卒 業 研 究 報 告 題 目

Word2vecを利用した単語のクラスター分析による文書の分類

Classification of Documents
by Cluster Analysis of Words using Word2vec

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

2017E33 藤井　康太

令 和 0 4 年（ 2 0 2 2 年 ）　　2 月 1 7 日 提 出

Abstract

The purpose of this study is to confirm the effectiveness of dimensionality reduction by cluster analysis proposed in the laboratory and the usefulness of the formula introduced. This method can reduce the dimension by considering the meaning of the word. We classify texts in experiments. This experiment compares latent semantic analysis with dimensionality reduction by cluster analysis The text used is a synopsis of the novel. We analysed using Python. Proposed dimensionality reduction by cluster analysis is performed using Word2vec. The text classification is done by cluster analysis using cosine similarity as the distance. The method of cluster analysis is Hierarchical clustering by the Ward's method. Figure 1 is an example of a dendrogram. The dendrogram and accuracy rate made from the analysis results are used to verify the effectiveness of the dimensionality reduction method. As a result of the experiment, the effectiveness of dimensionality reduction by cluster analysis was confirmed. The dimensionality reduction by cluster analysis could distinguish the meaning of the word. We used the mean vector to obtain the words that characterize each cluster. In addition, the usefulness of the formula introduced in this study was confirmed.

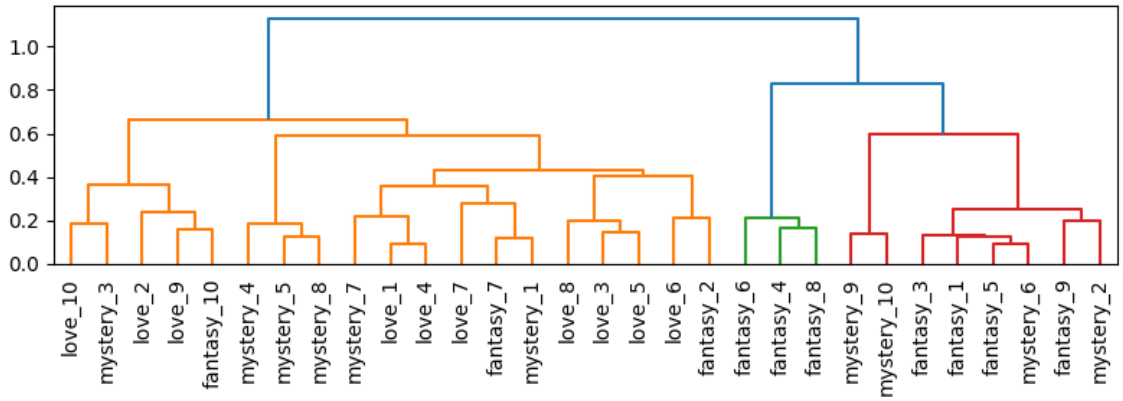

Figure 1 An example of dendrogram.

目 次

第1章　序論

現代社会では、コンピュータやスマートフォンといった情報端末から、膨大な量のデータが発信、収集、記録されており、検索をすれば欲しい情報は簡単に入手することができる。しかし、中には虚偽の情報や信頼性の低い情報も存在し、個人が有益な情報を見つけ出すことは困難になりつつある。そうした問題を解決するため、大量のデータを統計学や人工知能などの分析手法を駆使して、有益な情報を見出すための技術が生み出された。これを、データマイニングという。その中でも、大量のテキストデータに対し、言語処理技術を用いてデータ解析することをテキストマイニングという。テキストマイニングは、様々な事に応用されているが、現状では課題も多く、完成された技術ではないとされている [1)]。

本研究の目的は、文書間の類似度計算に活用される次元削減という技術において、研究室で提案された手法を実装し、その有効性を確認することと、その手法のなかで用いられる新しい計算式を提案し、その式の利点の確認を行うことである。提案された次元削減手法は、単語分散表現が取得できる Word2vec とクラスター分析の手法を併用したものであり、これにより単語が持つ意味を考慮した次元削減が可能であるとされた。また、クラスター分析による次元削減を行うなかで、新たな計算式を提案する。その計算式では、各クラスタの平均ベクトルを求めている。その平均ベクトルと Word2vec を用いることで、各クラスタの特徴を示す単語を得ることができると考えられる。類似度を計算するテキストデータの対象には、小説のあらすじを採用した。取得した小説のあらすじはいくつかのジャンルに分けられており、文書間の類似度が高いものほど似た作品、また同じジャンルに属する小説であると考えられるためである。

実験では、文書間の類似度について、従来の統計的な次元削減手法である潜在的意味解析 (LSA) を適用したものと、本研究室で提案された次元削減手法を適用したものをそれぞれ導出する。それをもとに、デンドログラムと呼ばれるデータの関係を表した樹形図を出力し、比較する。本研究ではそれに加え、機械学習モデルに用いられる評価指標を、本実験の出力結果に適用することで、有効性を判断するための材料とした。

これらを踏まえ、実験の条件に変化を付けた結果をいくつか出力することで、単語間の意味を考慮した次元削減の有効性の検討と、クラスター分析による次元削減での本研究より導入した式の利点の確認を行う。

第2章　実験で使用した技術・手法

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を解析することで、有益な情報や意思決定に必要とされる知識を特定するために用いられる手法のことである。人の手では確認作業でさえ多大な作業量を要し、有益な情報を発見することは困難である。そこで、コンピュータを活用することにより、高速かつ低コストでデータの法則等を発見することができる。このような場面で、データマイニングは用いられる。

2.1.2　テキストマイニング[2)]

テキストマイニングとは、テキストデータを対象としたデータマイニングのことである。言葉は意味を持っているため、数値を対象としたマイニングと比較して、分析の難度が高いとされている。テキストデータには、SNS の投稿やサービスに対するアンケート結果などの文書が例として挙げられ、それらをもとに市場におけるトレンドの分析や文書の検索・分類といった処理がされている。

2.1.3　形態素

形態素とは、単語において意味を持つ最小の単位のことで、それ以上分割できない言葉のことである。形態素と単語と異なる場合があり、形態素がそのまま単語となる場合と、そのままでは単語にならない場合がある。

2.1.4　形態素解析[2)]

形態素解析は、文章を文法や品詞の情報をもとに、形態素に分析する処理のことである。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。

形態素解析で文章を単語に分割する理由は、多くのテキストマイニングの技術において単語を入力値として与えて処理をすることが多いためである。

形態素解析を行うためのツールはオープンソースで公開されているものもある。本研究ではその中から MeCab というソフトウェアを使用した。

2.1.5 MeCab

MeCab 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である。日本語に対応しており、日本で使用されるツールの中でメジャーなものの一つである。MeCab は C 言語で作られたプログラムであり、多くの言語環境で使用できるよう作成された。

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。以下に、形態素解析を行った例を示す。

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 [2)]

自然言語とは、人間が日常的に使用し、意思疎通を行うための言語のことである。例として、日本語や英語、中国語などが挙げられる。自然言語と対比する概念として、人間によって人工的に作りだされてきた言語である人工言語が存在する。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。

自然言語と人工言語には、文法や解釈のしやすさといった違いがある。自然言語は、読み手によって単語や文が複数の意味に解釈されることがある。これに対し、人工言語は、解釈が複数存在することはなく、一通りである。これはコンピュータが簡単に意味を解釈、実行できるようにするためである。

2.2.2 自然言語の曖昧さ

自然言語の曖昧さには、多義性と類義性の二つの性質がある。

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をいう。多

義性の例として「ところ」という単語を挙げる。「ここは僕が昔住んでいたところです」という文では、ある具体的な場所を表していることがわかるが、「彼はたった今来たところだ」では時間的な表現として使われていることがわかる。このように「ところ」という単語には 2 つ以上の解釈が存在することになる。

類義性とは、異なる単語が同じような意味を持つという性質のことを言う。「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。こうした曖昧さが、テキストマイニングの分析の難易度を高めている。

2.3 計算手法

2.3.1 Tf-Idf[3)]

Tf-Idf とは、各文書中に含まれる各単語が、その文書内でどれくらい重要かを表す統計的尺度である。文書を特徴づける重要な単語を抽出したいときや、文書の類似性を測りたいときに有効である。Tf-Idf は Tf と Idf の積で求められる。Tf とは Term frequensy の略称であり、文書中の単語の出現頻度を表す。同じ単語が文書内で多く出現するほど Tf は大きくなり、重要であると考えられる。Idf とは Inverse document frequensy の略称である。単語の逆文書頻度と呼ばれ、文書集合における単語の分布の偏りを考慮する値である。ほとんどの文書に出現している単語は、あまり重要ではない単語と考えられ、Idf が低い値となる。逆に、ある文書にしか出現しない単語はその文書を特徴づけする重要な単語と捉えられ、Idf が高い値となる。Tf-Idf はこれら 2 つの数値の積となる。Tf と Idf には、いくつかの導出方法があるが、本研究では下記の式で計算を行った。

$$Tf_{ij} = \text{文書 } d_i \text{における } t_{ij} \text{の出現回数} \tag{2.1}$$

$$Idf_{ij} = \log \frac{\text{全文書数} + 1}{\text{単語 } t_{ij} \text{が出現する文書数} + 1} + 1 \tag{2.2}$$

$$TfIdf_{ij} = Tf_{ij} \times Idf_{ij} \tag{2.3}$$

2.3.2 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。cos 値が 1 に近いほど、ベクトル同士の挟む角は小さくなり、同じ方向を向いていることとなる。逆に cos 値が -1 に近いほどベクトル同士の挟む角は大きくなり、逆の方向を向いている

こととなる。本研究では、二つの文書ベクトルのなす角度の近さ、すなわち文書間の類似度を求めるために使用した。

ベクトル $\boldsymbol{a}$ とベクトル $\boldsymbol{b}$ の cos 類似度は以下の式で導出される。

$$\cos(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a} \cdot \boldsymbol{b}}{|\boldsymbol{a}||\boldsymbol{b}|} \tag{2.4}$$

2.3.3 主成分分析 [4)]

実験において、測定値の項目が少ない場合はグラフや図を用いて人の目でも容易に判断が可能である。しかし、項目数が多い場合、グラフや図では表現することが難しく、人の目による確認が困難になる場合がある。このような問題を解決する手法として、主成分分析 (principal component analysis; PCA) が存在する。

主成分分析とは、ベクトルなどの多次元のデータについて、本来持っている情報をできる限り損なうことなく次元を削減する手法である。この手法を用いると、変数が何百もある多次元データを 10 や 20 といった少ない次元数まで削減することができる。これにより、データ間の比較や可視化が容易となり、視覚的にも理解しやすくなる。

具体的な手順については、以下に式を交えて説明する。P 個のデータ $x_p (p = 1, 2, \ldots, P)$ について、$N (N \leq P)$ 個の主成分 $z_n (n = 1, 2, \ldots, N)$ とこれらの関係は、次の式のように互いに独立な線形結合として表される。

$$z_n = \sum_{p=1}^{P} a_{pn} x_p \tag{2.5}$$

ここで、z_n は第 n 主成分と呼ばれ、その結合係数 a_{pn} は次の式を満たす必要がある。

$$\sum_{p=1}^{P} a_{pn}^2 = 1 \qquad (\forall n) \tag{2.6}$$

主成分ができる限り多くの情報を持つようにするためには、データの分散に着目し、結合係数を上手く決める必要がある。例として、Figure 2.1 に示す二次元のデータについて考える。この図において、データの分散が最も大きくなる方向に着目すると、z_1 という軸ができる。これが第 1 主成分となり、このような軸ができるように式 (2.5) の結合係数を決定する。しかし、この軸だけでは、データが本来持つ情報を十分に表しているとは言い難い。そこで、z_1 に次いでデータの分散が大きくなる方向に着目し、z_2 という軸

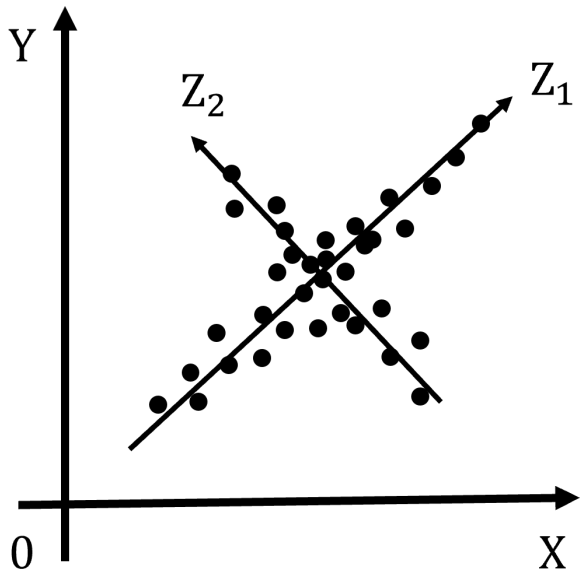

Figure 2.1　Example of two-dimensional data.

をとる。これが第 2 主成分となり、第 1 主成分だけでは表すことができないデータを補うことができる。このように結合係数を決めていくことで、情報量の損失を最小限に抑えながら、Figure 2.1 に示される X, Y の特性を把握することができる。今回の例では 2 次元データであったため、目視で判断しやすいものであった。そのため主成分分析の利点は少ないように思える。しかし、高次元のデータでは、データ量だけでなく分散も大きくなるため、その利点が大きく表れるようになる。

2.3.4　主成分数の選択

主成分分析において、主成分数をいくつに設定するかは極めて重要となる。主成分数が少なすぎると、重要な情報までも損なわれてしまい、解析データに綻びが生じてしまう。逆に主成分数が多すぎると、データの削減が十分ではなくなり、主成分分析を行う意味がなくなってしまう。そのため、以下に示す 3 通りの方法によって主成分数を設定する。

- 固有値が 1 を越える主成分を採用する。
- ある固有値とその次の固有値の差が小さくなるまでの主成分を採用する。
- 累積寄与率がある値に達するまでの主成分を採用する。

一つ目の方法は、平均と分散を共に 1 としたことで、分散 (固有値) がこの標準化された値である 1 よりも大きければ、説明力のある主成分として用いることができるという

考えに基づいている。二つ目の方法は、ある固有値とその次の固有値の差が小さければ、主成分の採用、非採用の区別に大きな意味はないという考えに基づいている。三つ目の方法は、主成分分析後のデータが、主成分分析前のデータが持つ情報の何割かを含んでいればよいという考えに基づいている。主成分ひとつひとつの情報量を計算していき、その情報量の合計が 60〜80 パーセントである主成分数で次元削減される場合が多い。

累積寄与率とはすべての主成分の寄与率を合計した値である。また寄与率とは、ある主成分のあらわす情報は、全体の情報に対してどの程度の情報を含んでいるかを表す値である。上記、3 つ目の方法の説明において記述した、情報量の合計が累積寄与率にあたり、主成分ひとつひとつの情報量が寄与率にあたる。寄与率は次式で表される。

$$P_n = \frac{\lambda_n}{\sum_{p=1}^{P} \lambda_p} \tag{2.7}$$

ここで、式中の λ_n は、n 番目の主成分の固有値を示している。累積寄与率はこの寄与率の総和であるため、主成分数が N の場合は次式で表される。

$$C_n = \sum_{i=1}^{N} P_i \tag{2.8}$$

2.3.5 潜在的意味解析

テキストマイニングにおいて、形態素解析後に生成される単語-文書行列が生成される。単語-文書行列は式 (2.9) で表される。

$$TD = \left(\begin{array}{c|cccc} Term & doc_1 & doc_2 & \cdots & doc_N \\ \hline w_1 & I_{w_1,doc_1} & I_{w_1,doc_2} & \cdots & I_{w_1,doc_N} \\ w_2 & I_{w_2,doc_1} & I_{w_2,doc_2} & \cdots & I_{w_2,doc_N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_M & I_{w_M,doc_1} & I_{w_M,doc_2} & \cdots & I_{w_M,doc_N} \end{array}\right) \tag{2.9}$$

単語-文書行列は高次元である事が多い。それにより計算処理に時間をかけてしまうことや、分析には必要ない単語が含まれており後々の妨げとなることがある。これらの問題を解決するときに、潜在的意味解析 (Latent Semantic Analysis; LSA) が用いられる。潜在的意味解析は、文書データから潜在的なトピックを推定し、そのトピック数まで次元

を削減する手法である。

潜在的意味解析では、特異値分解 (singular valuedecomposition; SVD) という行列分解手法を用いて次元を削減する。以下に、文書行列TDを特異値分解する式を示す。

$$TD = U\Sigma V^{\top} \tag{2.10}$$

この式におけるU, Σ, V^Tは行列を表しており、右辺は文書行列TDを3つの積で表したものである。Uは左特異 (ターム) ベクトル、Σは特異値を含むベクトル、V^Tは右特異 (文書) ベクトルと呼ばれる。特異値分解で得られた左特異ベクトルは含まれる情報の重要度が高い成分から順に並んでいる。そのため、行列の左からk列を抜き出した行列U_kと文書行列TDを式 (2.11) のように計算することで、元の行列の持つ情報をできるだけ保ったまま次元を削減した行列を生成することが可能となる。

$$TD_k = {U_k}^{\top} TD \tag{2.11}$$

2.3.6 クラスター分析

クラスター分析とは、様々な性質が混在したデータから、似た性質を持つ者同士に分類する手法である。教師なしの分類手法であり、主にデータの傾向をつかみたい場合に使用される。分類により作られたデータの集合はクラスターと呼ばれる。

クラスター分析には、分類が階層的になる階層的クラスター分析と、あらかじめクラスター数を指定して分類する非階層的クラスター分析の二種類に分けられる。本研究では、階層的クラスター分析を採用した。階層的クラスター分析とは、データ間の距離をもとに、距離が近いものから順にクラスターを作成していき、最終的に階層のようなクラスター構造を形成する分類手法である。形成されたクラスターの構造は、Figure 2.2 に示すデンドログラムによって視覚的に判断が可能となる。デンドログラムにおいて、結合点が末端に近いデータほど類似性の高い関係であるといえる。階層的クラスター分析は、クラスター間の距離測定の方法にいくつか種類があるが、本研究ではウォード法を採用した。ウォード法は2つのクラスターを融合した際に、同クラスター内の分散と他クラスター間の分散の比を最大化するようにクラスターを形成していく方法である。

2.3.7 正解率 [5)]

正解率 (Accuracy) とは、機械学習の分類問題に用いられる評価指標のひとつである。本研究では、分析手法の評価項目として導入した。分類問題の評価指標には正解率のほか

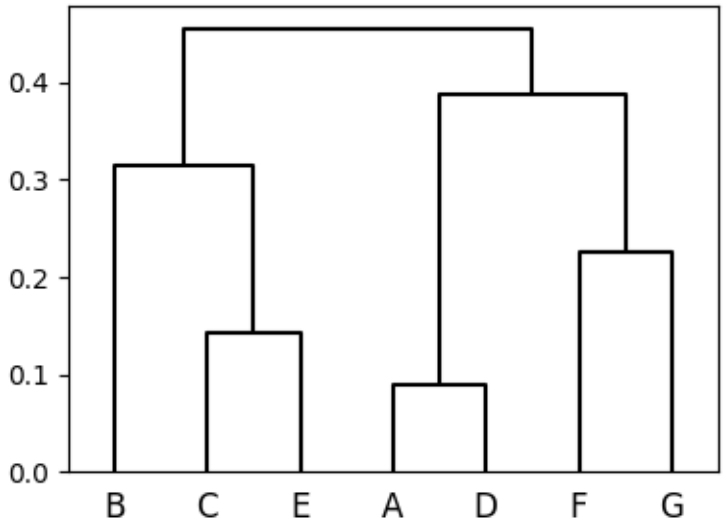

Figure 2.2　Dendrogram.

に適合率、再現率、F1 値といった値がある。これらは、ラベルごとのデータ数に偏りがある場合や重要視する評価の側面に応じて使い分けされる。本研究では、全体のパフォーマンスをわかりやすい数値で知りたかったことや分析データの特徴を考慮した結果、正解率を採用することにした。

正解率は式 (2.12) のように計算される。

$$\text{正解率} = \frac{\text{正解した数}}{\text{予測した全データ数}} \tag{2.12}$$

例として、正解した数が 24、予測した全データ数を 30 として実際に計算すると、正解した数 24 をデータの全体数である 30 で割った 0.80 が正解率となる。式 (2.12) からもわかるように、正解率は 1 に近いほど正しい予測がなされたという見方になる。

ここまで正解率について説明を行ってきたが、本来、教師なし学習に当たるクラスター分析の評価はほぼ不可能とされている。なんらかの正解データをもとに分類を行っているわけではないことや、人間の判断基準ではわからない評価を含んでいる可能性もあり、分析結果を見る側面によって良し悪しが変化するためである。そのため正解率も、通常はクラスター分析の評価に使用できるものではない。

しかし本研究では、あらかじめ正解ラベルの付いたデータを分析対象にすることで、疑似的に評価を導入できる環境を整えた。ここで、本研究における正解ラベルとは分析対象である小説のあらすじに割り振られた各ジャンルとなる。同じジャンルのあらすじなら、同一のクラスターとして分類され、異なるジャンルであれば別のクラスターに分類されるだろうという考えを適用したものである。これにより、分析手法の評価・比較

が可能となる。

2.4 WebAPI

2.4.1 API

API とは Application Programming Interface の略称で、あるソフトウェアから他のソフトウェアを制御するためのインターフェース（規約）を意味する。何かしらの決められた API がシステムに存在する場合、それを介することでシステムの内部構造を深く理解することなく、その機能を呼び出すことが可能となる。API の目的には、ソフトウェア開発における開発工程の大幅な削減や開発における標準化、利便性の向上などが挙げられる。

2.4.2 WebAPI[6)]

本研究で用いた WebAPI は、HTTP プロトコルを利用してネットワーク越しに呼び出す API である。ユーザー側がある URL にアクセスすることで、サーバ側の情報の書き換えやサーバ側に保存されている情報を取得できることが可能なウェブシステムのことを指す。プログラムからアクセスすることで、そのデータを機械的に利用することなどに用いられることが多い。

2.4.3 WebAPI を利用したテキストデータの取得[7)]

本研究では解析するテキストデータの取得に WebAPI を利用した。利用した API は、株式会社ヒナプロジェクトが運営する投稿型小説サイト「小説家になろう」に用意されたなろう小説 API というものである。この WebAPI は、ホームページやブログの管理者そして、システムエンジニア、プログラマに向けた各種技術情報の公開を目的として提供されている。いくつかのオプションを指定し、特定の URL にリクエストを行うことで Web サイトに投稿されている小説の情報が取得できるよう開発された。取得できる情報には、小説タイトル、小説のあらすじ、ジャンル、作者や小説の評価などが挙げられる。本研究では、小説タイトルとあらすじ、ジャンルを対象として作品情報の取得を行った。

2.5 Python

2.5.1 Python[1)]

Python とは、1991 年にオランダ人のグイド・ヴァン・ロッサム氏により開発された汎用プログラミング言語である。Python の特徴として以下のような点が挙げられる。

- インタプリタ形式の、対話型の言語
- オブジェクト指向プログラミング言語
- 移植が容易で、多くの Unix 系 OS、Mac、Windows で動作が可能
- オープンソースで運用されている
- コードに記述がシンプルであり、可読性が高いとされている
- 汎用的なライブラリから専門的なライブラリまで豊富に用意されている

こうした特徴から、人工知能をはじめとした様々な分野で活用されており、現在では何百万人ものユーザーが利用しているとされている。

2.5.2 MeCab[8)]

本研究では、Python から MeCab を呼び出すことで形態素解析を行った。MeCab の辞書には、標準のシステム辞書ではなく mecab-ipadic-NEologd という辞書を採用した。

mecab-ipadic-NEologd は、新語・固有表現に強く、語彙数が多いオープンソースソフトウェアとして公開されている。本研究の解析対象である小説のあらすじは、現在も投稿・更新が盛んにおこなわれている小説投稿サイトのものである。そのため、デフォルトの辞書には登録のない新しい語や表現も使用されている可能性があると考えられた。

また、mecab-ipadic-NEologd は厳密には形態素解析用の辞書ではなく単語分かち書き用の辞書とされている。辞書に登録されている単語によっては、形態素まで分解されていないものもあるためである。しかし、語彙数や多くの固有表現を網羅していることから、今回の実験においてはこちらの辞書の方がより正確に文書を解析できると判断した。

そうした理由から、mecab-ipadic-NEologd を本研究に使用する辞書として導入した。

2.5.3 WebAPI

Python では Requests というライブラリを使用することで HTTP 通信を行うことができる。HTTP 通信では利用目的に応じてリクエストメソッドを指定し、実行する。本研究では API を介してデータの取得を行うため、プログラムでいくつかのオプションを指

定した後、GET メソッドによる通信を行った。

2.5.4 Tf-Idf

Python では、scikit-learn ライブラリに用意されている TfidfVectorizer 関数を利用して Tf-Idf の計算が行える。TfidfVectorizer では、文字列のリストを入力として与え、いくつかのオプションを指定することで式 (2.1)、(2.2) の通りに Tf-Idf が導出される。また Tf-Idf の出力以外にも、計算に使用した単語の一覧を出力することも可能である。

2.5.5 cos 類似度

Python では scikit-learn ライブラリに用意されている cosinesimilarity 関数で cos 類似度の計算を行える。また、scipy ライブラリに用意されている pdist 関数では距離の公理に当てはめた cos 類似度の計算が行える。本来、cos 類似度は距離ではないため、距離として扱うためには別の計算が必要となる。

2 つのベクトルをベクトル $\vec{a}$ とベクトル $\vec{b}$ 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。

$$\cos\left(\vec{a},\vec{b}\right)_{distance} = 1 - \frac{\vec{a}\cdot\vec{b}}{|\vec{a}||\vec{b}|} \tag{2.13}$$

本研究では、文書間の類似度を値として確認する際に cosine-similarity 関数を使用し、pdist 関数はクラスター分析に与えるデータを計算する際に使用した。

2.5.6 主成分分析

Python では、scikit-learn ライブラリに用意されている PCA 関数で主成分分析を行うことができる。PCA 関数では、引数の n_components に削減後の次元数を指定することで主成分分析が行える。主成分分析の実行以外にも、各主成分の寄与率や累積寄与率といった値の出力も可能となっている。

2.5.7 潜在的意味解析

Python では、numpy ライブラリに用意されている svd 関数で特異値分解を行うことができる。この関数に Tf-Idf を与えることで、左特異（ターム）ベクトル、特異値、右特異（文書）ベクトルを取得できる。この関数を用いて取得した左特異（ターム）ベクトルと式 (2.11) を用いて潜在的意味解析を行った。

2.5.8 クラスター分析

Python では scipy ライブラリに用意されている linkage 関数で階層的クラスター分析を行うことができる。linkage 関数では、測定方法を指定し、pdist 関数で計算された距離行列を与えることでクラスター分析が実行できる。また分析を行ったデータに対して、fcluster という関数を使用すれば任意の数のクラスターに分類することが可能となる。クラスター分析した結果を樹形図として確認したい場合には、dendrogram 関数を使用すればデンドログラムが出力される。

2.5.9 正解率

Python では、scikit-learn ライブラリに用意されている classification_report 関数に、分析データの正解リストと予測リストを与えることで正解率を計算できる。

2.6 Word2vec

2.6.1 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・公開がされた。Word2vec を利用して単語をベクトル化することによって、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

具体的な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「king」、「man」、「woman」、「queen」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ことから、単語同士で次のような計算を行うことができる。

$$\begin{gathered} 「\text{king}」 - 「\text{man}」 + 「\text{woman}」 = 「\text{queen}」 \\ (\text{“王”} - \text{“男性”} + \text{“女性”} = \text{“女王”}) \end{gathered} \tag{2.14}$$

式 (2.14) は空間上にある単語の加減算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

よる学習済みモデルを用いることで、実際に行うことができる。

2.6.2 Word2vec による単語間の類似度計算

Word2vec では、学習済みモデルに対して単語を指定することで様々な操作が行える。その操作の一つに、指定した単語のベクトル表現の取得がある。これにより、単語がベクトル空間上のどこに位置するかを数値化して取得することができる。また、取得できる値はベクトルであることから、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ー分析を行うことが可能となる。本研究ではこれを利用して、解析対象の文書に含まれる単語間の類似度を導出した。

第3章　提案手法

3.1　Word2vecを用いたクラスター分析による次元削減

本来、クラスター分析は次元削減を行う技術ではない。しかし、Word2vecによる単語間の距離をもとにクラスター分析を行うことで、単語を複数のクラスターに分類することができる。Word2vecを用いて計算される単語間の距離は、単語同士の意味の近さを表すものになる。そのため、その距離を利用して形成されたクラスターは、意味が近い単語が集められたクラスターとなる。これにより、単語の数だけあった次元を、似た意味を持つ単語が集められたクラスターの数まで削減することが可能となる。この手法をクラスター分析による次元削減とする。

Tf-Idfを重要度とした式(2.9)のような文書行列に、クラスター分析による次元削減を行うことで、クラスターと文書からなるクラスター-文書行列が作成される。このとき、クラスター-文書行列の重要度は、式(3.1)で表されるクラスター-単語行列と式(2.9)で示した元の文書行列との積で求められる。

$$CW = \left(\begin{array}{c|cccc} Term & w_1 & w_2 & \cdots & w_M \\ \hline C_1 & a_{1,1} & a_{1,2} & \cdots & a_{1,M} \\ C_2 & a_{2,1} & a_{2,2} & \cdots & a_{2,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_N & a_{N,1} & a_{N,2} & \cdots & a_{N,M} \end{array}\right) \tag{3.1}$$

各クラスタを$C_i(i = 1, 2, \ldots, N)$とする。このとき各クラスタの平均ベクトル$\mathrm{vec}(C_i)$は次の式で求められる。

$$\mathrm{vec}(C_i) = \frac{1}{|C_i|} \sum_{w \in C_i} \mathrm{vec}(w) \tag{3.2}$$

単語を$w_j(j = 1, 2, \ldots, M)$、simをベクトルのcos類似度とする。このとき式(3.1)にある要素$a_{i,j}$は次の式で与えられる。

$$a_{i,j} = \begin{cases} \mathrm{sim}(\mathrm{vec}(C_i), \mathrm{vec}(w_j)) & (w_j \in C_i) \\ 0 & (w_j \notin C_i) \end{cases} \tag{3.3}$$

元々この手法では、$a_{i,j}$ を求めるために、式 (3.4) を用いていた。式 (3.4) において、$S_{k,j}$ は単語 w_k と w_j の cos 類似度を表している。

$$a_{i,j} = \begin{cases} \frac{1}{|C_i|} \sum_{k \in C_i} S_{k,j} & (w_j \in C_i) \\ 0 & (w_j \notin C_i) \end{cases} \tag{3.4}$$

式 (3.3) と式 (3.4) の相違点として、cos 類似度の計算を行うタイミングが挙げられる。式 (3.4) では、まず各単語同士の cos 類似度を計算し、その総和を求めた。その後、それらの平均を求め、$a_{i,j}$ としている。一方、本研究で用いる式 (3.3) では、まず各クラスタの平均ベクトルを求めている。その後、その平均ベクトルとの cos 類似度を求め、$a_{i,j}$ としている。この式を用いる利点は、文書に出てくる各クラスタの平均ベクトルを求めることができる点である。クラスタの平均ベクトルを取得することで、クラスタに登場する単語の傾向をつかむことができる。

このような理由から、本研究では $a_{i,j}$ を求める新しい計算方法として、式 (3.3) を導入する。

第4章　実験結果と考察

4.1　実験の概要

本実験では小説のあらすじを対象として、テキスト間の類似度計算を行う。出力される類似度は、潜在的意味解析を使用したものとクラスター分析による次元削減を使用したものの2種類となる。類似度の計算は累積寄与率をもとに次元数を何度か変更し、複数の結果を出力する。その後、出力された類似度をもとにクラスター分析を行い、テキスト間の関係を比較する。比較の評価には、クラスター分析の結果をもとに生成されたデンドログラムと、結果から導出する正解率を用いる。なお、正解率は評価材料として妥当かどうかわかっていないため、その判定を行うことも踏まえての採用である。この比較により、クラスター分析による次元削減が従来手法である潜在的意味解析と比べ、どのように機能しているかを確認することが本実験の最終的な目的である。

実験では、分析対象である小説のあらすじのジャンル数と作品数を4パターンに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出るか確認するためである。各実験パターンの詳細を以下に示す。

実験パターン1

- ジャンル数：3　作品数：30（各ジャンル10作品）

実験パターン2

- ジャンル数：3　作品数：150（各ジャンル50作品）

実験パターン3

- ジャンル数：6　作品数：60（各ジャンル10作品）

実験パターン4

- ジャンル数：6　作品数：300（各ジャンル50作品）

ジャンルの内容や各パターンについての詳細は4.2.4項で後述する。実験としては、上記4つのパターンのデンドログラムや正解率を比較することで考察を行うこととなる。

4.2　実験の準備

4.2.1　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。

- MeCab の導入
- Word2vec の学習済みモデルの取得
- テキストデータの取得

4.2.2 MeCab の導入

MeCab は、公式から配布されている 32bit 版ではなく、有志によって作成された 64bit 版の MeCab を使用した。これは今回使用する Python の 64bit とバージョンを合わせるためである。それに伴い辞書の mecab-ipadic-NEologd も、開発者である佐藤敏紀氏の GitHub ページより取得した。

4.2.3 Word2vec の学習済みモデルの取得 [9)]

単語のベクトルを取得するには Word2vec のモデルを作成する必要がある。しかし、モデルを作成するための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 年に学習されたモデルを採用した。

4.2.4 テキストデータの取得

分析対象である小説のあらすじは、2.4.3 項で述べた WebAPI を利用することで取得した。取得内容は、小説の作品名とあらすじ、ジャンルである。取得した作品名とあらすじ、ジャンルを csv ファイルにまとめることで解析対象とした。作品の取得を行った「小説家になろう」サイトでは、いくつかのジャンルが存在しており、各作品に 1 つのジャンルが割り当てられている。本実験では、そうしたジャンルの中から以下の 6 ジャンルを取得する作品ジャンルとして採用した。

- ハイファンタジー〔ファンタジー〕
- 現実世界〔恋愛〕
- 推理〔文芸〕
- ホラー〔文芸〕

- 空想科学〔SF〕
- 童話〔その他〕

ジャンルを指定して作品を取得した理由は、クラスター分析による分類が理由として挙げられる。クラスター分析において、同じジャンルの作品は同じクラスターに分類される可能性が高いと考えられ、そうした分類結果が本研究の目的である手法評価につなげることができるためである。指定した6ジャンルの作品は、各ジャンルごとに50作品、計300作品取得した。各作品の長さはジャンルごとに多少ばらつきがあったが、解析ではTf-Idfを用いるため、長さのばらつきはあまり影響はないと考えられる。

取得した作品は、4.1節で述べた4つの実験パターンに分割を行った。各パターンの詳細について順に説明していく。

まずはジャンル数についてである。実験パターン1と実験パターン2では、ジャンル数が3ジャンルである。これは上記のジャンルにあるハイファンタジー〔ファンタジー〕、現実世界〔恋愛〕、推理〔文芸〕の3つである。この3つを採用した理由は、それぞれのジャンルの方向性が異なっており、その違いが分析結果に影響を及ぼすのではないかと考えられたためである。実験パターン3と実験パターン4のジャンル数は6ジャンルである。これは、上記6つのジャンル全てを含んで解析を行うということである。

次に作品数である。実験パターン1と実験パターン3では作品数が各ジャンル10作品とした。これは、選択した各ジャンル50作品の中から、それぞれ10作品ずつを抽出して解析対象にしたものである。実験パターン2と実験パターン4では、各ジャンル50作品すべてを含めたものとした。

以上の通りに、4つの実験パターンを用意して実験を行った。

4.3 データの前処理

4.3.1 テキストデータの形態素解析

取得した小説のあらすじに対して、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。本実験ではテキスト1つの文量がさほど多くないことから、抽出する品詞を名詞（数以外)、動詞、形容詞の3つに設定し、それ以外の単語は削除した。これらの作業を行った結果、各実験パターン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。

- 実験パターン1 (単語種類数) $\cdots$ 1196 語

- 実験パターン 2 (単語種類数) ··· 4303 語
- 実験パターン 3 (単語種類数) ··· 2052 語
- 実験パターン 4 (単語種類数) ··· 6700 語

4.3.2 Tf-Idf の計算

Python を用いて Tf-Idf を計算する。計算は、2.5.4 項に示したように、scikit-learn ライブラリの TfIdfVectorizer 関数に形態素解析を行ったテキストデータを引数として渡すことで行う。ここで、計算が終了した後に TfIdfVectorizer 関数のメソッドである get_feature_names で計算に使用された単語のリストを抽出しておく。クラスター分析による次元削減を行う際に、Tf-Idf で使用された単語一覧が必要となるためである。

4.4 次元削減

4.4.1 実験パターンにおける主成分数の決定

本実験では前述した 2 つの次元削減手法を比較するために、主成分数の変更を何度か行って結果を出力する。主成分数の変更は累積寄与率に基づいて行う。2.5.6 項に示した PCA 関数で主成分分析を行い、累積寄与率が、50%から 55%、60%、65%と 5%ずつ間隔をあけて 90%に至るまでの主成分数を記録していく。そのため各実験パターンの手法ごとに 9 回次元削減を行うことになる。通常は 2.3.4 項にも示したように 60%〜80%の累積寄与率で削減数を決定することが多いが、本研究では次元削減への影響の確認や結果の値を多く取るため、累積寄与率を 50%〜90%という範囲に設定した。Table 4.1 に各実験パターンで主成分分析を行い、決定した主成分数をまとめたものを示す。

4.4.2 潜在的意味解析による次元削減

2.5.7 に示したように、numpy ライブラリの svd 関数に Tf-Idf 値を与えることで、潜在的意味解析による次元削減を行った。次元数は Table 4.1 に示してあるように、各実験パターンの各累積寄与率に適する主成分数に合わせて指定した。

4.4.3 クラスター分析による次元削減

クラスター分析による次元削減は以下の手順で実行した。

1. 単語の意味を、Word2vec を用いてベクトルとして抽出する

Table 4.1　Dimensional quantity of each pattern.

cumulative contribution ratio	**pattern1**	**pattern2**	**pattern3**	**pattern4**
50%	9	51	17	90
55%	11	57	20	103
60%	12	64	22	116
65%	13	71	25	130
70%	15	79	28	145
75%	16	87	31	161
80%	18	96	35	179
85%	20	106	39	198
90%	22	117	43	220

2. 抽出した単語のベクトルを基に、式 (2.4) で単語間の類似度を計算する
3. 単語間のクラスター分析を行い、Table 4.1 にある主成分の数に合わせて、クラスターの数を分割する
4. クラスタ内の総単語数を計算する
5. 各クラスターに含まれる単語を確認し、有無を数値として行列にまとめる
6. 2.，4.，5. を用いて、式 (3.3) の値を求め、行列としてまとめる

1. では Word2vec を用いて単語の意味をベクトルとして取得している。しかし、取得したい単語の中には Word2vec に登録されていない単語も存在する。以下に各実験パターンごとの Word2vec 非登録語の数を示す。この非登録単語に関しては、単語間の類似度が計算できないため、行列からこの単語を削除することで対応した。

- 実験パターン 1 (Word2vec 非登録語) ··· 96 語
- 実験パターン 2 (Word2vec 非登録語) ··· 383 語
- 実験パターン 3 (Word2vec 非登録語) ··· 154 語
- 実験パターン 4 (Word2vec 非登録語) ··· 582 語

3. では単語のクラスター分析を行った後に、Table 4.1 の主成分数に合わせてクラスターの数を分割している。これは潜在的意味解析の次元数と次元を合わせ、なるべく同じ条件で比較を行うためである。

4.4.4　平均ベクトルを用いた各クラスタに含まれる単語の傾向の取得

式 (3.2) で求められる各クラスタの平均ベクトルを用いて、各クラスタを表す単語を取得した。単語の取得方法としては、各クラスタの平均ベクトルとの類似度が高い単語を Word2vec で検索し、最も類似度が高い単語を取得した。

4.5　文書のクラスター分析

4.5.1　文書間の cos 類似度

次元削減された行列に対して、2.5.5 項で示したように cosinesimilarity 関数と pdist 関数を用いることで 2 種類の cos 類似度を計算した。それぞれ cosinesimilarity 関数による cos 類似度は値の確認を行うために計算し、pdist 関数による値は、次に行うクラスター分析用に計算を行った。

4.5.2　クラスター分析

前項で計算した文書間の類似度をもとにクラスター分析を行った。2.5.8 項にあるように階層的クラスター分析を実行し、距離の測定方法には 2.3.6 項で説明を行ったウォード法を指定した。

4.5.3　デンドログラムの出力

2.5.8 項に示したように、linkage 関数と dendrogram 関数を使用することで、デンドログラムを出力した。デンドログラムは、実験パターン 1 と実験パターン 3 に関するものを複数個出力し、確認に使用した。実験パターン 2 と実験パターン 4 に関しては文書の数が多く、目視でのデンドログラムによる確認が難しいため確認には使用しなかった。

4.6　正解率の計算

4.6.1　クラスター分析結果の取得

4.5.2 項で出力した分析結果に対して fcluster 関数を使用し、実験パターンごとに割り当てられたジャンル数までクラスターの分割を行う。実験パターン 1 と実験パターン 2 では 3 つのクラスター、実験パターン 3 と実験パターン 4 では 6 つのクラスターに分割することとなる。これは分割を行ったクラスターに各ジャンルを順に割り当てていき、ジャンルの偏りが発生しているかを調べるためである。また、次項で述べる正解率導出の手

順の一つでもある。

4.6.2　正解率の計算

各実験パターンの各手法でジャンル数に分割されたクラスターに対して、2.5.9 項で示したライブラリを用いて以下の手順で正解率を導出する。

1. 分割されたクラスターに、それぞれ適当なジャンルを割り当てる
2. classification_report 関数で正解率を求める
3. 1. とは別のジャンルを各クラスターに割り当て、2. を行う。以後クラスターに対して割り当てるジャンルを繰り返し変更していき、すべてのパターンの正解率を求める
4. 求め終わった正解率の中で、最も正解率が高いものを選ぶ

これにより、どの程度各クラスター内でジャンルの偏りがあったかを正解率で確認することができる。これが手法評価の材料となる。

4.6.3　グラフの作成

計算された正解率を実験パターンごとに折れ線グラフにまとめる。グラフには、潜在的意味解析による次元削減の正解率、クラスター分析の結果による正解率、2 つの手法の正解率の平均値が 1 つのグラフにまとめて記載されている。

4.7　実験結果

4.7.1　実験結果の評価方法

本実験では、導出した正解率を比較することでクラスター分析による次元削減手法の有効性を評価する。しかし、正解率が手法の評価として使用できるか判断がついていないため、実験パターン 1 と実験パターン 3 の出力結果を用いて、手法評価における正解率の妥当性を決定する。妥当性の判断には正解率のグラフとデンドログラムを用いる。正解率のグラフより、同じ次元数で正解率が大きく離れている場所を見つけ、その時のデンドログラムと正解率の関係を手法ごとに確認する。これは正解率の値によって、デンドログラムのまとまりに差が出ているのではないかという考えから行うものである。正解率が低ければ、デンドログラムにジャンルごとのまとまりがなく、正解率が高ければ、ジャンルごとのまとまりができているだろうという予想のもと行う。実験パターン 1 と

実験パターン 3 にした理由は、デンドログラムを可視化した際に、人間が識別できると判断したのがこの 2 つであったためである。

また、本研究より導入した式 (3.3) の有用性の確認も行う。

4.7.2 正解率の推移

実験パターン 1 の正解率を Figure 4.1、実験パターン 2 の正解率を Figure 4.2、実験パターン 3 の正解率を Figure 4.3、実験パターン 4 の正解率を Figure 4.4 に示す。グラフは縦軸が正解率、横軸が累積寄与率となっており、次元削減数による正解率の推移を表している。グラフの見やすさの観点から、縦軸 (正解率) の最大値を Figure 4.1, Figure 4.2 では 0.8 に、Figure 4.3, Figure 4.4 では 0.4 に設定した。各グラフより、パターン 1 ではクラスター分析による次元削減の正解率に比べ、潜在的意味解析の正解率の方が高い水準となっていることがわかる。それに対し、パターン 2〜4 では潜在的意味解析の正解率に比べ、クラスター分析による次元削減の正解率の方が高い水準となっていることがわかる。同じジャンル数のパターン同士を比較すると、3 ジャンル、6 ジャンルどちらとも作品数の多いパターンの方がクラスター分析による次元削減の正解率が高いことがわかる。また、作品数が増えるほど全体の正解率も低下する傾向がグラフより読み取れる。

4.7.3 潜在的意味解析, クラスター分析による次元削減のデンドログラム

正解率の妥当性を確認するため、実験パターン 1,3 におけるデンドログラムをそれぞれ出力する。デンドログラムは正解率に最も差があるところを各手法ごとに出力するため、計 4 つのデンドログラムを生成することとする。実験パターン 1 では、累積寄与率 65%において、潜在的意味解析による分析結果の正解率がクラスター分析による次元削減のものより高く、グラフの中で最も差が開いていることがわかる。実験パターン 3 では、累積寄与率 85%において、クラスター分析による次元削減の正解率が潜在的意味解析のものより高い値であり、グラフの中で最も差が開いていることがわかる。このことから、実験パターン 1 の累積寄与率 65%における各手法のデンドログラム、実験パターン 3 の累積寄与率 85%における各手法のデンドログラムを出力する。デンドログラムは、各実験パターンのジャンル数と同じクラスター数になるように出力した。その結果、Figure 4.5、Figure 4.6、Figure 4.7、Figure 4.8 のようなデンドログラムとなった。各実験パターンのデンドログラム同士について比較を行う。まず実験パターン 1 からである。

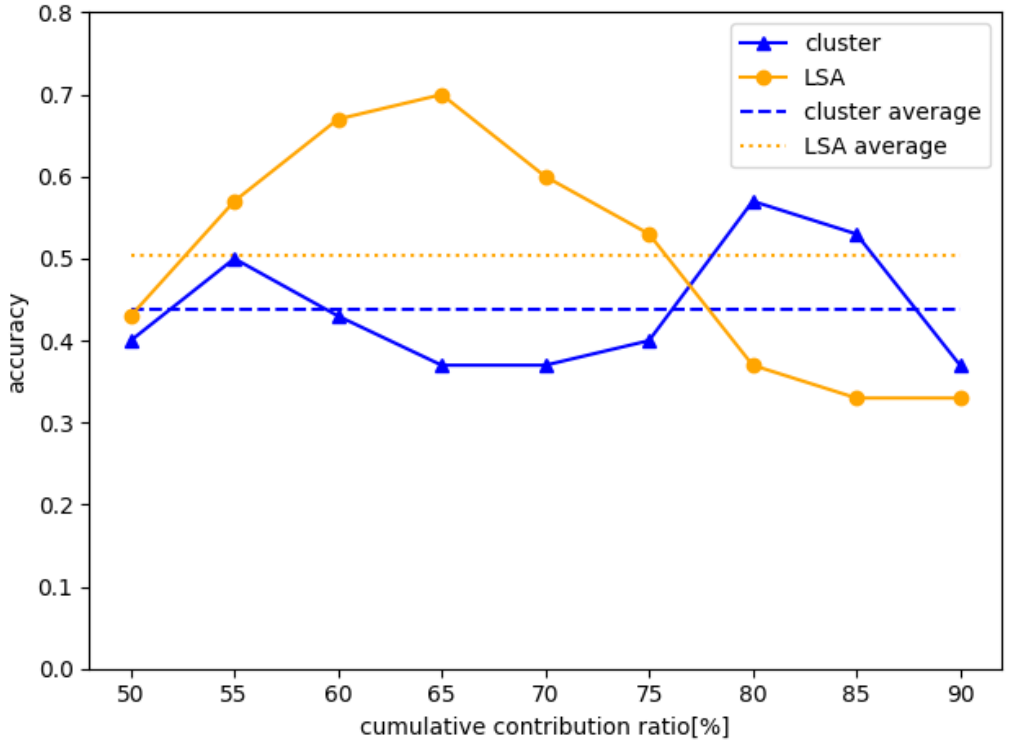

Figure 4.1 Accuracy of 3 genres and 30 works.

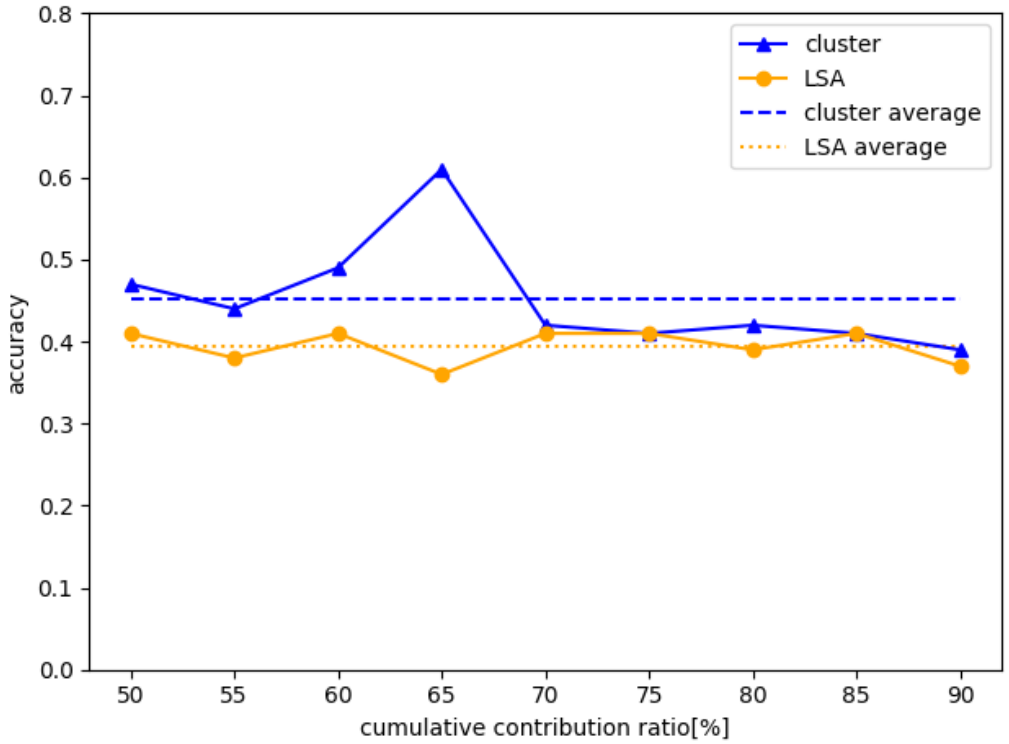

Figure 4.2 Accuracy of 3 genres and 150 works.

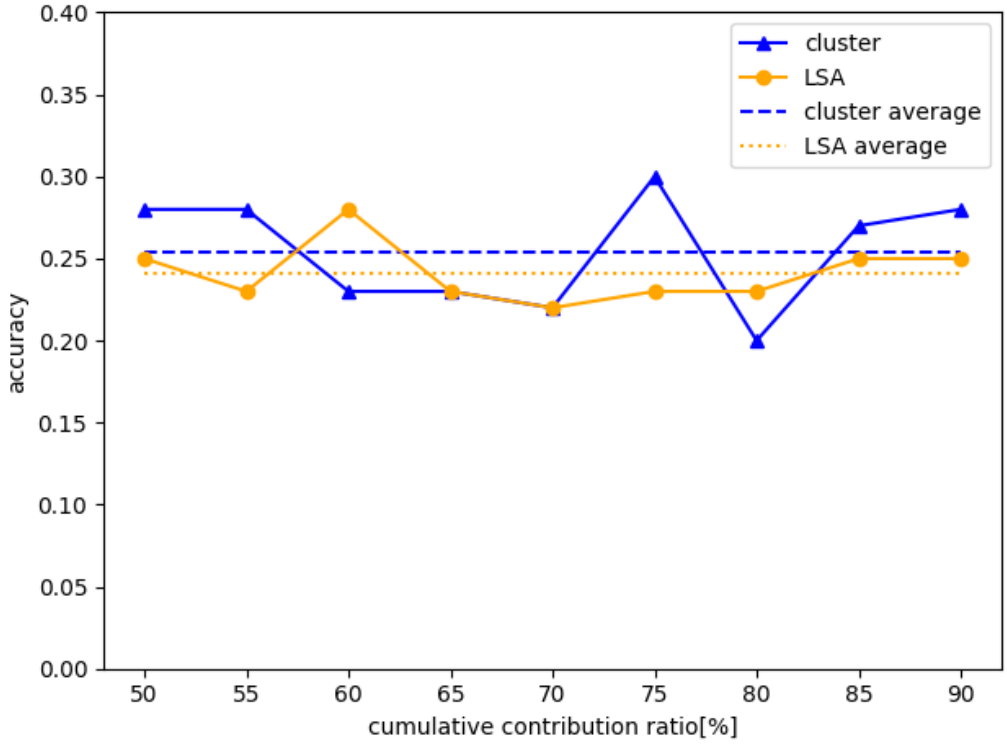

Figure 4.3　Accuracy of 6 genres and 60 works.

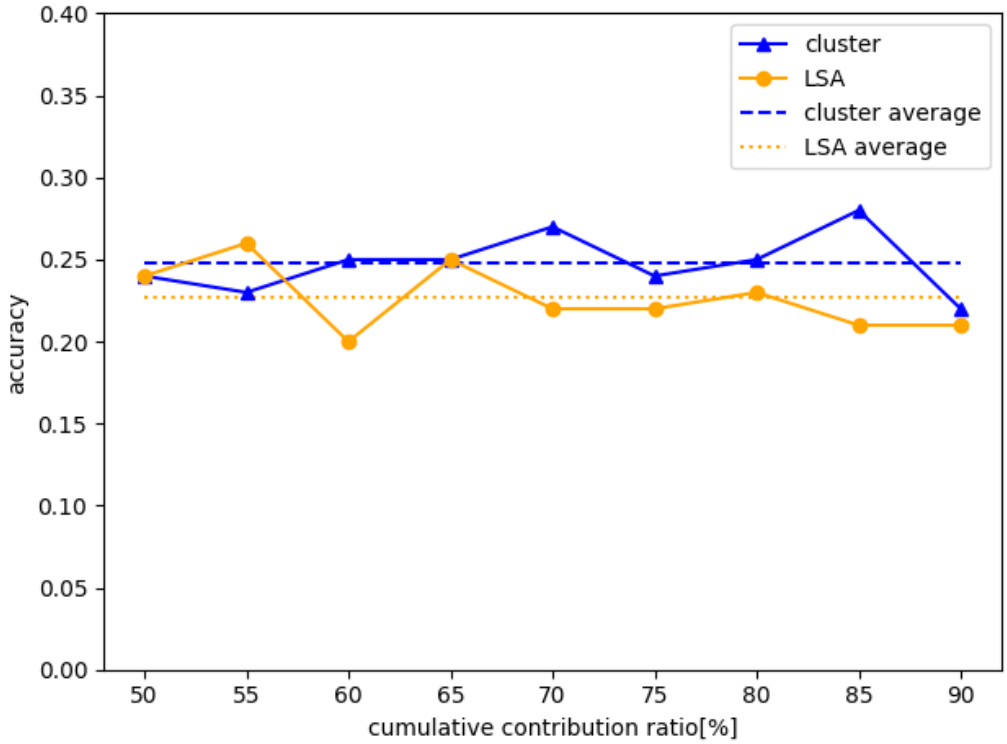

Figure 4.4　Accuracy of 6 genres and 300 works.

Figure 4.5 の実験パターン 1 における潜在的意味解析による結果を確認すると、2 つ目のクラスターは推理、3 つ目のクラスターはファンタジーにほぼ分類できていることがわかる。次に、Figure 4.6 のクラスター分析による次元削減のデンドログラムを確認する。各クラスターを見ると、どのクラスターもうまく分類できていないことがわかる。このことから、実験パターン 1 の累積寄与率 65%では潜在的意味解析のほうがこちらの意図した通りに分類を行えているといえる。

次に実験パターン 3 について比較を行う。Figure 4.7 の実験パターン 3 における潜在的意味解析による結果を確認すると、2 つ目以降のクラスターでは大まかな分類ができていることがわかる。しかし、1 つ目のクラスターでは様々なジャンルが混在しておりうまく分類できていないことがわかる。それに対して、Figure 4.8 のクラスター分析による次元削減のデンドログラムをみると、Figure 4.7 のデンドログラムに比べ、各クラスタ内での文書間の距離が近いことがわかる。これらのことから、クラスター分析による次元削減のほうがこちらの意図した通りに分類を行えているといえる。

4.7.4　平均ベクトルを用いた各クラスタに含まれる単語の傾向の取得

3.1 節で述べた式 (3.3) の有用性を確認するため、式 (3.2) で求められる各クラスタの平均ベクトルを用いて、各クラスタを表す単語を取得した。単語の取得は、Table 4.1 より、最もクラスタ数が少なくなる実験パターン 1 の累積寄与率 50%において行った。

Table 4.2 に、実験パターン 1 の累積寄与率 50%における、各クラスタの平均ベクトルと最も類似度が高い単語を示す。Table 4.2 より、各クラスタの平均ベクトルと最も類似度が高い単語を取得することで、各クラスタに含まれる単語の傾向を取得することができたといえる。

4.8　考察

4.8.1　手法評価における正解率適用の妥当性

4.7.3 項の結果より、本研究の手法評価における正解率の適用は概ね妥当であると考えられる。その理由として、デンドログラムと正解率の関係性が挙げられる。4.7.3 項でデンドログラムと正解率を比較した際に、正解率が高いときは各クラスターがジャンルごとに分類できていたのに対して、正解率が低いときは各クラスターにジャンルごとのまとまりが見られなかった。このことから、正解率が高いほど単語の意味を考慮し、関連

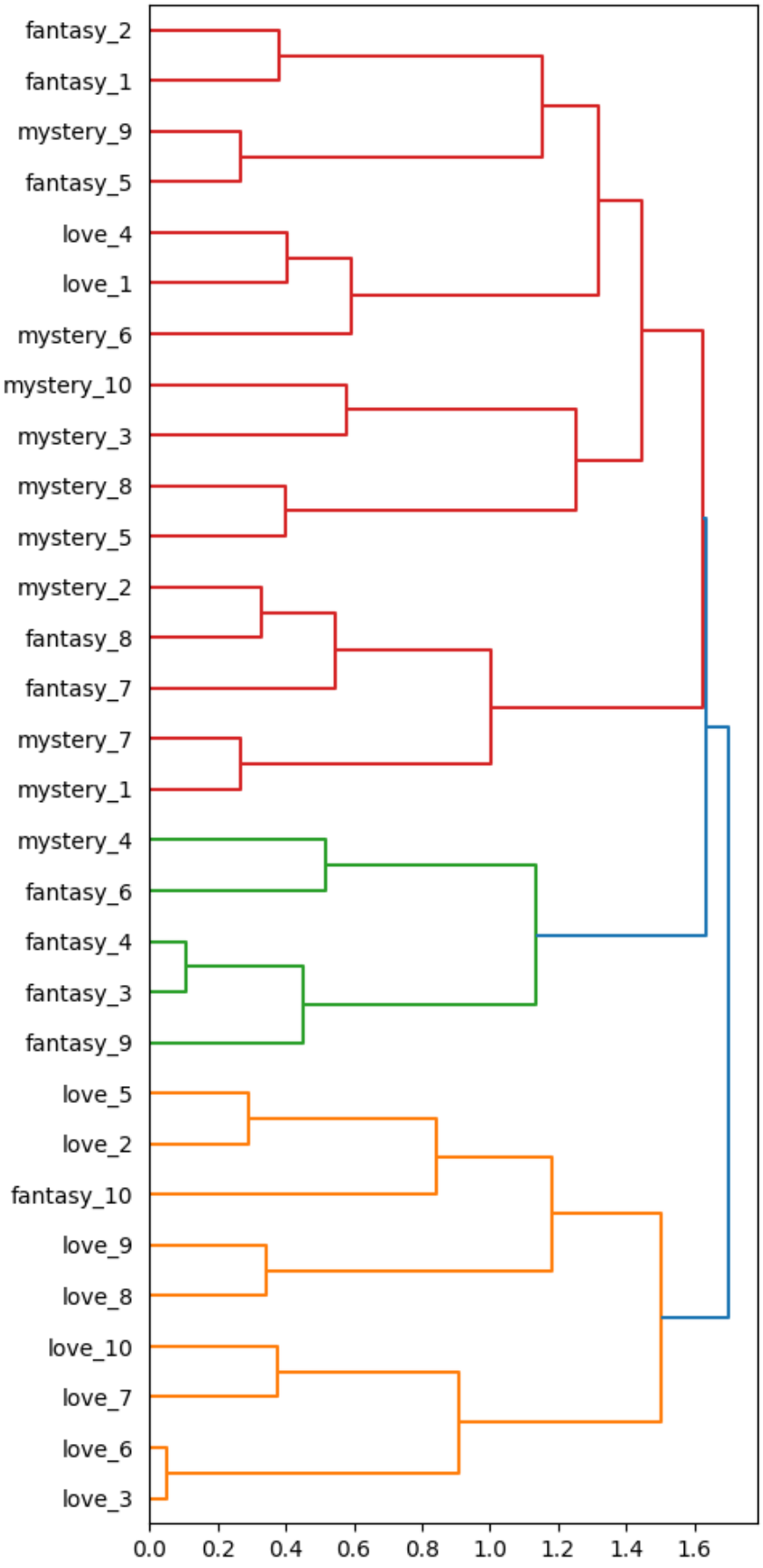

Figure 4.5 Result of dimension reduction by LSA (3genres).

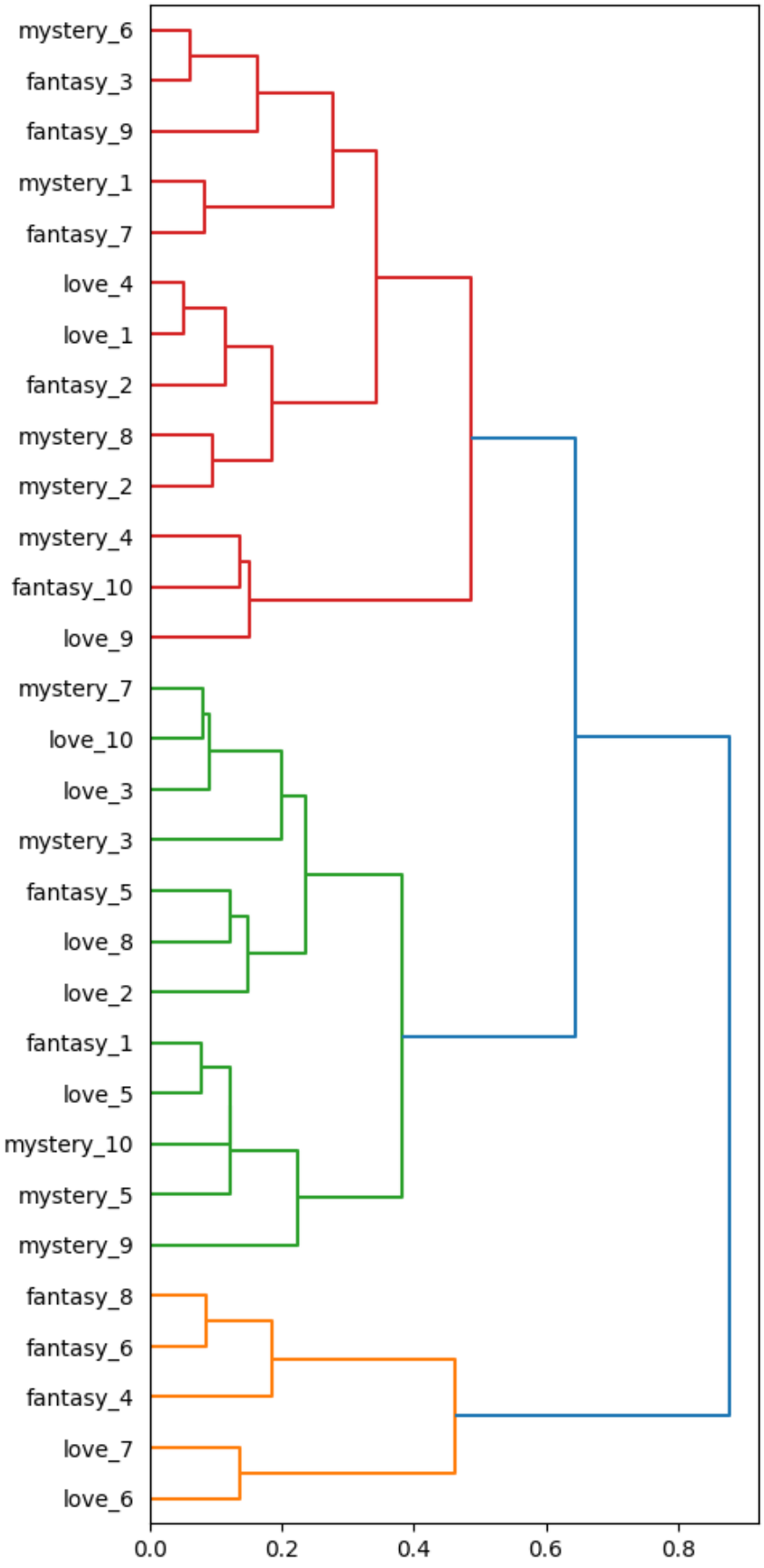

Figure 4.6 Result of dimension reduction by cluster analysis (3genres).

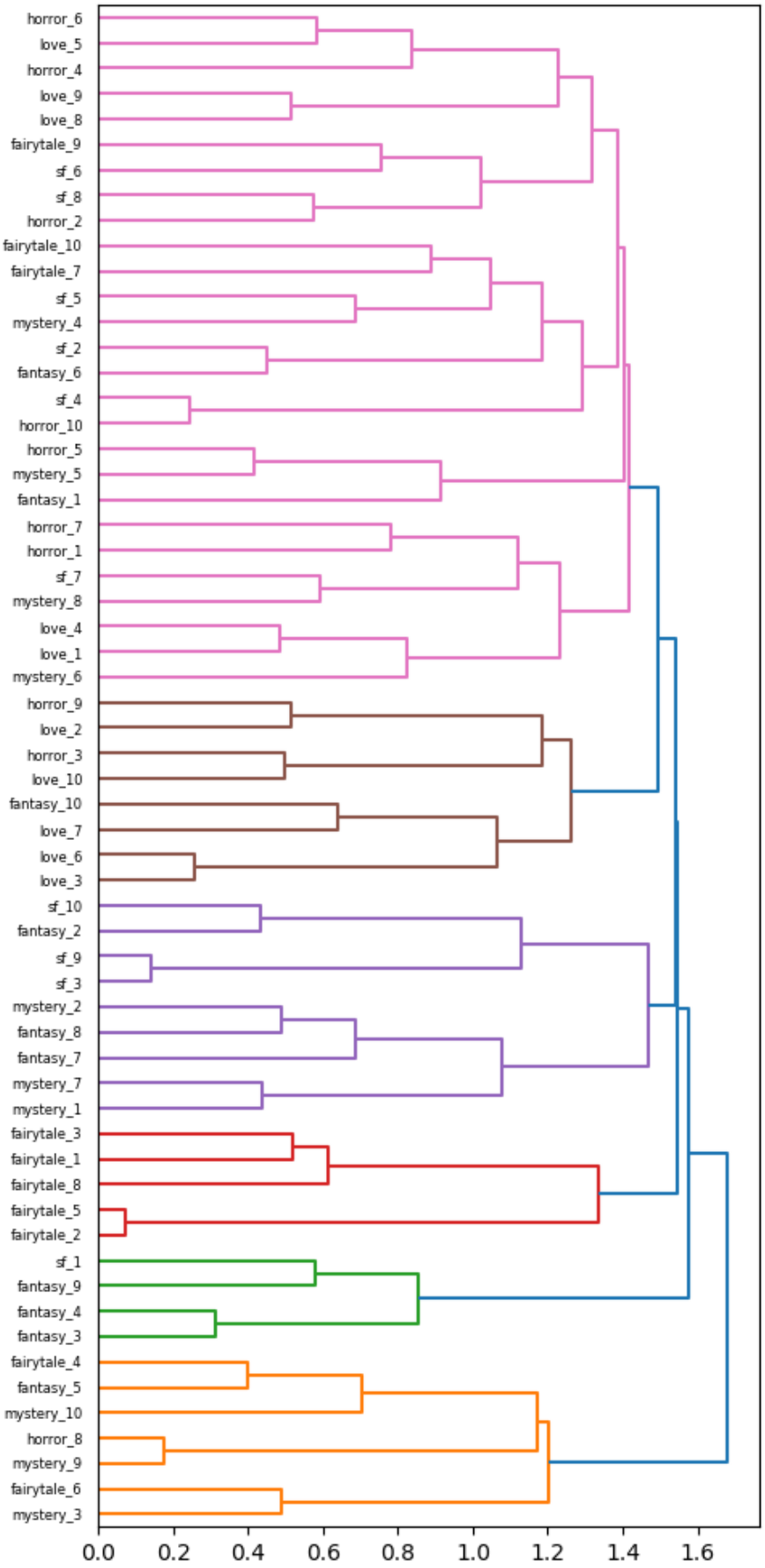

Figure 4.7 Result of dimension reduction by LSA (6genres).

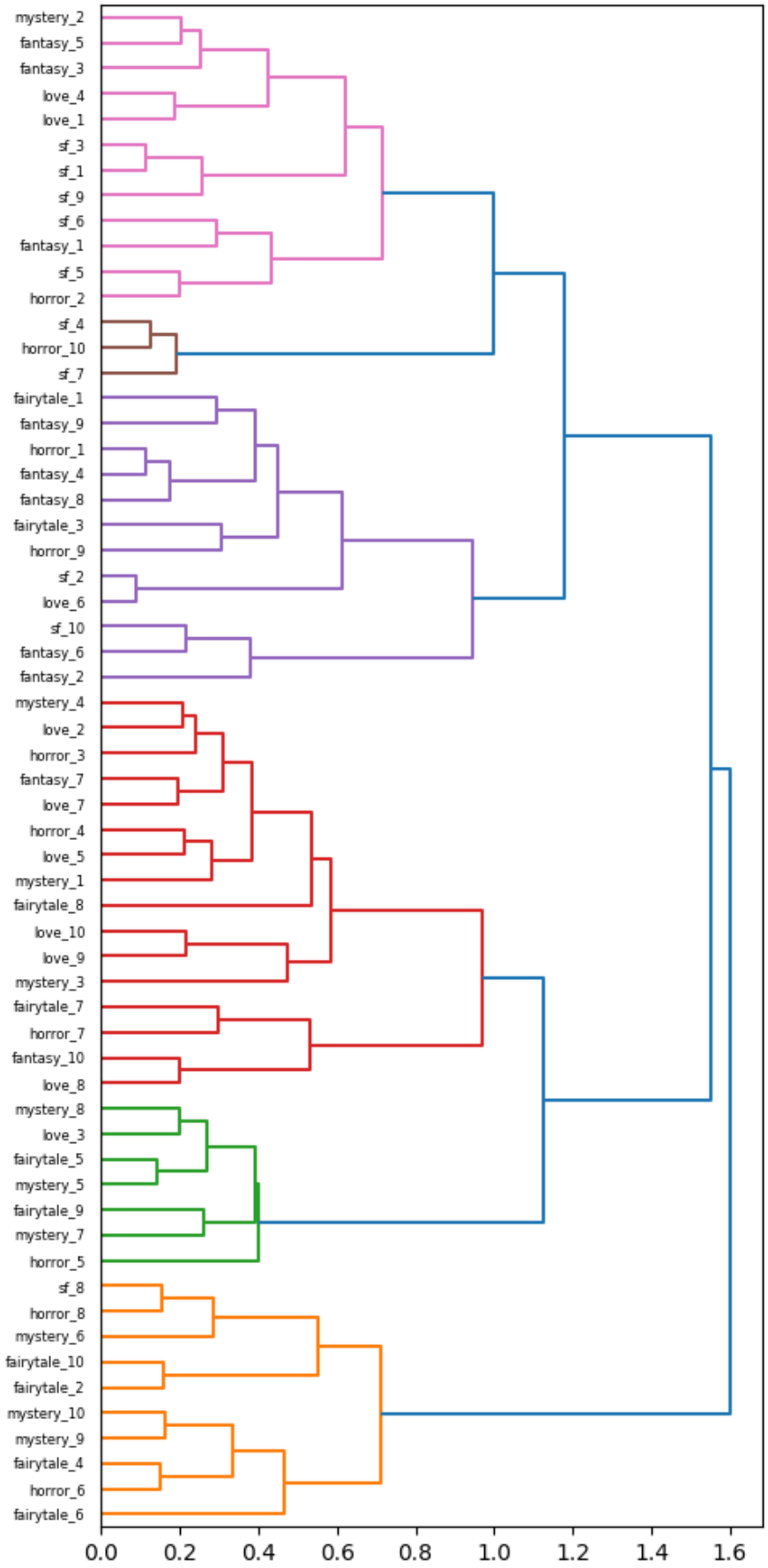

Figure 4.8 Result of dimension reduction by cluster analysis (6genres).

Table 4.2　Words representing clusters (pattern1, 50%).

cluster	**most similar word**
cluster1	笑わ
cluster2	覚え
cluster3	セーラ
cluster4	苦悩
cluster5	ええ
cluster6	seven
cluster7	簡単
cluster8	コミック
cluster9	高校生

性が高いものを同じクラスターとしてまとめていたと考えられる。

しかし、正解率だけを評価手法とすることには問題点もある。正解率で評価することができるのは割合であり、各クラスターに含まれる要素数は正解率に影響しない。つまり、クラスターの要素数に偏りがあっても、それが直接評価に影響しないということである。本研究では、ジャンルごとに均等な個数の小説のあらすじを分析対象として読み込ませた。そのため、各クラスターの要素数は均等になるのが理想である。しかし、Figure 4.7など、分析結果によっては各クラスターの要素数に偏りが生じている。そのため、正解率のみで判断するのではなく、各クラスターの要素数を比較し、均等にクラスタリングできているかなどの指標を導入する必要があると考えられる。

4.8.2　潜在的意味解析, クラスター分析による次元削減の比較

Figure 4.1〜Figure 4.4 と 4.8.1 項の考察より、クラスター分析による次元削減は作品数が増えるほど、ジャンルごとにまとまった分類をしていることがわかった。特に実験パターン 2〜4 では潜在的意味解析を上回っている。このような結果が得られたのは、分析対象である小説のあらすじの特徴が理由であると考えられる。小説のあらすじは文章はとしては短いが、それぞれに作品を象徴するための単語が含まれていた。ジャンルごとに出現する単語を調べたところ、恋愛では「彼」や「お見合い」、ファンタジーでは

「異世界」や「召喚」、推理では「事件」や「解決」といった象徴的な単語が何回か出現していた。また、単語の出現回数について調べたところ、上記のよく使われる特徴を表した語以外にあたるほとんどの単語は 1,2 回しか出現していなかった。これは、小説のあらすじによって表現方法が異なることや使用した形態素解析の辞書による影響が理由として考えられる。このように豊富な単語が出現してくるなか、Word2vec ではそれに対応する形でほとんどの単語のベクトルを取得することができた。これは Word2vec を使用した大きな強みといえる。以上のことから、クラスター分析による次元削減において、単語間のクラスターを生成する際に意味が近い多くの単語同士でクラスターが形成され、結果にも影響を及ぼしたと考えられる。

また、分析対象と手法の関係性も結果への影響を与えていると考えられる。潜在的意味解析は、単語の出現回数などから統計的に次元削減をしているため、単語自体の区別はついていない。そのため、与えられたデータに捉えたい傾向とは別の単語が多く出てくると対処することが難しいという側面がある。それに対し、クラスター分析による次元削減は、Word2vec を利用することで単語自体に区別を付けることが可能となり、単語間の関係をデータとして表現することができるという強みがある。こうした理由から、単語数が多い実験パターン 2〜4 では、クラスター分析による次元削減が潜在的意味解析の正解率を上回る結果になったのだと考えられる。

4.8.3 前年度の研究結果との比較 [10)]

本研究では、前年度に本研究室で行われた研究をなぞり実験を行った。前年度の研究との主な変更点は、式 (3.3) での計算式の変更である。本研究は前年度の研究と比べ、対象となる文書や使用した Word2vec の学習済みモデルが異なるため、数値を用いた直接的な結果の比較を行うことはできない。そのため、実験結果の傾向の比較を行う。

前年度の実験結果の正解率のグラフ、デンドログラムをみると、本研究の実験結果と似たような特徴が出現している。また、結果や考察においても、本研究と同じ傾向であることが読み取れる。このような理由より、本研究では前年度の研究と同様の傾向が得られたといえる。

本来であれば前年度の研究を再現した実験を用意し、本研究で比較すべきである。しかし、今回は研究時間の都合の関係で行えなかった。

また、本研究では式 (3.3) を用いて計算を行ったが、実験結果で前年度の研究と同様の

傾向が得られたことにより、式 (3.3) の有効性が確認できた。さらに、4.7.4 項の結果より、この式を用いる有用性についても確認をすることができた。これらの理由より、前年度で用いた式 (3.4) と本研究で導入した式 (3.3) では、本研究で導入した式 (3.3) のほうが、使用するメリットが大きいと考えられる。

第5章　結論

本研究では、小説のあらすじについて、従来手法である潜在的意味解析と本研究室で提案されたクラスター分析による次元削減を行い、類似度を計算した。これらの実験は、クラスター分析による次元削減の有効性を確認するために行った。また、クラスター分析による次元削減において、本研究より導入した式の利点の確認も行った。

まず、クラスター分析による次元削減の有効性の確認である。手順としては、初めに小説のあらすじの取得、形態素解析、Tf-Idf の計算、主成分数の決定を行った。その後、潜在的意味解析とクラスター分析による次元削減を行い、分析結果を正解率とデンドログラムで表した。なお、正解率は評価材料として妥当かどうかわかっていないため、その判定を行うことも踏まえて導入した。以上の作業を、ジャンル数や作品数などの条件を変えた 4 パターン行い、結果を比較した。

正解率の妥当性を確認したうえで、出力された結果を比較したところ、クラスター分析による次元削減の有効性を得ることができた。この結果が得られた理由として、分析対象と次元削減手法の相性が挙げられた。クラスター分析による次元削減は潜在的意味解析と比べ、単語間の関係をデータとして表現することができるという強みがある。そのため、本研究の分析対象である小説のあらすじは、その強みを活かすことができるデータであったと考えられる。それにより、良い結果が得られたのだと判断した。こうして、1 つ目の目的であったクラスター分析による次元削減の有効性の確認を行えた。

しかし、本実験を通して気になる点もいくつか浮かんできた。まず、正解率だけを評価手法とする点である。デンドログラムと正解率の関係性より、ある程度の妥当性は確認できた。しかし、クラスターの数に偏りがあった場合、正解率のみで評価していいとは言い切れない。そのため、正解率とは別の評価指標を用意し、それらを併用することでより良い評価ができるかもしれない。次に、クラスター分析による次元削減における単語間の距離の測り方である。今回は文書のクラスター分析で用いられる cos 類似度を単語のクラスター分析にも適用したが、他の測定方法については計算していない。そのため、測定方法によってはより良い結果が得られる可能性がある。最後に、文書間のクラスター分析の手法である。本研究では階層的クラスター分析で分類を行ったが、他の分析手法は試していない。そのため、他の分析手法を試してみることで、より良い結果が得られる可能性がある。

次に、クラスター分析による次元削減において、本研究より導入した式の利点の確認を行った。文書間の類似度計算で求めた各クラスタの平均ベクトルと Word2vec を用いて最も類似度が高い単語を取得したところ、各クラスタの特徴を示す単語を取得することができた。また、実験結果を前年度の研究と比較し、大きな差がないことを確認した。このような理由から、前年度の研究で用いた式よりも、本研究より導入した式のほうが使用するメリットが大きいと考えられる。こうして、2 つ目の目的であった新たに導入した式の利点の確認を行えた。

本研究では、取得した単語は結果の確認として使用した。しかし、この各クラスタの特徴を示す単語は、クラスター分析による次元削減の評価指標として用いることができるかもしれない。この単語を新たな評価手法のひとつとして導入することができれば、より良い結果が得られる可能性がある。

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並びに同クラスの皆様に厚く御礼申し上げます。

参考文献

1) 山内長承 著, Python によるテキストマイニング入門, オーム社, 2017.
2) 斎藤康毅 著, ゼロから作る Deep Learning 2—自然言語処理編, オライリージャパン, 2018.
3) 一色政彦, tf-idf（term frequency - inverse document frequency）とは？ https://atmarkit.itmedia.co.jp/ait/articles/2112/23/news028.html（2022 年 2 月 11 日アクセス）.
4) 加納学, 主成分分析, 京都大学大学院工学研究科化学工学専攻プロセスシステム工学研究室, 1997. http://manabukano.brilliant-future.net/document/text-PCA.pdf
5) 有賀康顕 中山心太 西林孝 著, 仕事で始める機械学習, オライリージャパン, 2018.
6) 水野貴明 著, Web API: TheGood Parts, オライリージャパン, 2014.
7) 株式会社ヒナプロジェクト, なろうデベロッパー. https://dev.syosetu.com/（2021 年 12 月 6 日アクセス）.
8) Toshinori Sato, Neologism dictionary based on the language resources on the Web for Mecab, 2015. https://github.com/neologd/mecab-ipadic-neologd（2021 年 11 月 25 日アクセス）.
9) 鈴木正敏, 日本語 Wikipedia エンティティベクトル. http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/（2021 年 11 月 15 日アクセス）.
10) 長谷川翔海, Word2vec を利用したクラスター分析による文書の分類, 岐阜工業高等専門学校電気情報工学科卒業研究報告, 2021. http://www.gifu-nct.ac.jp/elec/deguchi/sotsuron/hasegawa/
11) 服部修平, テキストマイニングによる文書の類似度計算に関する研究, 岐阜工業高等専門学校電気情報工学科卒業研究報告, 2017. http://www.gifu-nct.ac.jp/elec/deguchi/sotsuron/hattori/
12) 長尾彪真, 文書の類似度計算における次元削減手法に関する研究, 岐阜工業高等専門学校電気情報工学科卒業研究報告, 2019. http://www.gifu-nct.ac.jp/elec/deguchi/sotsuron/nagao/