclustering rate:100,unit number:200.

オーバークラスタリング率を 100 に変えた場合の結果を Figure 6.6(a)〜6.6(e) に示す。Figure 6.6(a) は「鶏肉」「ダレ」「コーラ」といった食べ物関連の単語が多く強調されていることが分かる。Figure 6.6(b) は「経理」「企業」「戦略」など経済関連の単語が多いが、「説明」「未来」といった単語が強調されている。Figure 6.6(c) は「三振」「基礎」「好投」「全力」など野球関連の単語が多いが、「多額」「全盛」といった類似度の低い単語も分類されている。Figure 6.6(d) は学校、教育関連の単語が多く、強調されていることがわかる。Figure 6.6(e) は「表層」「気温」「気圧」といった気象関連の単語が多い。また「成分」「素子」「物資」といった単語が強調され、「ミリ」「キロ」「メートル」といった単位も分類されていることが分かる。おおよそどのクラスタにも、類似性の高い単語が一定数集まっているため、オーバークラスタリング率が 10 の場合よりクラスタリング精度は向上したことが分かる。

(a) Cluster 0.

(b) Cluster 1.

(c) Cluster 2.

(d) Cluster 3.

(e) Cluster 4.

Figure 6.7: IIC,clusters:5,overclustering rate:400,unit number:200.

オーバークラスタリング率を 400 に変えた場合の結果を Figure 6.7(a)〜6.7(e) に示す。それぞれの WordCloud を見ると、Figure 6.7(e) では「投球」が強調され、「打席」や「ピッチャー」といった単語が分類されているが、そのほかに強調されている単語やクラスタに属している単語の類似性は非常に低いことが分かる。これまでのクラスタリング結果と比較しても明らかに野球関連の単語が強調されていないため、関連する単語がまとまりにくいことが分かる。Figure 6.7(a) はカタカナや英語、Figure 6.7(b) は食べ物や場所、Figure 6.7(c) は感情や PC 関連、Figure 6.7(d) は地名といった単語が現われてはいるが、全体的に各クラスタで類似性の高い単語が少ないことが分かる。他のクラスタにも食べ物や場所、地名に関連する単語は分類されているため、関連する単語がまとまりにくいことが分かる。また Figure 6.7(c) で「期待」と「無駄」が同程度強調されていることが分かる。

(a) Cluster 0.

(b) Cluster 1.

(c) Cluster 2.

(d) Cluster 3.

(e) Cluster 4.

Figure 6.8: IIC,clusters:5,overclustering rate:750,unit number:200.

オーバークラスタリング率を 750 に変えた場合の結果を Figure 6.8(a)〜6.8(e) に示す。Figure 6.8(e) をみると野球関連の単語が多く、オーバークラスタリング率が 400 の時より、類似性の高い単語が多く分類されていることが分かる。しかしこれまでの結果とは異なり、強調されている単語やその他の単語にカタカナが多くなっていることが分かる。反対にその他のクラスタでは強調されている単語に漢字が多く、全体的に各クラスタでの単語の類似性が低いことが分かる。また Figure 6.8(c) は「参加」「吸収」「卒業」の単語がほぼ同じ大きさで強調されていることが分かる。Figure 6.8(e) についてはオーバークラスタリング率が 400 の時より関連する単語はまとまったが、全体的なクラスタリング結果としては関連する単語がまとまりにくいことが分かる。

この他にもユニット数を 100 に固定した場合や、ユニット数を極端に小さく設定した場合なども検証したが、オーバークラスタリング率を減らすと、単語が元のジャンルに関

係無くばらばらに分散し、強調されている単語の類似性も低い結果となった。またオーバークラスタリング率を増やしすぎると、一見類似性の高い単語が分類されたように見えるが、同一クラスタ内で複数のジャンルの単語がいくつも集まり、クラスタ内に複数の集合が出来てしまった。よってユニット数が 400 でオーバークラスタリング率が 100 の IIC と、ユニット数が 200 でオーバークラスタリング率が 100 の IIC の結果が最も関連する単語がまとまりやすいことが分かった。これらの比較は次項で行う。

6.1.3　ユニット数による変化

クラスタ数は前項と同じくジャンル数の 5 を基準とした。実験パターン 2 について、文書データ 1 でのユニット数は 5〜4000 まで任意に変化させ、15 の値を設定した。第 6.1.2 節での結果より、精度の高かったオーバークラスタリング率 100 について、ユニット数を変化させた場合について比較する。まずすでに結果が示されている、ユニット数が 400 の場合の結果 Figure 6.2(a)〜6.2(e) と、ユニット数が 200 の場合の結果 Figure 6.6(a)〜6.6(e) を比較する。ユニット数が 400 の時、各 WordCloud の内容は以下に示す通りであった。

- 野球関係（投手、二塁、三振、投球、エース　など）
- 気象関係（気圧、気温、積雪、大雨、発達　など）
- 食べ物関係（醤油、野菜、レシピ、料理、炭酸　など）
- 教育関係（教育、指導、分野、育成、科学　など）
- 産業、経済関係（政府、生産、米中、対中、インテル　など）

全体的に類似性の高い単語が同一クラスタ内に多く分類されており、強調されている単語の類似性もかなり高く、おおよそ 5 つジャンルにまとまっているといえた。ユニット数が 200 の場合、各 WordCloud の内容は以下に示す通りであった。

- 野球関係（三振、基礎、好投、全力　など）
- 気象と単位、物質関係（表層、気温、気圧、成分、素子、物資、ミリ、キロ、ナノ　など）
- 食べ物関係（鶏肉、ダレ、ちなみ、コーラ、じゃがいも　など）
- 教育関係（教師、授業、学校、卒業、小学、成績　など）
- 経済関係（経理、企業、戦略、状況、競争　など）

経済関連の単語クラスタでは、「説明」「未来」といった単語が強調され、野球関連の単

(a) Cluster 0.

(b) Cluster 1.

(c) Cluster 2.

(d) Cluster 3.

(e) Cluster 4.

Figure 6.9: IIC,clusters:5,overclustering rate:100,unit number:5.

語クラスタでは、「多額」「全盛」といった類似度の低い単語も分類されていた。おおよそどのクラスタにも、類似性の高い単語が一定数集まっているが、ユニット数が 400 の場合より関連する単語がまとまりづらいことが分かる。

ユニット数を 5 とした場合の WordCloud を Figure 6.9(a)〜6.9(e) に示す。Figure 6.9(a) より、「設備」「戦力」「各社」といった単語が強調され、会社に関連する単語が、Figure 6.9(b) より「打席」「ベンチ」などが強調され、野球関連の単語が多いことが分かる。Figure 6.9(c) では「通算」「記録」「得点」「リーグ」などが強調され、スポーツ関連の単語が多いことが分かる。Figure 6.9(d) では「輸出」「小型」「助手」などが強調され、産業関連の単語が多い。いくつかの類似度の高い単語がまとまって他のクラスタ内に分類されており、会社は産業の一部、野球はスポーツの一種とすると、それぞれ 2 つのクラスタに類似性の高い単語が分散していることが分かる。また Figure 6.9(e) では「公費」

(a) Cluster 0.

(b) Cluster 1.

(c) Cluster 2.

(d) Cluster 3.

(e) Cluster 4.

Figure 6.10: IIC,clusters:5,overclustering rate:100,unit number:1000.

「鉄鋼」「地方」などが強調され、クラスタ内の単語の類似性が低いことが分かる。

ユニット数を 1000 にした場合の WordCloud を Figure 6.10(a)〜6.10(e) に示す。 Figure 6.10(b) には気象と料理関連の単語が、Figure 6.10(c) には産業と経済関連の単語が、Figure 6.10(e) には野球関連の単語が強調されていることが分かる。強調されていない単語にも注目すると、これらに関連する類似度の高い単語も多いが、全く類似性のない単語も多い。各クラスタに分類されている単語全体で、類似度の高い単語が多いわけではないことがわかる。その他の 2 つのクラスタは類似性の低い単語が多く、強調されている単語の類似度も低い。またカタカナと英語が Figure 6.10(a) に多いことが分かる。

ユニット数を 4000 にした場合の WordCloud を Figure 6.11(a)〜6.11(e) に示す。 Figure 6.11(e) には教育関連の単語が、Figure 6.11(d) の一部には野球関連の単語が多く、強調されていることが分かる。その他の 3 つのクラスタは類似性の低い単語が多く、強調さ

(a) Cluster 0.

(b) Cluster 1.

(c) Cluster 2.

(d) Cluster 3.

(e) Cluster 4.

Figure 6.11: IIC,clusters:5,overclustering rate:100,unit number:4000.

れている単語の類似度も低く、関連のない単語があちこちに分散していることがわかる。またカタカナと英語が Figure 6.11(b) に多いことが分かる。ユニット数が 5,1000,4000 の場合の結果をユニット数が 400,200 の場合と比較すると、5 つのクラスタ全てにおいて類似度の高い単語を分類できておらず、全体的にクラスタ内の単語群の類似性が低いことが分かる。よってユニット数による検証ではユニット数が 400 の場合が最も関連する単語がまとまりやすかった。

またユニット数の増減による結果の変化は、ユニット数が少なすぎると、類似性のより高い単語が 2 つ以上のクラスタに分散してしまうことが分かった。ユニット数が多すぎると、適切な値の場合で同一クラスタ内だった類似度の高い単語が、別のクラスタ内に現れ、まとまっていき、一つのクラスタ内にいくつかのまとまりが形成されていく。さらにユニット数を増加させると、関連する単語がまとまりにくくなることが分かった。

またユニット数を増加させると、カタカナと英語がある 1 つのクラスタに偏って分類され、そのほかのクラスタで強調されている単語が、徐々に漢字ばかりになっていくことが分かった。

6.1.4　オーバークラスタリング率と出力層による変化

実験パターン 3 について、前項までで示した結果以外にも、実験パターン 1 について、変数値の組み合わせによって 50 以上の結果を得た。しかし、単語が分散してしまい特徴をつかめないクラスタや、全く関連のない単語ばかりが強調されるクラスタや、変数の変化による結果の変化がほとんどみられないものも多かった。特にユニット数の変化では、極端に増減しないと、大きな変化は得られなかった。よってユニット数が 400、オーバークラスタリング率が 100 の場合以上の結果を得ることは出来なかった。

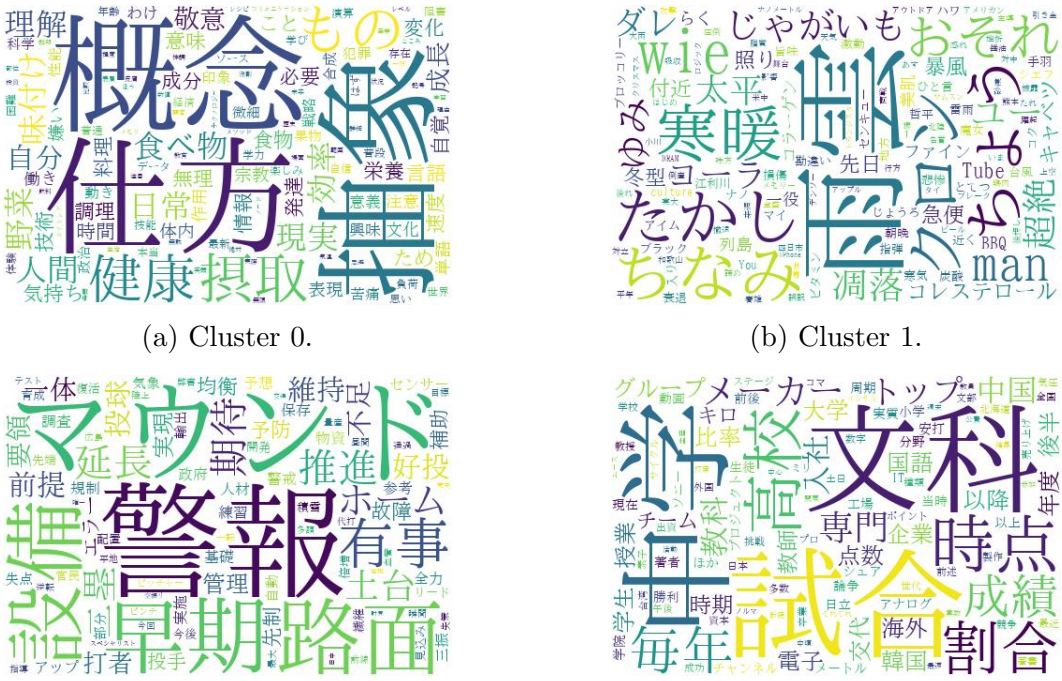
6.1.5　クラスタ数による変化

これまでの結果ではクラスタ数をジャンル数の 5 に設定しているが、ユニット数が 400、オーバークラスタリング率が 100 の IIC でクラスタ数を 1 増減させた場合の結果を検証する。

クラスタ数が 4 の場合を Figure 6.12(a)〜6.12(d) に示す。Figure 6.12(a)〜Figure 6.12(c) のどのクラスタを見ても、類似度の高い単語が少なく、関連性を特定できない。Figure 6.12(d) のみ教育関連の単語が多いことが分かる。クラスタ数を減らすことで関連する単語がまとまりづらくなったことが分かる。

クラスタ数が 6 の場合を Figure 6.13(a)〜6.13(f) に示す。 Figure 6.13(b) では「皮膚」「摂取」「発達」など人間自身に関連する単語、Figure 6.13(c) では食べ物関連の単語、Figure 6.13(d) では教育関連の単語、Figure 6.13(e) では「無理」「苦痛」「恐れ」など感情に関連の単語、Figure 6.13(f) では野球関連の単語が多いことが分かる。また Figure 6.13(a) には類似性の低い単語が多いことが分かる。また産業や気象に関連する単語が多く集まるクラスタが消え、人間自身や感情に関連する単語が多く集まることが分かった。また 1 つのクラスタには、類似性の低い単語が集まることが分かった。

オーバークラスタリング率 100、ユニット数 400 でクラスタ数が 5 の場合と比較すると、クラスタ数が 4 の場合は、関連する単語がまとまりづらくなった。クラスタ数が 6 の場合は、文書データ 1 での 5 つのジャンルには関連しないクラスタが生成されること



(a) Cluster 0.

(b) Cluster 1.

(c) Cluster 2.

(d) Cluster 3.

Figure 6.12: IIC,clusters:4,overclustering rate:100,unit number:400.

が分かる。またある一つのクラスタには類似度の低い単語ばかりが集まるが、そのほかのクラスタでの単語は類似性の比較的高い単語が集まる。よって関連のある単語がまとまりやすくなるが、その分関連のない単語が一つのクラスタにまとまることが分かる。

6.1.6 文書による変化

文書データ 2 でのオーバークラスタリング率は 1〜20、ユニット数は 50〜400 の範囲で任意に変化させた。

このうち最も精度の高い結果を得たオーバークラスタリング率 10、ユニット数 200 の場合の WordCloud を Figure 6.14(a)〜6.14(g) に示す。 Figure 6.14(a) には物に関連する単語、Figure 6.14(b) には英語、Figure 6.14(c) には人名や名称、Figure 6.14(d) には経済関連の単語、Figure 6.14(e) には文学に関連する単語、Figure 6.14(f) には PC に関連する単語が多いことが分かる。Figure 6.14(g) のみ学校とスポーツ、イベントといった複数のジャンルに関連する類似性の高い単語が集まっていることが分かる。おおよそどのクラスタにも、類似性の高い単語が一定数集まっていることが分かる。

文書データ 3 でのオーバークラスタリング率は 1〜30、ユニット数は 50〜600 の範囲で任意に変化させた。このうち最も精度の高い結果を得たオーバークラスタリング率 20、ユニット数 400 の場合の WordCloud を Figure 6.15(a)〜6.15(e) に示す。 Figure 6.15(a) には

(a) Cluster 0.

(b) Cluster 1.

(c) Cluster 2.

(d) Cluster 3.

(e) Cluster 4.

(f) Cluster 5.

Figure 6.13: IIC,clusters:6,overclustering rate:100,unit number:400.

文学、メディアに関連する単語、Figure 6.15(b) には PC に関連する単語、Figure 6.15(d) には人間のもつ感想、Figure 6.15(e) には英単語、Figure 6.15(f) には大きさや場所に関連した単語、Figure 6.15(g) には地名が多いことが分かる。Figure 6.15(c) は人間関係、「高価」「豪華」「本格」など人の評価といった複数のジャンルに関連する類似性の高い単語が集まっていることが分かる。また地名ではないカタカナが多い。文書データ 2 ほどではないが、どのクラスタにも、類似性の高い単語がそこそこ集まっていることが分かる。

6.1.7 ward 法による変化

文書データ 1,2 において、ward 法での階層的クラスタリングを行い、WordCloud を作成した。

まず文書データ 1 について、ジャンル数 5 に対してクラスタ数が同じ場合のワードク

(a) Cluster 0.

(b) Cluster 1.

(c) Cluster 2.

(d) Cluster 3.

(e) Cluster 4.

(f) Cluster 5.

(g) Cluster 6.

Figure 6.14: IIC,clusters:7,overclustering rate:10,unit number:200.

ラウドをそれぞれ Figure 6.16(a)〜6.16(e) に示す。 それぞれの WordCloud より、Figure 6.16(b) では「打席」や「代打」、「好投」など野球関連の単語が多い。Figure 6.16(c) では「量産」や「生産」、「衰退」など産業関連の単語といった特徴がみられるが、小さな単語を見るとその特徴とは異なる単語も多い。またその他のクラスタでは、一部の単語は類似しているが、全体的に類似性の低い単語が多く、一つのクラスタ内に複数のジャ

(a) Cluster 0.

(b) Cluster 1.

(c) Cluster 2.

(d) Cluster 3.

(e) Cluster 4.

(f) Cluster 5.

(g) Cluster 6.

Figure 6.15: IIC,clusters:7,overclustering:20,unitnumber:400.

ンルの単語が集まっているかのような印象を受ける。さらに Figure 6.16(b) の単語数が他のクラスタよりかなり少なくなっている。

次に指定するクラスタ数を 1 増やして、再度 WordCloud を作成した。文書データ 1 について、ジャンル数 5 に対してクラスタ数が 6 のワードクラウドをそれぞれ Figure 6.17(a)〜6.17(e) に示す。 それぞれの WordCloud を見ると、Figure 6.17(c) では「右腕」や「代

(a) Cluster 0.

(b) Cluster 1.

(c) Cluster 2.

(d) Cluster 3.

(e) Cluster 4.

Figure 6.16: Ward,clusters:5.

打」、「打席」など野球関連の単語が多い。Figure 6.17(d) では「量産」や「生産」、「衰退」など産業関連の単語といった特徴がみられる。これらは先ほどのクラスタ数が 5 であった時の Figure 6.16(b) や Figure 6.16(c) と同様の特徴である。クラスタ数が 5 の場合との違いは、Figure 6.17(a) に「設備」や「効率」、「素子」など部品や生産に関連する単語が多い点である。またクラスタ数が 5 の場合では、強調して表現されていない単語が多く集まっている。しかしその他のクラスタでは、一部の単語は類似しているが、全体的に類似性の低い単語が多く、強調されている単語の類似性も低い。また「もの」と「競争」、「あす」と「平年」と「おそれ」といった単語はどちらのクラスタリングの場合でも同一のクラスタに分類され、強調されていることが分かる。

次に文書データ 3 について、ジャンル数 7 に対してクラスタ数が同じ場合のワードクラウドを Figure 6.18(a)〜6.18(g) に示す。それぞれの WordCloud を見ると、Figure 6.18(e)

(a) Cluster 0.

(b) Cluster 1.

(c) Cluster 2.

(d) Cluster 3.

(e) Cluster 4.

(f) Cluster 5.

Figure 6.17: Ward,clusters:6.

では「選手」や「引退」などスポーツ関連の単語が多く、Figure 6.18(f) では「記事」や「全文」、「ラジオ」や「スタジオ」やなど文学、メディア関連の単語といった特徴がみられる。しかしその他のクラスタでは、一部の単語は類似しているが、全体的に類似性の低い単語が多く、強調されている単語の類似性も低い。

文書データ 3 について、ジャンル数 7 に対してクラスタ数が 8 のワードクラウドをそれぞれ Figure 6.19(a)〜6.19(h) に示す。 それぞれの WordCloud を見ると、Figure 6.19(g) では「記事」や「全文」、「ラジオ」や「スタジオ」やなど文学、メディア関連の単語といった特徴がみられる。しかしその他のクラスタでは、先ほどのクラスタ数が 7 の場合と同様に、一部の単語は類似しているが、全体的に類似性の低い単語が多く、強調されている単語の類似性も低い。クラスタ数が 7 の時、Figure 6.18(e) のようにスポーツ関連の単語が多かったが、その特徴を持つクラスタが無くなり、代わりに Figure 6.19(d) の

(a) Cluster 0.

(b) Cluster 1.

(c) Cluster 2.

(d) Cluster 3.

(e) Cluster 4.

(f) Cluster 5.

(g) Cluster 6.

Figure 6.18: Ward,clusters:7.

「宮崎」や「都内」、「北京」といった場所や地名に関連する単語が多いクラスタが現われた。

文書データ 1,3 に対する ward 法での階層的クラスタリングでは、全体的に各クラスタ内の単語の類似性が低く、類似性の高い単語が集まったクラスタは 2 つ程度であった。

(a) Cluster 0.

(b) Cluster 1.

(c) Cluster 2.

(d) Cluster 3.

(e) Cluster 4.

(f) Cluster 5.

(g) Cluster 6.

(h) Cluster 7.

Figure 6.19: Ward,clusters:8.

6.2 考察

6.2.1 オーバークラスタリング率とユニット数

実験で定義したニューラルネットワークは全結合層が 2 層で構成されている。ユニット数を変化させることで、全結合層の一層目のアウトプットするユニット数が変化する。正の値はそのまま出力され、負の値は 0 となる ReLU 関数を用いているが、ここでユニッ

ト数を増やすとより多くの特徴量を出力することになる。オーバークラスタリング率を変化させ、指定したクラスタ数と掛け算をすると、全結合層の 2 層目（最終層）のアウトプットするユニット数が変化する。二層目は活性化関数である Softmax 関数用いているため、データがどのクラスに属するか判断する際の、クラス数が増加する。つまり入力値ベクトルの各要素を 0.0〜1.0 の範囲の確率値に相当する出力値ベクトルに変換しているため、その出力ベクトルが増加していく。その後これらの値を出力層で指定したクラスタ数にグループ化していく。

オーバークラスタリング率を減らすと、単語が元のジャンルに関係無くばらばらに分散し、強調されている単語の類似性も低い結果となった。またオーバークラスタリング率を増やしすぎると、一見類似性の高い単語が分類されたように見えるが、同一クラスタ内で複数のジャンルの単語がいくつも集まり、クラスタ内に複数の集合が出来てしまった。よってクラスタ数が 5、ユニット数が 400 でオーバークラスタリング率が 100 の IIC と、ユニット数が 200 でオーバークラスタリング率が 100 の IIC の結果が最も関連する単語がまとまりやすいことが分かった。この 2 つではユニット数 400 の方がより、関連する単語がまとまりやすかった。文書データ 1 の単語数は 455 個であり、単語ベクトルの次元数は 200 次元であり、文書データ 1 では、次元数に対して 2 倍程度の場合が、関連する単語が集合すると考えられる。少なすぎると、指定したクラスタにまとめる際に類似度のかなり高い単語のみがクラスタになると考えられる。多すぎるとオーバークラスタリングをしすぎてしまい、重要な特徴を持たないクラスタが増えすぎてしまうと考えられる。全結合層の 2 層目での 0.0〜1.0 の範囲の確率値の数が多すぎると、確率値に大きな違いが生まれず、グループ化の際に類似度の高い単語同士をうまく分類できないと考えられる。

今回の検証では単語の意味によるクラスタリングを行いたいが、ユニット数を増やしすぎると、意味に基づく分類ではなくなってしまうと考えられる。これはユニットが多すぎることで、必要となる特徴以外のものにも着目してしまい、多くの特徴量から必要な特徴だけをうまく取り出せていないと考えられる。ユニット数が少なすぎると、取り出す特徴が少なすぎると考えられる。またユニット数が多い場合は、結果から単語の種類、英語、カタカナといった情報を取り出してしまっているとも考えられる。ユニット数は適度な値にすることが重要である。

ユニット数、オーバークラスタリング率を大きくしていくと、学習時間は増加していっ

た。レイヤの数、レイヤあたりのユニット数を増やすことで、更新する重みの数が増えるため学習には時間がかかることが分かった。

全体的に、「とてつ」「ちょう」「こと」「らく」といった意味のない単語も多く、英語では「You」「Tube」など一つの単語が二つ以上に分裂しているものも多かった。WordCloudをで可視化した際に、目視で類似度を判断できない。これらの単語は今回のWordCloudでの可視化の際に、影響を与えていると考えられる。

また今回、訓練データとは異なるデータでは、検証を行っていない。ニューラルネットワークはデータに対応し、目的を達するのに十分な精度を持っている必要がある。さらに、訓練データに対する精度が十分であっても、モデルが訓練されたデータとは異なる未知のデータに対しても同様に高い精度を達成することが重要であるため、未知のデータに対する精度が低ければモデルは実用的な価値がない。一般的なパターンや特徴を学習し、未知のデータにも適用可能な汎化能力が必要であると考えられる。

6.2.2 クラスタ数

オーバークラスタリング率100、ユニット数400でクラスタ数が5の場合と比較すると、クラスタ数が4の場合は、関連する単語がまとまりづらくなった。クラスタ数が6の場合は、文書データ1の5つのジャンルには関連しないクラスタが生成されることが分かる。またある一つのクラスタには類似度の低い単語ばかりが集まるが、そのほかのクラスタでの単語は類似性の比較的高い単語が集まる。よって関連のある単語がまとまりやすくなるが、その分関連のない単語が1つのクラスタに集まることが分かる。

クラスタ数を減らすと、オーバークラスタリングを行ったあとのグループ化の際に、関連の低いグループも同じクラスタにまとめてしまうと考えられる。クラスタ数を増やすと、ある一つのクラスタに他のどのクラスタとも類似度が低い単語が集まるため、そのほかの単語の類似性が高くなり、他のクラスタでは関連する単語がかなりまとまりやすくなると考えられる。1つのクラスタを使わず、元のジャンルを考慮しない場合は、オーバークラスタリング率、ユニット数が少ない場合や階層的クラスタリングと比較して、クラスタ数を1増やす場合の方が関連する単語がまとまりやすいと考えられる。元のジャンル数を考慮する場合は、クラスタ数がジャンルと同じ場合が関連する単語が集まりやすい。

6.2.3　文書による違い

文書データ 1 は livedoor ニュースに現在掲載されている記事から取得している。文書データ 2,3 は livedoor ニュースコーパスから取得している。取得しているデータ先が異なるため、文書データ 1 と文書データ 2,3 での単語数と、ユニット数、オーバークラスタリング率にはそれほど、相関がない。これは元の文書データ内に、どれほど類似度の高い単語が含まれているかによって結果に影響が出たためだと考えられる。

文書データは 2 と 3 では単語数は 1.5 倍であり、良い結果の場合のユニット数が 10 と 20、オーバークラスタリング率が 100 と 200 であり、どちらも 2 倍の値となっている。単語数が 1.5 倍となると、各変数は 2 倍程度必要となると考えられる。これは全結合層での変換に起因すると考えられる。単語数が多すぎると、各単語のベクトルの意味を表す重要な値を、捉えることが出来ないと考えられる。よって単語数はクラスタリング結果に影響を与える。

6.2.4　階層的クラスタリングとの比較

ward 法での階層的クラスタリングでは、情報量最大化クラスタリングと比較すると、全体的に各クラスタ内の単語の類似性が低く、類似性の高い単語が集まったクラスタは少なかった。しかし情報量最大化クラスタリングで、オーバークラスタリング率が極端に低い場合や、ユニット数が多すぎる場合と比較すると、各クラスタ内の単語の類似性に大きな違いはなく、クラスタリング精度は同程度であると考えられる。またクラスタ数を増やした場合でも、強調されている単語と属しているクラスタに大きな変化はなかったため、これらの単語はクラスタ数による変化を受けないほど、このクラスタの中心座標との類似性が高いと考えられる。この点においては情報量最大化クラスタリングより関連する単語がまとまりやすいと考えられる。オーバークラスタリング率や出力層、クラスタ数を適切に設定できれば、情報量最大化クラスタリングの優位性は高いと考えられる。クラスタ数による変化では、クラスタが 1 増えると、増える前の WordCloud では強調されていない単語が同一クラスタとして分類されることが多いことが分かった。これはクラスタ数を任意に指定しているため、文書内の単語に対する適切なクラスタ数を指定できていないためであると考えられる。また強調されていない単語はそのクラスタの中心座標との類似性が元々低かったため、新たにクラスタが増えるとそのクラスタに属しやすくなると考えられる。

第7章　結論

本研究では、livedoor ニュースの記事について、Word2vec と情報量最大化クラスタリングを用いた単語分類を提案し、従来手法である ward 法での階層的クラスタリングとの比較によってその優位性を検証した。また情報量最大化クラスタリングは、従来の教師なし学習での問題点を解消する手法であり、オーバークラスタリング率やユニット数、クラスタ数を変化させることで、クラスタリング精度を高め、有用性を検証した。

手順としては、初めに livedoor ニュースの記事の取得、形態素解析、Word2vec による単語のベクトル化、情報量最大化クラスタリングとクラスタ分析を行い、分類結果を WordCloud で表した。情報量最大化クラスタリングではデータセット生成、ニューラルネットワークの構築、学習、分類を行った。ward 法による階層的クラスタリングでは、クラスタの中心座標と各単語間の COS 類似度によって WordCloud で可視化した。

情報量最大化クラスタリングにおけるオーバークラスタリング率とユニット数を変えると、入力ベクトルに対して、オーバークラスタリングの際にどの程度、単語を分類するか、どの程度特徴量を取り出すかが変化する。良い結果を得るには入力ベクトルの次元数に対して少なすぎず、多すぎない適切な値を設定する必要性があると考えられる。クラスタ数については、元のジャンルを考慮するかによって、関連度は異なる。文書データについては、元の文書内に含まれている類似度の高い単語の量が大きく影響を与えると考えられる。

階層的クラスタリングである ward 法に対しては、オーバークラスタリング率とユニット数、クラスタ数を適切に設定すれば、提案手法の方が関連する単語がまとまりやすくなった。しかし適切に値を設定していないと、階層的クラスタリングの方がまとまりやすかった。よって適度なオーバークラスタリング率、ユニット数、クラスタ数を定義できれば、提案手法の有効性は高いと言える。単語の類似度計算での優位性と、階層的クラスタリングに対しての優位性を確認できたため、文書の類似度計算において、これまでの階層的クラスタリングにかわり、提案手法の情報量最大化クラスタリングでの結果を応用することができると考えられる。

しかし、本実験を通して 2 つの懸念点が浮かんできた。1 つは今回結果の比較は、WordCloud を見て、どのような単語は分類されているか、目視で判断している。そのため、強調されていない単語や、単語の類似性などが、数値などの定量的な判断ではなく、定性的

な判断となっている点である。2 つ目は、MeCab による形態素解析を行う前の元の文書データ内に、どれほど有用性のある単語が含まれているか、類似度の高い単語が含まれているかによって、より同一クラスタ内に関連する単語がまとまりやすくなるか変わってしまう点である。これらの懸念点を改善し、ユニット数やオーバークラスタリング率の適正値を求め、より関連する単語が多く集まるように、検証を行うことで、さらに実用性が高まると考えらる。

謝辞

最後に、本研究を進めるにあたり、ご多忙中にも関わらず多大なご指導をしていただきました出口利憲先生、また、共に勉学に励んだ同研究室のメンバーに厚く御礼申し上げます。

参考文献

1) 元田浩　津本周作　山田高平　沼尾正行　共著, データマイニングの基礎, オーム社, 2008,
2) 金子　冴, 【技術解説】形態素解析とは？MeCab インストール手順から Python での実行例まで, ミエルカ AI media, 2018-05-14,
https://mieruca-ai.com/ai/morphological_analysis_mecab, (参照 2024-01-16)
3) 新納　浩幸　古宮　嘉那子　著, 文書分類からはじめる自然言語処理入門ー基本から BERT までー, 科学情報出版株式会社, 2022
4) ヤン・ジャクリン, 機械学習で「超重要な」特徴量とは何か？ 設計方法などについてわかりやすく解説する, ビジネス+IT, 2021,
https://www.sbbit.jp/article/cont1/76066, (参照 2024-01-20)
5) SONY, ディープラーニングにおける中間層の役割とは？基本的な仕組みや考え方を解説,
https://dl.sony.com/ja/deeplearning/about/middle_layer.html, (参照 2024-01-24)
6) atmarkIT,［活性化関数］ソフトマックス関数（Softmax function）とは？, 一色政彦,2023-10-02,
https://atmarkit.itmedia.co.jp/ait/articles/2004/08/news016.html, (参照 2024-01-16)
7) 機械学習ナビ, ソフトマックス関数 (softmax 関数) とは?機械学習の視点で分かりやすく解説!!, 2021-11-19,
https://nisshingeppo.com/ai/softmax-function/, (参照 2024-02-06)
8) AI 教師あり学習の精度を超えた !? 相互情報量の最大化による教師なし学習手法 IIC の登場！, 2020-02-01,
https://ai-scholar.tech/articles/treatise/iic-ai-367, (参照 2024-01-25)
9) 高校数学の美しい物語, 相互情報量の意味とエントロピーとの関係,2022,
https://manabitimes.jp/math/1403, (参照 2024-01-25)
10) 初心者 DIY プログラミング入門,【実践】Python で WordCloud（ワードクラウド）しようぜ！,2023-11-10,
https://resanaplaza.com/2022/05/21/【実践】python で wordcloud（ワードクラウ

ド）しようぜ！/, (参照 2024-01-28)

11) Smiley,PyTorch とは？特徴やメリットからインストールの方法まで解説,2024-01-25, https://aismiley.co.jp/ai_news/pytorch/, (参照 2024-02-01)

12) Qiita,Pytorch のニューラルネットワーク (CNN) のチュートリアル 1.3.1 の解説,2020-01-29,@poorko, https://qiita.com/poorko/items/c151ff4a827f114fe954, (参照 2024-01-24)