卒 業 研 究 報 告 題 目

Word2Vecを用いた
モデルの違いによる文書分類の差

Differences in document classification
due to model differences 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

2018E27 西川　司優

令 和 0 5 年（ 2 0 2 3 年 ）　　2 月 1 6 日 提 出

Abstract

The purpose of this study is to confirm the effectiveness of dimensionality reduction and the differences on document classification between different Word2Vec models. The documents were analyzed using Word2Vec and the inter-document similarity was calculated. These experiments were conducted to confirm the effectiveness of dimensionality reduction through document clasification. Then, a comparison with latent semantic analysis was conducted to clarify the impact of the difference. First of all, the synopsis of the novel was obtained, morphological analysis was performed, TfIdf was derived, and the similarity between words was calculated by the similarity between words obtained by Word2Vec. Then, latent semantic analysis and dimensionality reduction by cluster analysis were performed, and the analysis results were expressed as a tree diagram and the accuracy. The above work with six patterns is performed, varying the number of documents and the Word2Vec models. Comparision of the results of document classification proved the effectiveness of dimensionality reduction by cluster analysis. Furthermore, by comparing the cluster analysis results applying several different Word2Vec models for the same number of documents, the higher the number of dimensions, the proportionally higher the accuracy for the Word2Vec model. These results indicate that the high dimensional Word2Vec model provides a deeper understanding of the meaning of words than the low dimensional Word2Vec model.

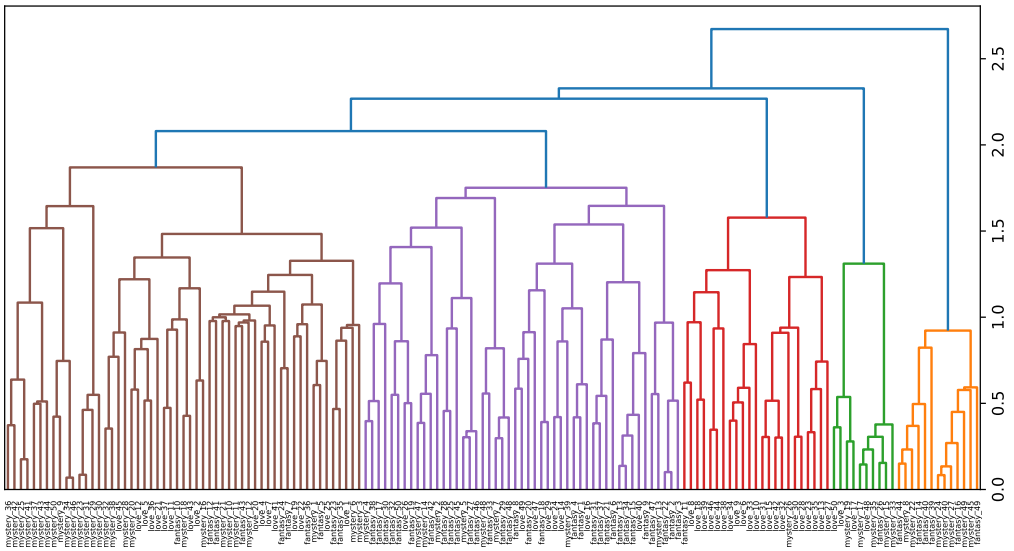

Figure 1 Example of dendrogram.

目 次

第1章　序論

私たちが暮らす現代社会は、急速な情報化により日常のあらゆるものがデータ化され管理・共有されている。そんな情報化社会に暮らす多くの個人は、タブレットやスマートフォンなどの端末を所有しており、日々の生活のなかで様々な意思決定に役立てている。すなわち現代社会においてこれまでに蓄積されてきた大量のデータは、私たちの生活を支えるライフラインの一部であり、必要不可欠なものといえる。その一方で、近年ではソーシャルネットワーキングサービスの発達により、だれでも情報発信ができることになったことで、虚偽の情報や信頼性に欠ける情報も多く出回いる。そのため、現代の情報化社会において私たちは玉石混交のデータの中から、自身にとって有益なものを見極める情報リテラシーが常に求められている。しかし、現在世界中に存在する莫大な量のデータを一個人が信頼できるものか否か見極めることは極めて困難である。

このような問題を解決するための技術の一つがデータマイニングである。データマイニングとは大量のデータを統計学やパターン認識、人工知能などの分析手法を駆使して、膨大な量のデータの中から有益なパターンやルールを見出すための技術である。データマイニングの中でも自然言語処理の技術を用いて、分析の対象を自然言語で構成されるテキストデータに限定した技術をテキストマイニングと呼ぶ。テキストマイニングは、現在様々なことに応用されている技術だが、自然言語の曖昧性ゆえに多くの課題が残っており、いまだ完成された技術ではないとされている。

本研究の目的はテキストマイニングで使用される文書分類において、文書間の類似度の計算を効率化するために用いられる、次元削除を対象としている。その中でも本実験で対象とする次元削減手法は、単語分散表現ができる Word2Vec とクラスター分析の手法を併用したものであり、これによって単語の意味を考慮した次元削減が期待される。加えて本研究では、３つの次元数が異なる Word2Vec モデルを採用する。また、類似度を計算するテキストデータの対象としては、小説のあらすじを採用した。対象のあらすじはなろう API で取得したものであり、いくつかのジャンルに分割されている。本研究においてこれを採用した理由は、文書間の類似度が高いほどに似た作品であり、また同じジャンルに属する小説である可能性が高い、よって、類似性の高い文章同士で構成されたクラスターは、同じジャンルで偏ったものになるだろうと考えられたためである。

実験では、本研究室で提案された次元削減手法を適用したテキスト間の類似度につい

て、いくつかの異なる Word2Vec の学習済みモデルで実行することで、Word2Vec モデルの次元数の差が、クラスター分析にどのような影響を及ぼすのかを、文書分類の正解率を指標として確認する。さらに、従来の統計的な次元削減手法である潜在的意味解析 (LSA) を適用したテキスト間の類似度も導出し、Word2Vec を用いた次元削除の比較対象とする。そして、それらをもとにデータの関係性を表すデンドログラムを出力し、比較する。

第2章　テキストマイニング

2.1　テキストトマイニング[1)]

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、統計学や人工知能などの様々な分析手法を駆使し、解析することによって、有益な情報や有用なパターン、意思決定に必要とされる知識をなどを特定するために用いられる手法のことである。少量のデータであれば人の手によって解析し、有益な情報やパターンを見出すことは可能である。しかし、企業が扱うビッグデータなど膨大な量のデータの中から、人間が自力で有益な情報や法則性を見出すことは不可能に等しい。

2.1.2　テキストトマイニング

テキストマイニングとは、データマイニングの中でもテキストデータを対象とした技術のことである。自然言語解析の手法を活用し、文章を名詞、動詞、形容詞などの単語に分割し、それらの出現頻度や相関関係を分析することで、テキストデータの中から有益な情報やパターンを発見することができる。また、言葉は数値と異なり意味を持っているのでる。その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く、いまだ完成された技術でないとされている。テキストデータの例として、SNS の投稿や顧客からの問い合わせ、アンケート調査の自由記述分などの文書が挙げらる。近年の多くの企業ではこれらのデータに対しテキストマイニングを行うことで、経営戦略やマーケティングに役立てている。

2.1.3　形態素

形態素とは、それ以上分解したら意味をなくさなくなるところまで分割されたまとまり、すなわち意味を持つ表現要素の最小単位である。また、形態素と単語とでは異なる場合もあり、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

形態素解析とは、自然言語処理の一部であり、自然言語で構成されたテキストデータを、文法や品詞の情報をもとに形態素に分解し、それぞれの品詞や変化などを判別することである。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。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である。日本語は「お金」の「お」のように、単独では用いられず必ず他の形態素とともに使用される拘束形態素が存在すること、英語と異なり単語同士が区切られていないため、英語などの言語と比べ分解が困難であり、膨大な辞書データが必要となる。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。

2.1.5　MeCab[2)]

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なものの一つである。また、MeCab は C 言語で作られたプログラムであり、多くの言語環境に適応している。

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。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

自然言語の曖昧さに多義性と類義性の二つが存在する。

多義性とは、ある単語が複数の意味で用いられ得ること、あるいは解釈できることである。例えば「このはしわたるべからず」という文章において、「はし」は「端」という

意味と「橋」という意味の 2 パターンが存在する。また、日本語に限らず英語においても、「race―人種/競争」や「capital―首都・資産」のように、同一の単語が複数の意味を持つ事例が多数存在する。

類義性とは、語形は異なっているが、意味の似かよっている二つ以上の単語が存在することである。例えば、「形見」と「遺品」は読み書きが全く異なる単語同士だが似た意味を持つ。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。

第3章　実験で使用した技術・手法

3.1　計算手法

3.1.1　Tf-Idf[3)]

TF-IDF とは、各文章に含まれる特定の喃語が、その分全体でどのくらい重要かを表す統計的尺度である。TF-IDF は TF と IDF との積で表される。

TF とは TermsofFrequency の略称であり、文書中の単語の出現頻度を表す。TF は文書中の出現頻度が多いほど大きな値をとり、その単語は文書とのかかわりが深いものであると考えることができる。

IDF とは InverseDocumentFrequency の略称で、文書全体における単語の偏りを考慮する値である。IDF は逆文書頻度、すなわち特定の単語の文書出現頻度の逆数であり、その単語の希少性を表す。ある文書にしか出現しない単語は、その文書を特徴づけるものであると考えられ、IDF は大きな値をとる。一方で、複数の文章に頻繁に出現する単語は、希少性が低く、出現するいづれの文章に対しても、それらを特徴づける単語ではないことを意味し、IDF は小さい値となる。TF-IDF は、これら二つの積となる。TF と IDF には様々な導出方法が存在するが、本実験では以下のような式で計算した。IDF は、数値が 0 にならないよう、分母分子に 1 を足している。また対数計算結果が 0 となってしまわないように、全体に 1 を足している。

$$Tf_{ij} = \text{文書 } d_i \text{ における } t_{ij} \text{ の出現回数} \tag{3.1}$$

$$Idf_{ij} = \log \frac{\text{全文書数} + 1}{\text{単語 } t_{ij} \text{ が出現する文書数} + 1} + 1 \tag{3.2}$$

$$TfIdf_{ij} = Tf_{ijij} \tag{3.3}$$

3.1.2　cos 類似度

cos 類似度とは、ベクトル空間モデルにおいて、2 つのベクトルの類似性を表す指標であり。ベクトル間の cos 値を求めることによって、ベクトル同士がどの程度同じ方向を向いているのかを求めることが可能となる。cos 値が 1 に近いほどベクトル同士が挟む角は

小さくなり、同じ方向を向いているということになる。一方で、cos 値が -1 に近い値になるほど、ベクトル同士を挟む角は大きくなり、逆の方向を向いていることになる。本研究では、ベクトル文書同士を比較するための類似度計算手法として使用した。

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 類似度は以下の式で導出される。

$$\cos(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|} \tag{3.4}$$

3.1.3 主成分分析 [4)]

実験や調査において、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、人の目による確認も容易にできないパターンがある。こういった事態を解決する技術として、主成分分析 (principalcomponentanalysis;PCA) が存在する。主成分分析とは、ベクトルといった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。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ようなことを言う。これにより、データ間の比較や可視化が容易となる。

具体的な方法については、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、$N(N)$ 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。

$$z_n = \sum_{p=1}^{P} a_{pn} x_p \tag{3.5}$$

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。

$$\sum_{p=1}^{P} a_{pn}^2 = 1 \qquad () \tag{3.6}$$

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 3.1 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 (3.5) 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 という軸をとる。これが第 2 主成分となり、第 1 主成分にて表せ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 3.1 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、その利点が大きく表れるようになる。

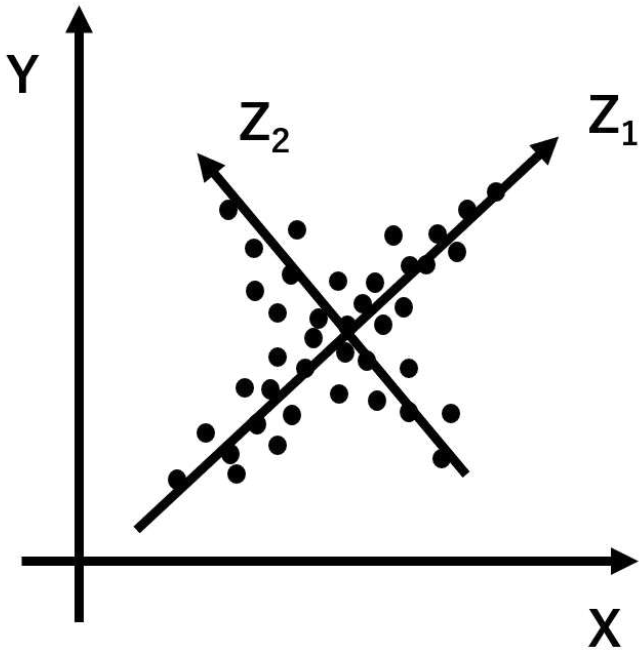

Figure 3.1　Exampleoftwo-dimensionaldata.

3.1.4　主成分数の選択

主成分分析において、主成分数の設定は極めて重要な問題となる。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、主成分数の決定には以下に示す 3 通りの方法が存在する。

- 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。

累積寄与率とは主成分の寄与率を合計した値を言う。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ものである。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。

$$P_n = \frac{\lambda_n}{\sum_{p=1}^{P} \lambda_p} \tag{3.7}$$

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。

$$C_n = \sum_{i=1}^{N} P_i \tag{3.8}$$

3.1.5 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 (3.9) で表される。

$$TD = \left(\begin{array}{c|cccc} Term & doc_1 & doc_2 & \cdots & doc_N \\ \hline w_1 & I_{w_1,doc_1} & I_{w_1,doc_2} & \cdots & I_{w_1,doc_N} \\ w_2 & I_{w_2,doc_1} & I_{w_2,doc_2} & \cdots & I_{w_2,doc_N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_M & I_{w_M,doc_1} & I_{w_M,doc_2} & \cdots & I_{w_M,doc_N} \end{array}\right) \tag{3.9}$$

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 (LatentSemanticAnalysis;LSA) が用いられる。潜在的意味解析は、文書データには潜在的なトピックが存在すると推定し、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 (singularvaluedecomposition;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。

$$TD = U^T \tag{3.10}$$

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 (3.11) のように計算することで、必要な情報のみを抜き出した行列を生成することが可能となる。

$$TD_k = U_k^T TD \tag{3.11}$$

3.1.6 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 3.2 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。

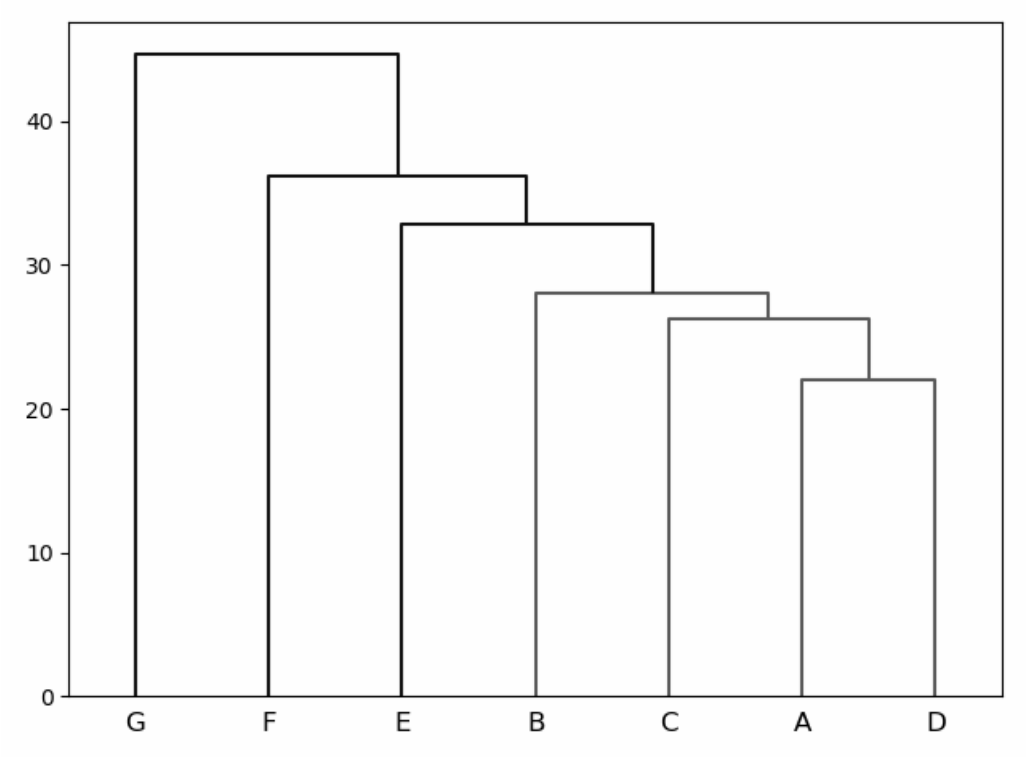

Figure 3.2　Dendrogram.

3.1.7　正解率

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の 1 つとして導入した。分類問題の評価指標には正解率のほかに適合率、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、分類の結果をまとめた混同行列を用いて計算される。以下に混同行列の具体例を交えて正解率の導出方法について説明する。Figure 3.3 は 3 クラス分類における混同行列の具体的な例である。混同行列では各列が予測されたラベルを、各行が真のラベルを表す。そのため行列の対角成分にある値が、各ラベルにおいて正しく予測がなされたものとなる。この正しく予測がなされた値と全体のデータ数により、正解率は式 (3.12) のように計算される。

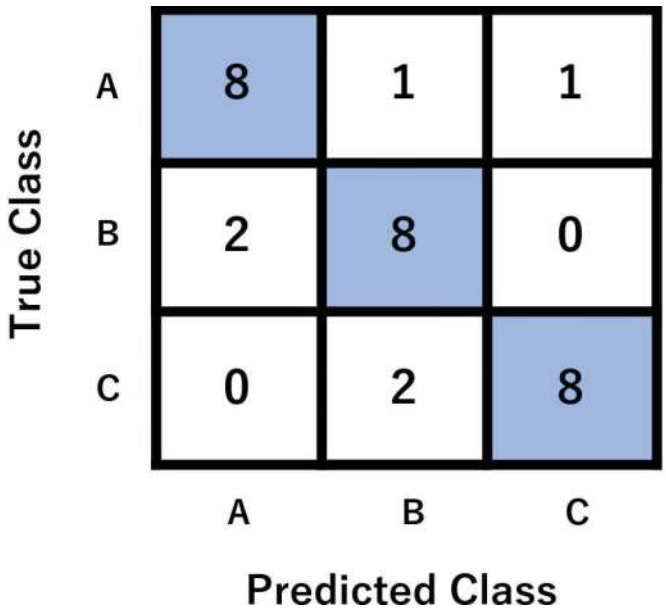

Figure 3.3　Confusion Matrix.

$$\text{正解率} = \frac{\text{正解した数}}{\text{予測した全データ数}} \tag{3.12}$$

実際に例の混同行列で計算すると、対角成分にある正しく予測された値の合計値 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 3.12 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。

3.2 WebAPI

3.2.1 API

AIP とは ApplicationProgrammingInterface の略称であり、ソフトウェアやプログラム、Web サービス間をつなぐインターフェースのことである。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。

3.2.2 WebAPI

WebAPI ではない API の多くは、利用者側が用いるプログラミング言語と同一の言語を用いて提供されている。しかし、WebAPI では、言語が異なっていても通信が可能な HTTP/HTTPS 方式が採用されている。

本研究で用いた WebAPI も、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 URI 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。

有名な例として、Google が提供する GoogleMapAPI や Amazon の AmazonMWSAPI-API、YouTube が提供する YouTubeAPI などが挙げられる。WebAPI を公開することには新たな機能やサービス開発を助ける、より良い顧客体験を提供する、自社サービスのユーザーを拡大するといった機能が存在する。そのため近年では WebAPI を導入する動きが進んでおり、各種 SNS や EC サイト等で幅広く活用されている。

3.2.3 WebAPI を利用したテキストデータの取得 [5)]

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、小説の評価などが挙げられる。本研究では、3 つのジャンルから小説タイトルとあらすじを対象として作品情報の取得を行った。

3.3 Python

3.3.1 Python[6)]

Python とは、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。1991 年に初のリリースがされ、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。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、対話的な言語
- 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営されている
- コードの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。

このような特徴から Python は、アプリケーションの開発、人工知能、データ解析、専門的なライブラリなど、人工知能をはじめとした様々な用途に使用されており、日本国内に限らず世界中に多くのユーザーから支持を集めている。

3.3.2 MeCab

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 IPA を導入した。

3.3.3 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、実行を行う。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。

3.3.4 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 TfIdf 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 (3.1)、(3.2) の通りに TfIdf が導出される。また TfIdf 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

3.3.5 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 という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。

$$\cos\left(\vec{a},\vec{b}\right)_{distance} = 1 - \frac{\vec{a}\cdot\vec{b}}{|\vec{a}||\vec{b}|} \tag{3.13}$$

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。

3.3.6 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。

3.3.7 潜在的意味解析

Python では、numpy という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 TfIdf 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。そうして出力された左特異（ターム）ベクトルと式を用いて潜在的意味解析を行った。

3.3.8　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 関数を使用すれば結果が出力される。

3.3.9　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。また混同行列は、同じく scikit-learn ライブラリの confusion_matrix 関数に classification_report 関数と同じデータを与えることで生成できる。

3.3.10　scikit-learn[7)]

scikit-learn は Python のオープンソース機械学習ライブラリである。分類、回帰、クラスタリングなどのアルゴリズムモデルやサンプルのデータセットを備えており、容易に機械学習を試すことができる。本研究では TF-IDF、主成分分析、潜在意味解析などを Python で実装するために使用した。

3.3.11　Gensim

Gensim は自然言語処理に用いられる 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するために作られたライブラリで、他に Word2vec のような Wordembedding 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。

3.3.12　SciPy

SciPy は高度な科学技術計算をおこなうことが可能な Python のライブラリである。微積分や統計などの計算が可能で、機械学習やデータ分析などに使われる。

3.4 Word2vec

3.4.1 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」、「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。

$$「\text{king}」 - 「\text{man}」 + 「\text{woman}」 = 「\text{queen}」 \tag{3.14}$$

式 3.14 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。

3.4.2 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。その操作の中に、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で判断することができる。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基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関係を導出した。

3.4.3 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。ここで、Word2vec にお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、意味が似通った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。

TfIdf を重要度とした式 3.9 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 3.15 で表されるクラスターと名詞の行列と式 3.9 にある元の文書行列都の積で求められる。

$$CW = \left(\begin{array}{c|cccc} Term & w_1 & w_2 & \cdots & w_M \\ \hline C_1 & a_{1,1} & a_{1,2} & \cdots & a_{1,M} \\ C_2 & a_{2,1} & a_{2,2} & \cdots & a_{2,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_N & a_{N,1} & a_{N,2} & \cdots & a_{N,M} \end{array}\right) \tag{3.15}$$

各クラスターを $C_i(i = 1, 2, \ldots, N)$ 単語を $w_j(j = 1, 2, \ldots, M)$ とする。このとき、式 3.15 にある要素 $a_{i,j}$ は次の式で与えられる。

$$a_{i,j} = \begin{cases} \dfrac{1}{|C_i|} \displaystyle\sum_{k_i} S_{k,j} & (j_i) \\ 0 & (j_i) \end{cases} \tag{3.16}$$

第4章　実験

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幾度か変更し、複数の結果を出力する。その後、出力された類似度を基に文書分類を行い、テキスト間の関係を比較する。比較の材料には、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、正解率の評価材料としての妥当性は、過去の研究によりおおむね証明されている。この比較により、Word2Vec のモデルによるクラスター分析の違いを、文書分類の正解率をもとに確認すること、および、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、Word2Vec モデルによるクラスター分析結果の違いを確認するため、100 次元、200 次元、300 次元の異なるモデルの Word2Vec を用意した。さらに、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、分析対象である小説のあらすじのジャンル数と作品数2パターンに変化させ、結果を出力した。以下が各実験パターンの詳細である。

実験パターン1 ・ジャンル数:3 作品数:30(Word2Vec100 次元)

実験パターン2 ・ジャンル数:3 作品数:30(Word2Vec200 次元)

実験パターン3 ・ジャンル数:3 作品数:30(Word2Vec300 次元)

実験パターン4 ・ジャンル数:3 作品数:150(Word2Vec100 次元)

実験パターン5 ・ジャンル数:3 作品数:150(Word2Vec200 次元)

実験パターン6 ・ジャンル数:3 作品数:150(Word2Vec300 次元)

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6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。

- Python,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、homebrew を使用し、バージョン 0.996 をインストールした。

4.2.3 Word2vec の学習済みモデルの取得

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多大な時間を要する。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、2019 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、3.2.3 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である。作品ごとに、作品名とあらすじを同じテキストファイルにまとめることで解析対象の 1 つとした。

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 3 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〕

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 3 ジャンルの作品は、ジャンルごとに 50 作品、

計 150 作品取得した。

取得した作品は、4.1 節で述べた 6 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。本研究ではすべてのパターンで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 つを採用している。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とりわけ異なっており、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。また、本研究では過去実験のように、作品のジャンル数を変化させなかった。これは過去 2 年の研究において、ジャンル数はクラスター分析結果に大した影響を与えないことが判明しているからである。

実験パターン 1,3,5 では作品数が各ジャンル 10 作品となっている。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 2,4,6 では選択した各ジャンルにおいて、取得した 50 作品ずつ含めたものということになる。以上の通りに、6 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 1,2,3 (単語種類数) … 1364 語
- 実験パターン 4,5,6 (単語種類数) … 4409 語

4.3.2 TfI-df の計算

Python を用いて Tf-Idf を計算する。計算は、3.3.4 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

Table 4.1　Dimensional quantity of each pattern.

cumulative contribution ratio	pattern 1	pattern 2	pattern 3	pattern 4
50%	9	40	15	70
55%	10	47	18	80
60%	12	53	21	95
65%	13	60	24	110
70%	15	68	27	125
75%	17	77	30	140
80%	19	87	35	160
85%	21	100	39	185
90%	23	110	43	210

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。3.3.6 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 7 回次元削減を行うことになる。通常は 3.1.4 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

3.3.7 項に記述があるように、numpy ライブラリの svd 関数に TfIdf 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をもとに指定した。

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
2. 抽出した単語のベクトルを基に、式 (3.4) 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（3.16）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のラベル部分を抜くことで対応した。

- 実験パターン 1,2,3 (Word2vec 非登録語) … 77 語
- 実験パターン 4,5,6 (Word2vec 非登録語) … 273 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。

4.5　文書の分類

4.5.1　文書間の cos 類似度

次元削減された行列に対して、3.3.5 項で記述したように cosinesimilarity 関数と pdist 関数を用いることで 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基にクラスター分析を行った。3.3.8 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 3.1.6 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

3.3.8 項にあるように、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不可能であったため出力は行わなかった。

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分までクラスターの分割を行う。取得したあらすじが 3 ジャンルであるため、3 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。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分に分割されたクラスターに対して、3.3.9 項で述べた各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、適当なジャンルを割り当てる。_matrix 関数で、割り当てられたジャンルに基づく混同行列を生成する _report 関数で正解率を求める
2. 1. とは別のジャンルを各クラスターに割り当て、2.3. 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
3. 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。内容は縦軸が正解率、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。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 正解率の推移

実験パターン 1,2,3 の正解率を Figure 4.1、実験パターン 4,5,6 の正解率を Figure 4.2 に示す。両グラフを比較すると、作品数の多い Figure 4.2 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、より作品数が増えるほど全体の正解率もやや低下する傾向が読み取れる。

これらのパターン同士を比較すると、多少のズレは生じているものの、Word2Vec モデルの次元数が高い物ほど、高い正解率となっていることがわかる。また、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。

4.7.2 各 Word2Vec モデルによるクラスター分析結果の正解率

まず、作品数が 30 の時の文書分類結果について出力する。Table 4.2 より、300 次元のモデルによる文書分類の正解率は、100 次元のモデルと比べ 3.7%、200 次元のモデルと比べ 4.8%、上回る結果となった。しかし、実験パターン 1 と実験パターン 2 の比較では、Word2Vec モデルの次元数が低いパターン 1 が、実験パターン 2 の正解率を 1% ほど上回る結果となった。次に、作品数が 150 の時の文書分類結果について出力する。Figure 4.2 より、300 次元の Word2Vec モデルを採用した、実験パターン 6 の文書分類の正解率は、100 次元の Word2Vec モデルを採用した実験パターン 4 の正解率を 14%、200 次元の Word2Vec モデルを採用した実験パターン 5 の正解率を 6.4%上回る結果となった。実験パターン 4 と実験パターン 5 を比較しても、次元数の高いモデルを採用している実験パターン 5 の正解率が、実験パターン 4 の正解率を 7.9%も上回る結果となった。

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

実験パターン 1,2 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 6 つのデンドログラムを生成することとなる。

まず 100 次元の Word2Vec につて調べる。実験パターン 1 では、累積寄与率 80%にお

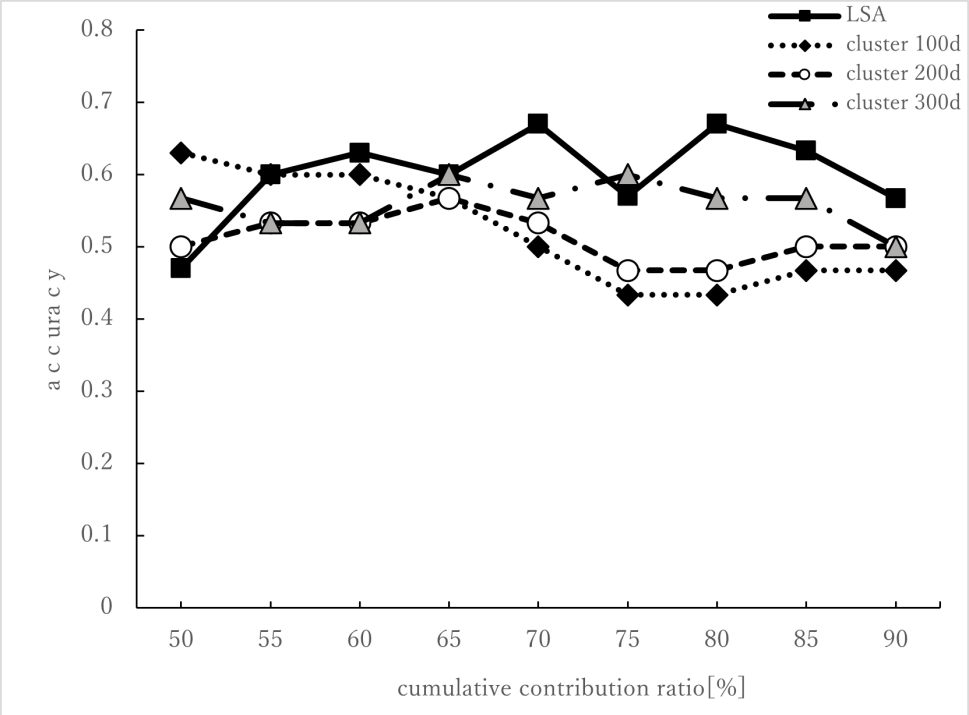

Figure 4.1　Accuracy of 3genres and 30 documants.

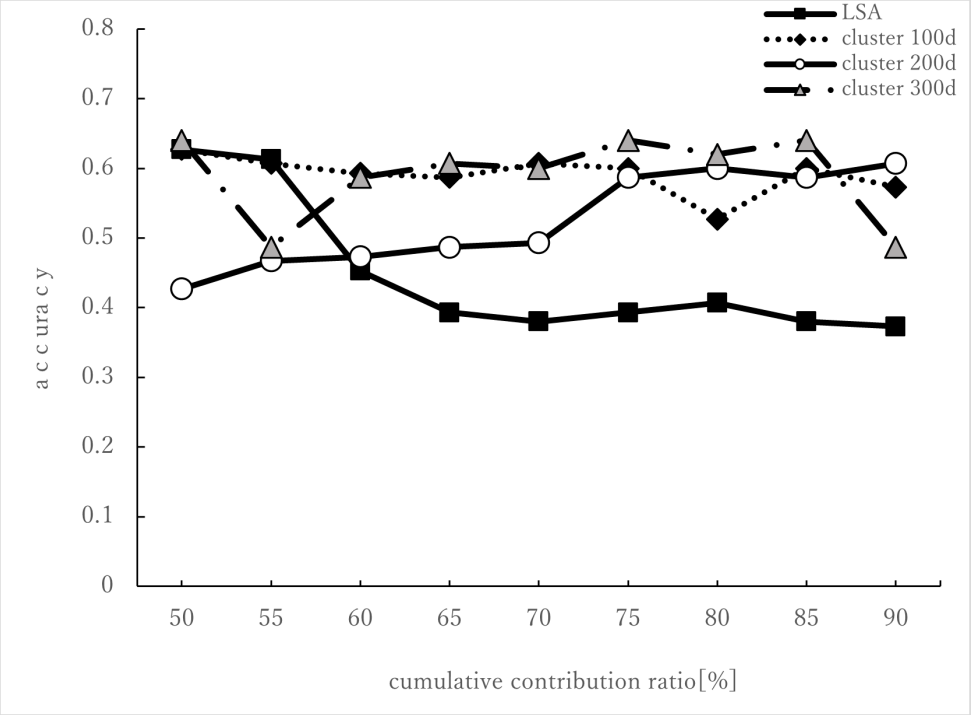

Figure 4.2　Accuracy of 3genresand 150 documents.

いて、潜在的意味解析による分類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 4 では、累積寄与率 70%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のものよ

Table 4.2　Average accuracy of each dimension and LSA.

	50works	**150works**
LSA	0.6011	0.4466
cluster100d	0.5219	0.4463
cluster200d	0.5111	0.5253
cluster300d	0.5593	0.5898

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。

次に 200 次元の Word2Vec につて調べる。実験パターン 2 では、累積寄与率 80%において、潜在的意味解析による分類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 5 では、累積寄与率 90%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。

最後に 300 次元の Word2Vec につて調べる。実験パターン 3 では、累積寄与率 80%と 70%において、潜在的意味解析による分類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 6 では、累積寄与率 75%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。

以上のことから、実験パターン 1 の累積寄与率 80%における各手法のデンドログラム、実験パターン 2 の累積寄与率 80%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 80%における各手法のデンドログラム、実験パターン 4 の累積寄与率 70%における各手法のデンドログラム、実験パターン 5 の累積寄与率 90%における各手法のデンドログラム、実験パターン 6 の累積寄与率 75%における各手法のデンドログラムの計 6 つをそれぞれ出力する。その結果、Figure 4.3、Figure 4.4、Figure 4.5、Figure 4.6、Figure 4.7、Figure 4.8、Figure 4.9、Figure 4.10、Figure 4.11、Figure 4.12 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、100 次元の場合について行う。はじめに、Figure 4.3 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、下部において恋愛のクラスターができていることが分かる。一方で、上部においては、ミステリーとファンタジーが合わさったクラスターができていることが分かる。次に、Figure 4.4 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。Figure 4.4 では 2,3 個の要素からなるクラスターにはジャンルごとのまとまりが見られるが、それより大きなクラスターになるとジャンルが混合してしまっていることが分かる。このことより、実験パターン 1 の累積寄与率 80%では潜在的意味解析の方がこちらの意図した通りに分類を行えている。そして、実験パターン 4 について比較を行う。Figure 4.5 の潜在的意味解析による結果を確認すると、下部では恋愛のクラスターが、上部ではミステリーのクラスターが形成されていることが分かる。それに対して、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では下部のクラスターでファンタジーと恋愛がまとまっている。

次に、200 次元の場合について行う。初めに、実験パターン 2 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。Figure 4.7 をみると、上部では恋愛のクラスターが形成されている。一方で、下部のクラスターに関しては、2、3 個のクラスターにはジャンルのまとまりが見られるものの、全体としては 3 つのジャンルが混合してしまっている。このことより、200 次元の Word2Vec によるクラスタリングと潜在的意味解析には大きな差がないことが分かった。

次に実験パターン 5 について比較を行う。Figure 4.8 の実験パターン 5 における潜在的意味解析による結果を確認すると、違うジャンルの要素が少し混ざってはいるものの、上部ではファンタジー、中部では恋愛、下部ではミステリーのクラスターができていることが分かる。それに対して、Figure 4.9 のクラスター分析による次元削減のデンドログラムでは上部のクラスターでは上部には恋愛のクラスターが見られるものの、それより下はファンタジーとミステリーが混合してしまっている。ただし、6,7 個ほどの要素からなるクラスターにおいてはジャンルごとのまとまりが見られた。

最後に 300 次元の場合について行う。初めに、実験パターン 3 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。Figure 4.10 より、上部では、恋愛のクラスターができていることが分かる。下部ではミステリーのクラスターが形成されている。このことより、300 次元の Word2Vec によるクラスタリングと潜在的意味解析によるクラスタリングでは、同程度の精度であることが考えられる。

次に実験パターン 6 について比較を行う。Figure 4.11 の実験パターン 2 における潜在的意味解析による結果を確認すると、2,3 この要素からなるクラスターにはジャンルごとのまとまりが見られるものの、全体としては、3 つのジャンルが混合してしまっている。それに対して、Figure 4.12 のクラスター分析による次元削減のデンドログラムでは

上部のクラスターではミステリーがまとまり、下部のクラスターではファンタジーがまとまっている。

4.8 考察

4.8.1 Word2Vec モデルの次元数が文書分類結果に及ぼす影響

まず、4.7.2 項の結果より、Word2Vec の次元数は高くなるほど正解率も高くなることが判明した。しかし、その誤差は数パーセントであり、特に大きな差がなかったともいえる。これは、私たち人間が普段単語を識別するとき、その単語に対し数百の意味付けを意識していないように、Word2Vec を用いたクラスタリングにおいても、100 次元は十分な次元数であると考えられる。ただし、4.7.3 項にあるように、作品数が 30 と少ない、実験パターン 1、2、3 では、クラスター分析による正解率は、潜在的意味解析のそれと比べ 5%〜9%程の差が生じていたため、作品数が少ない時は、クラスター分析による次元削減の精度は低くなると考えられる。一方で作品数が 150 の実験パターン 4、5、6 においては、Word2Vec モデルの次元数と、正解率とに正の相関関係が生じた。さらに、実験パターン 5、6 では潜在的意味解析による正解率を、クラスター分析による正解率が大きく上回っている。また、実験パターン 4 の正解率と潜在的意味解析の正解率を比較しても、0.02%の差しか生じなかった。

次に、潜在的意味解析, とクラスター分析による次元削減を比較する。Figure4.1〜Figure4.4 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確認できた。特に実験パターン 5，6 では潜在的意味解析を上回っていることがわかる。

このような結果となった理由として、潜在的意味解析には、単語の出現回数などから統計的な処理で次元削減をしており、単語自体の区別はついていないため、テキスト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できないという側面がある一方、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の区別を付けることが可能となり、単語間の関係をデータに付加できるという強みがあることが挙げられる。このことから、単語自体の意味をどれほど把握しているかがクラスター分析による次元削減の正確性に影響を与えるといえる。以上のような理由から、実験パターン 4、5、6 では、Word2Vec モデルの次元数と正解率とに正の相関関係が表れたと考えられる。

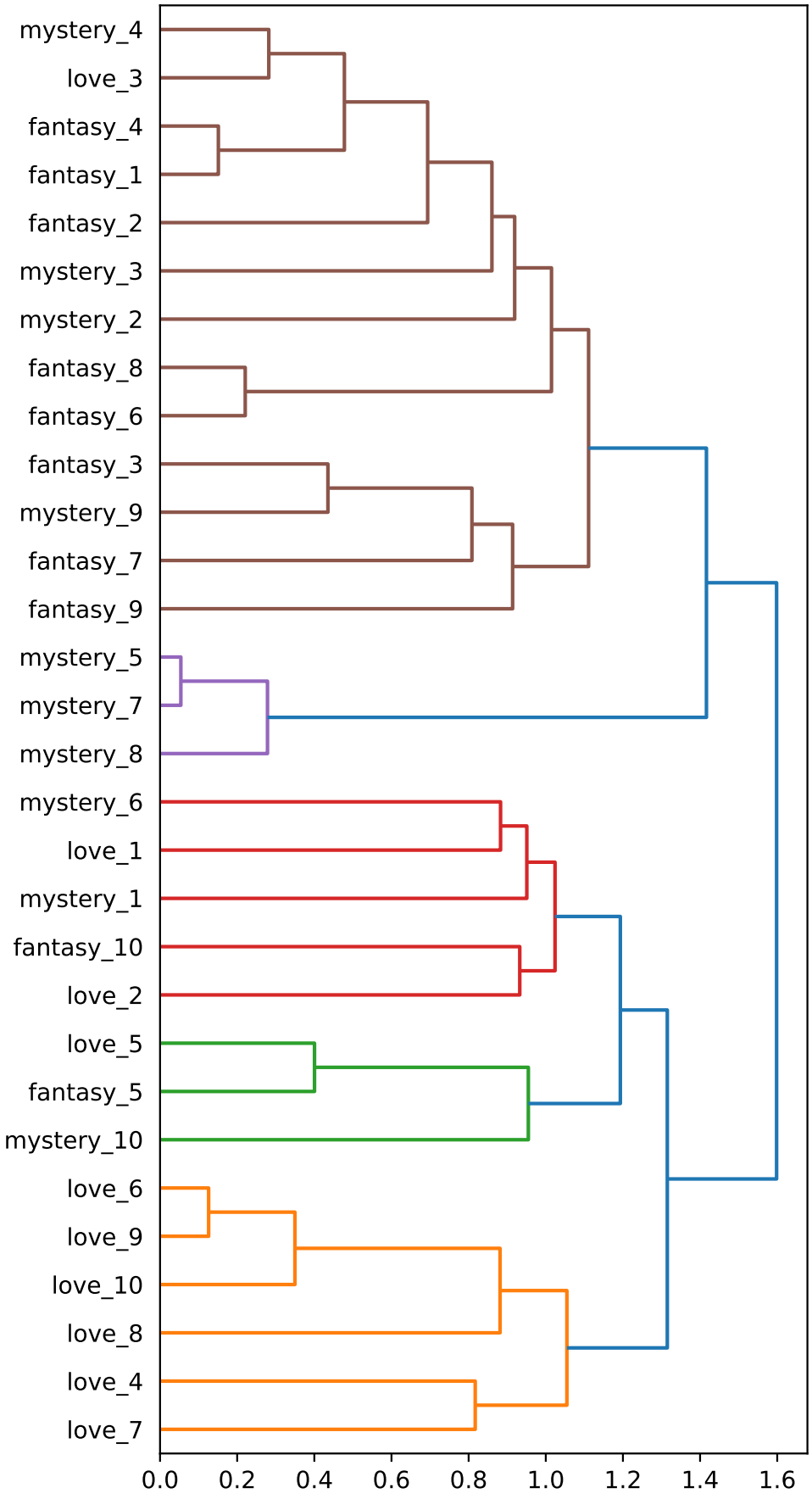

Figure 4.3 ResultofdimensionreductionbyLSA(3_30_80%).

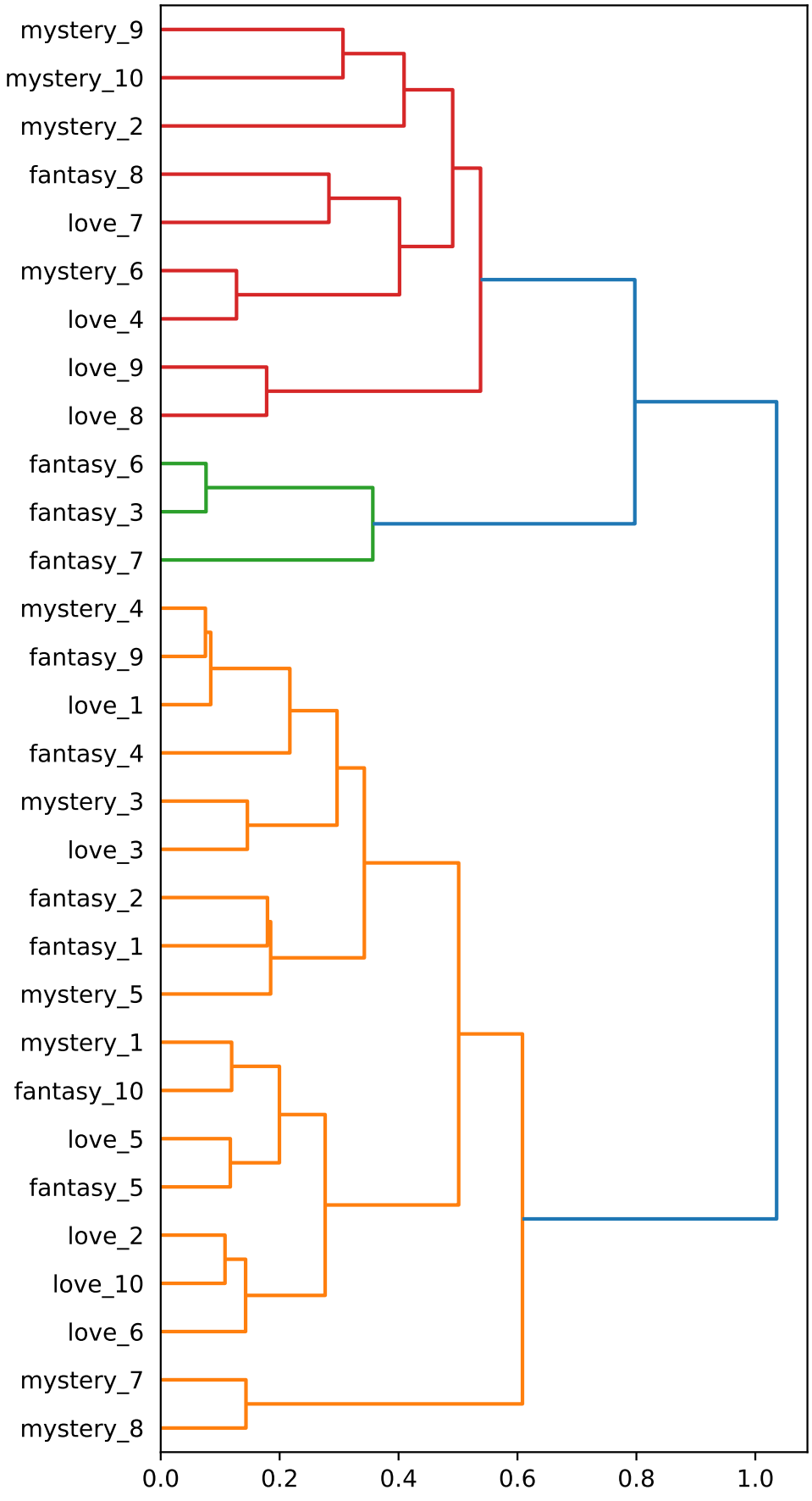

Figure 4.4 Resultofdimensionreductionbyclusteranalysis(3_30_100d_80%).

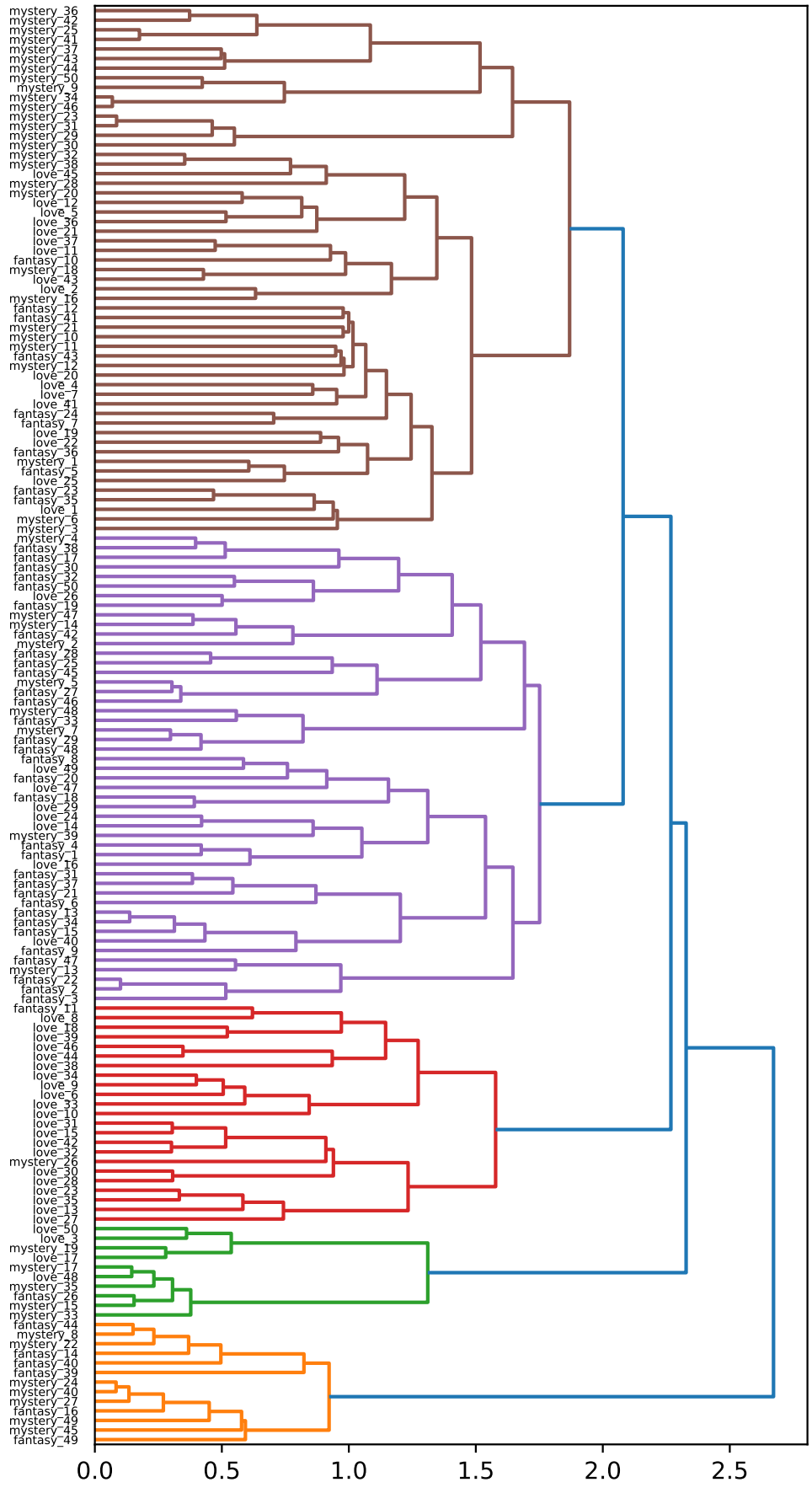

Figure 4.5 ResultofdimensionreductionbyLSA(3_150_70%).

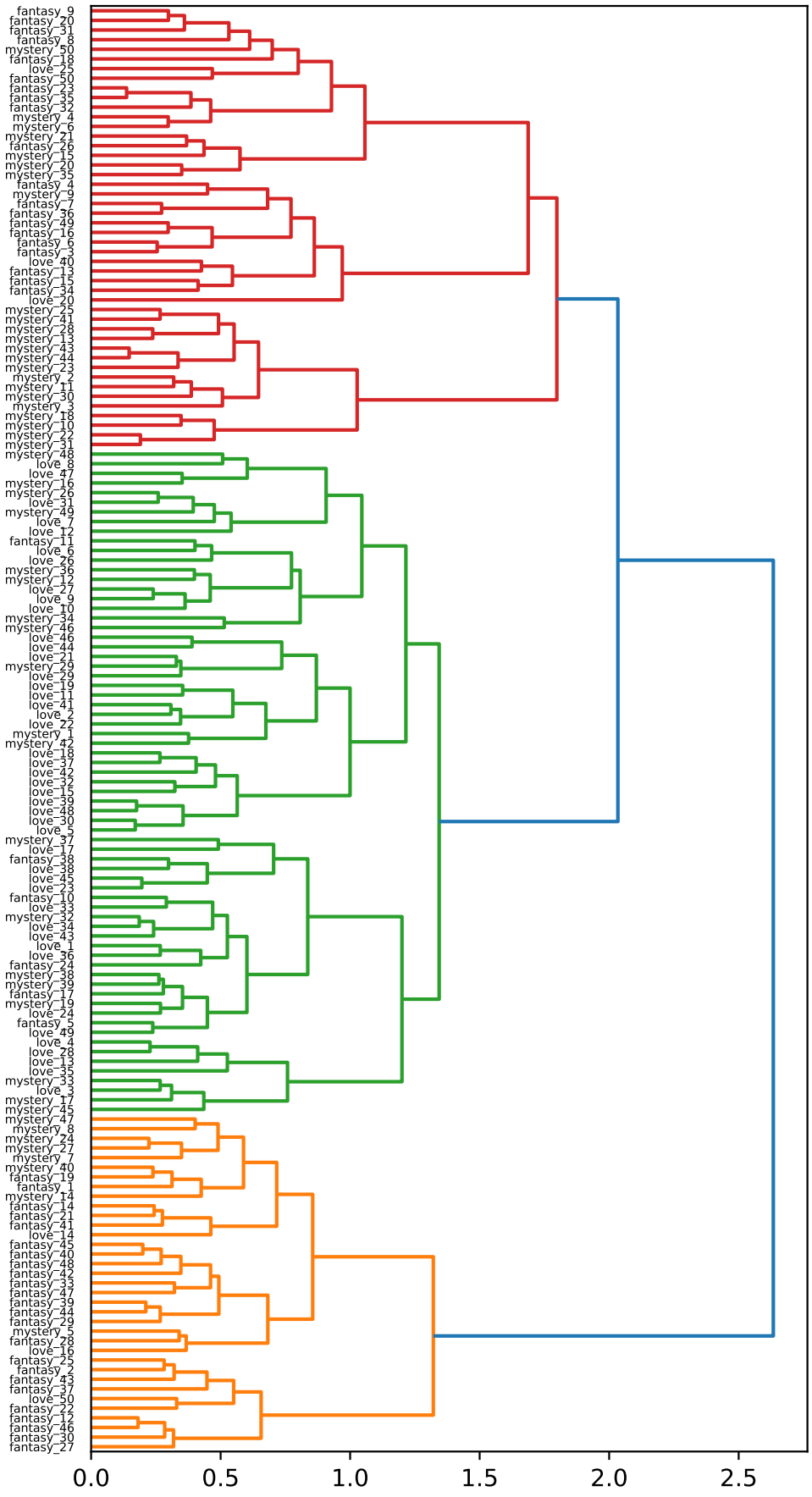

Figure 4.6 Resultofdimensionreductionbyclusteranalysis(3_150_100d_70%).

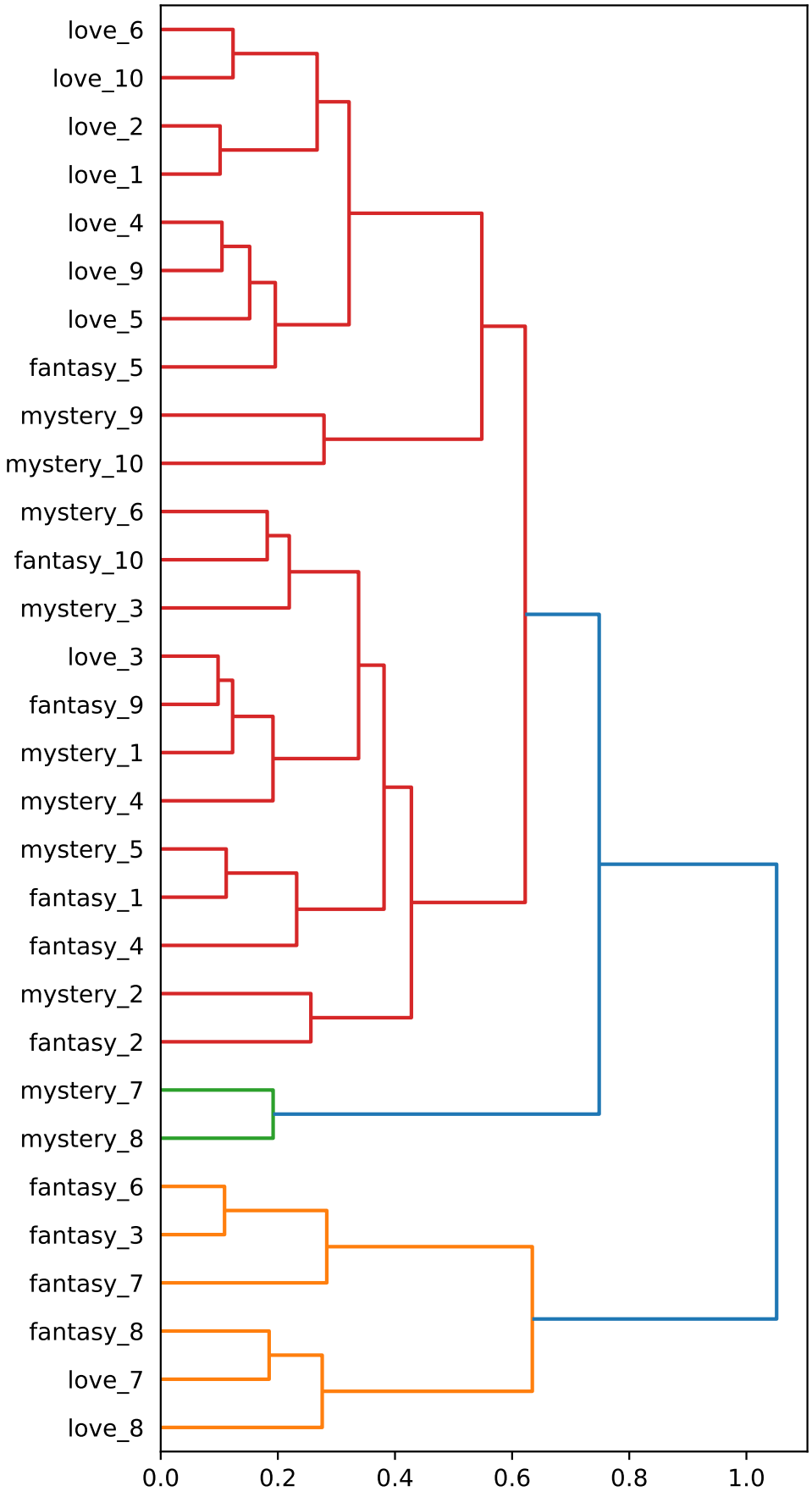

Figure 4.7 Resultofdimensionreductionbyclusteranalysis(3_30_200d_70%).

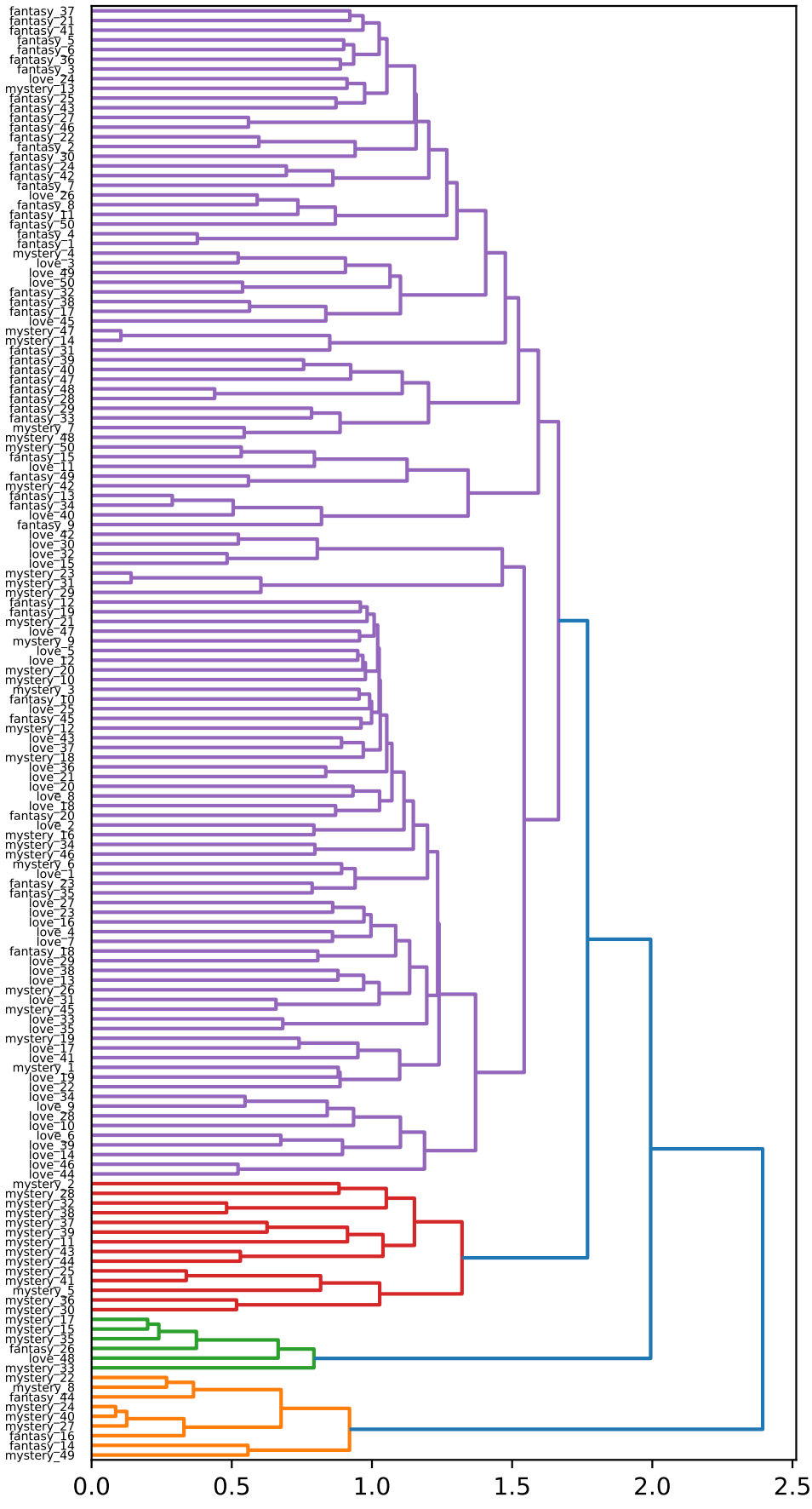

Figure 4.8 ResultofdimensionreductionbyLSA(3_150_90%).

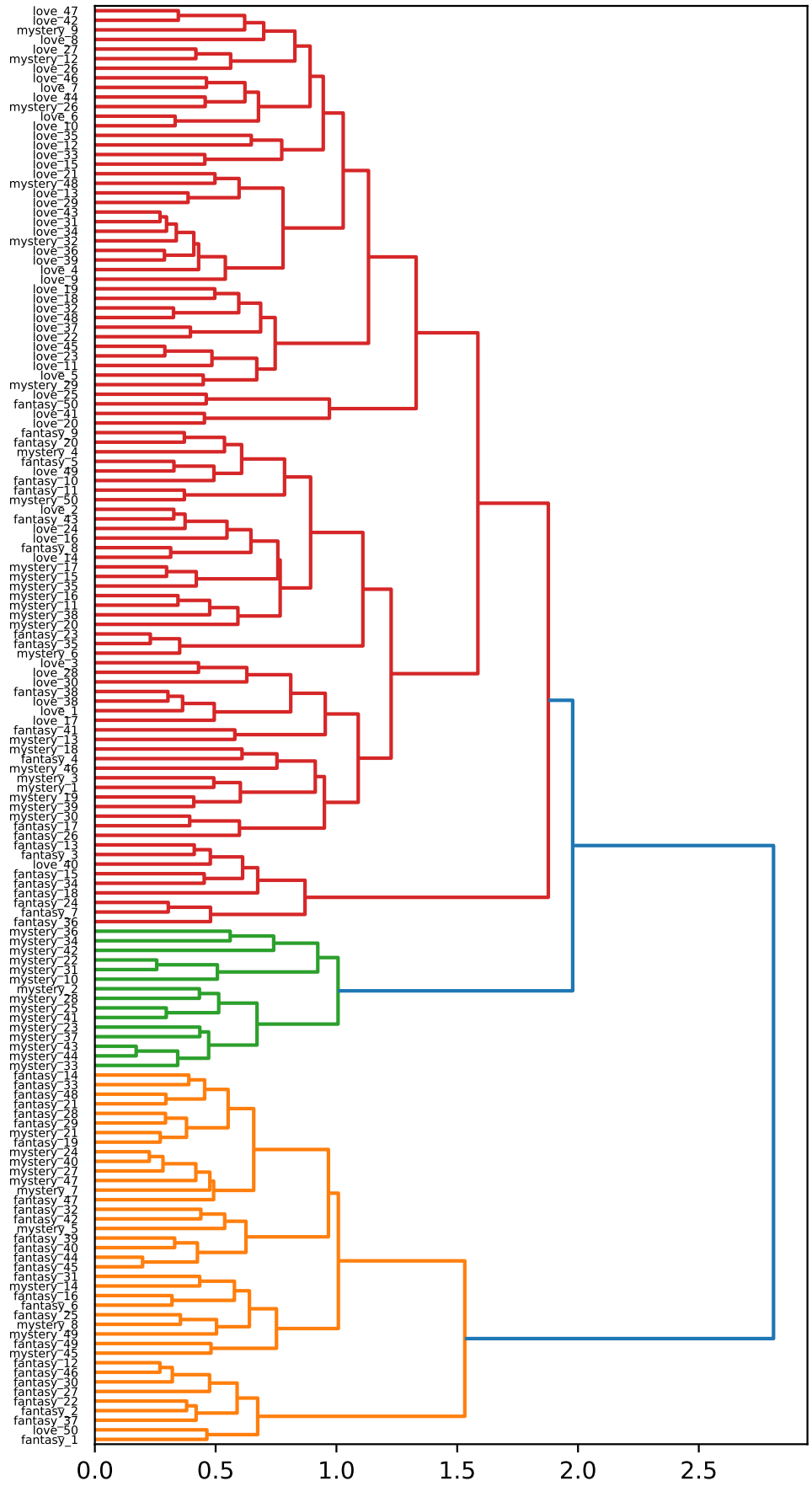

Figure 4.9 Resultofdimensionreductionbyclusteranalysis(3_150_200d_70%).

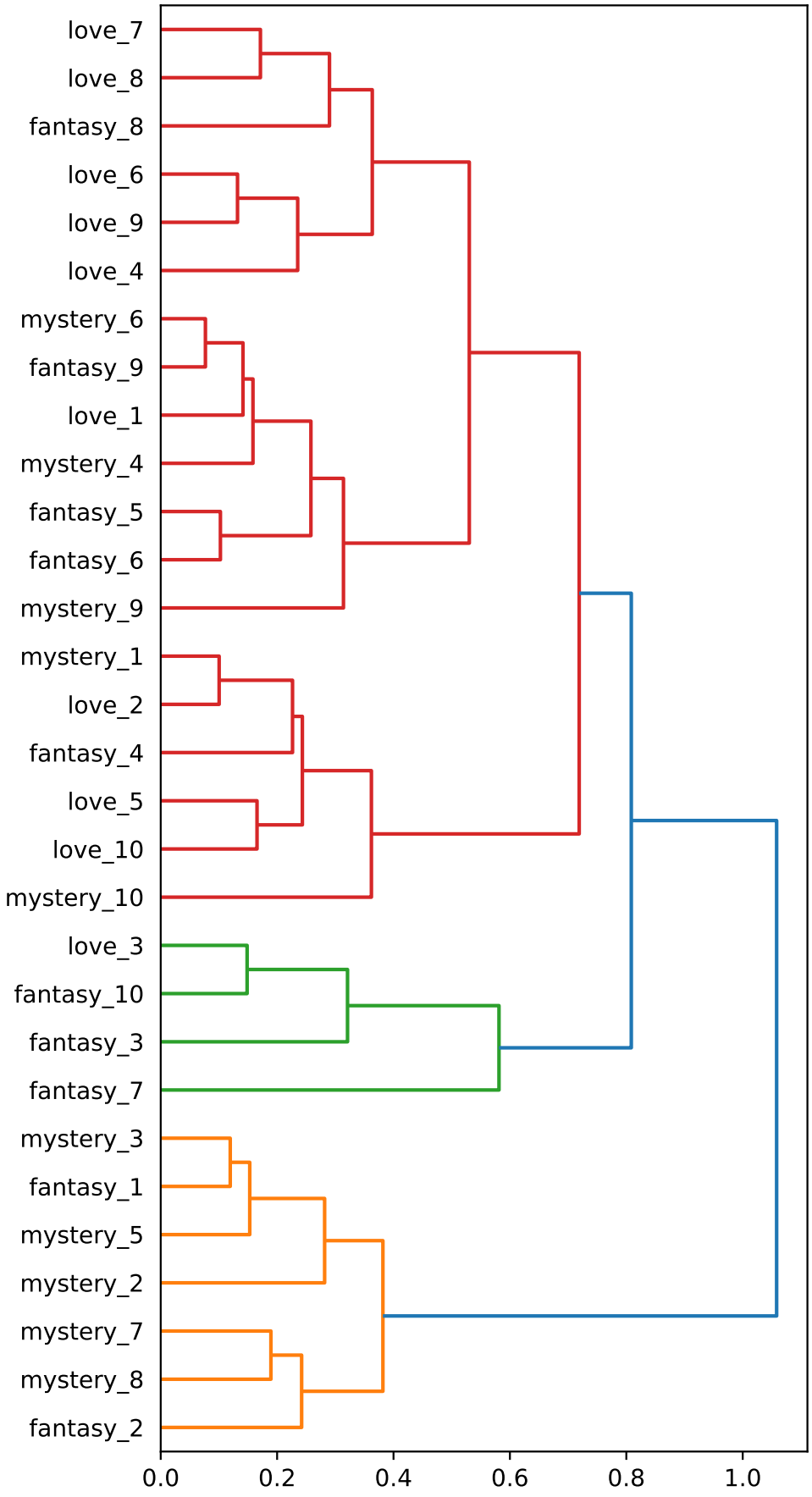

Figure 4.10 Resultofdimensionreductionbyclusteranalysis(3_30_300d_70%).

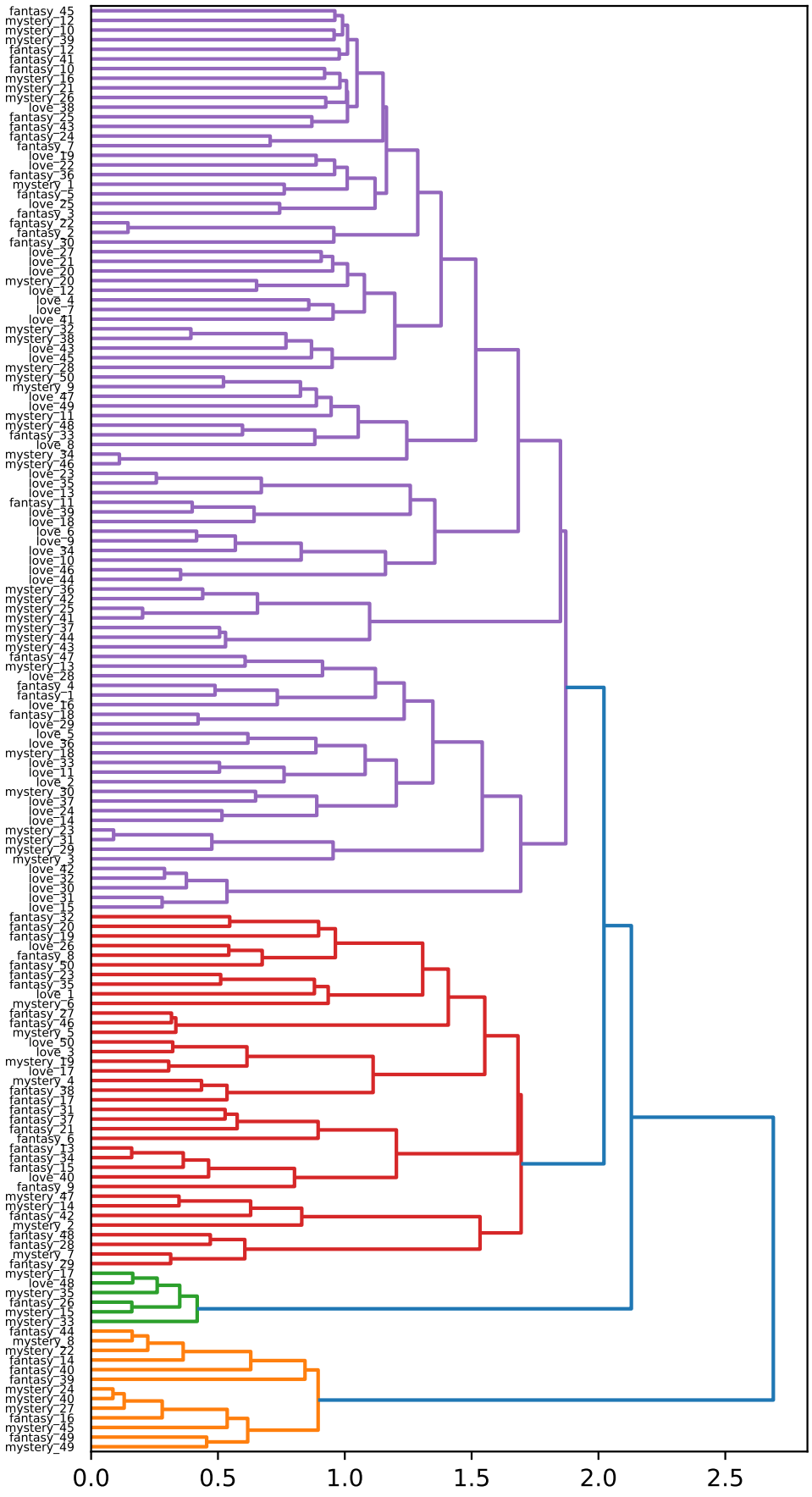

Figure 4.11 ResultofdimensionreductionbyLSA(3_150_70%).

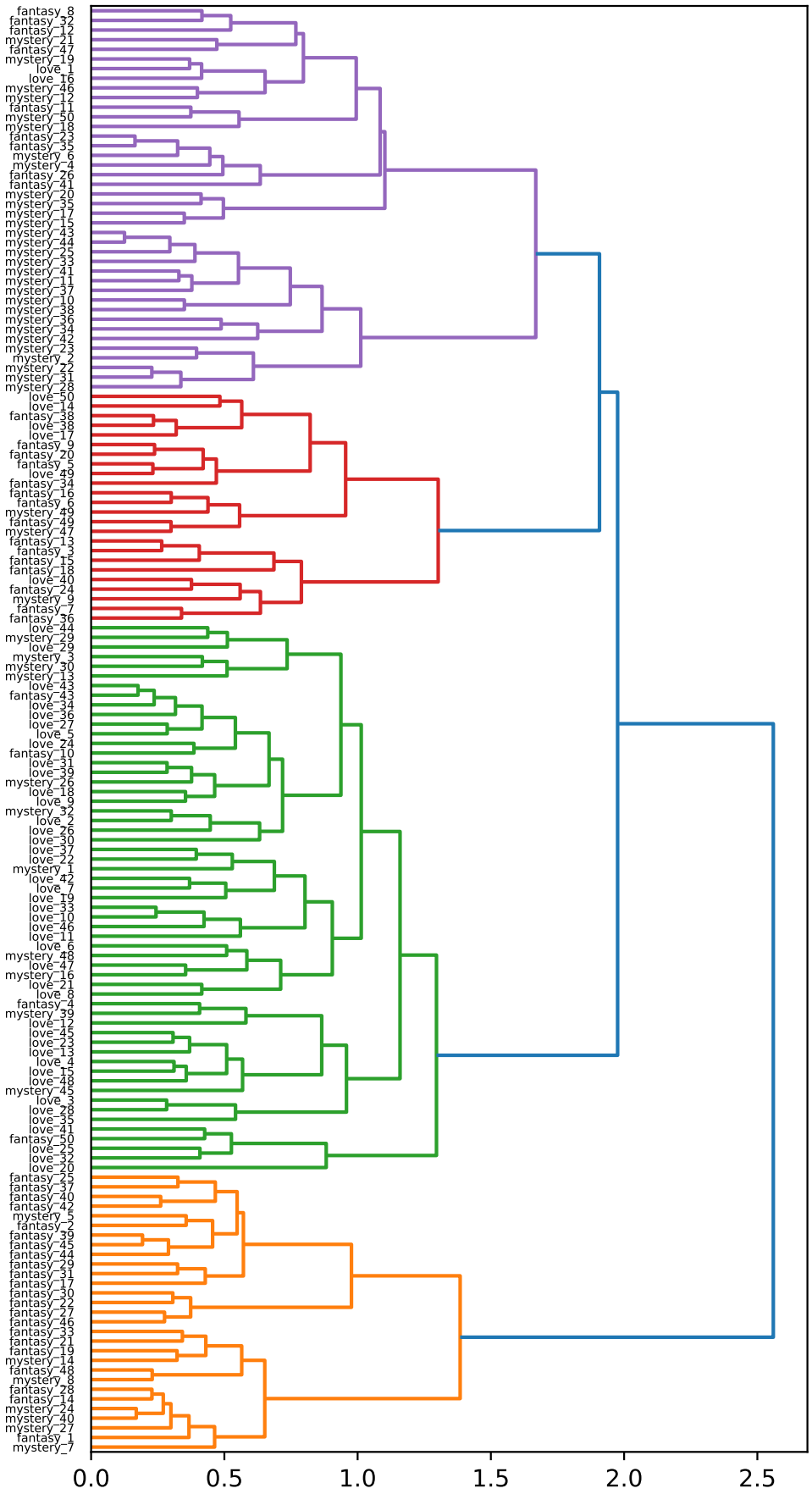

Figure 4.12 Resultofdimensionreductionbyclusteranalysis(3_150_300d_70%).

4.8.2　過去の研究との比較

本研究は過去に本研究室で行われた実験をなぞり実施した。過去の研究との違いは、Word2Vec の次元数とクラスタリングの関係について明らかにするため、異なる次元数の Word2Vec を導入した点である。過去の実験でも同じ東北大学の学習済みモデルを採用しているが、バージョンが異なるため、結果に差が生じた。

具体的には、パターン 1、2、3、4 において過去研究では潜在的意味解析の正解率より、クラスター分析による正解率が低くなっている点である。しかし、パターン 5，6 においては過去の研究と同じくクラスター分析による正解率の方が高いという結果となった。これは、作品数が少ないパターン 1、2，3 においては、単語自体の区別がつかないという潜在的意味解析の長所がさほど表れていないためだと考えられる。これに対して、作品数の多いパターン 2 では、単語自体の意味をとらえることができる Word2Vec によるクラスター分析の方が、有利に働いたと考えられる。また、クラスター分析による利点が表れたパターン 4、5、6 において、Word2Vec モデルの次元数と正解率との間の正の相関関係が顕著に表れた。このことから、より次元数の高い Word2Vec モデルの方が、より深く単語の意味を理解し、より明確に単語の違いを把握することができると考えられる。

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異なる次元数の Word2Vec を採用し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、潜在的意味解析との比較により、クラスター分析による次元削減の有効性を確認したうえで Word2Vec モデルの次元数の差がクラスター分析に与える影響を、小説の分類結果の正解率から明らかにするために行った。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 の導出、Word2vec を用いた単語間の類似度計算を行った。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、文書の類似度を計算する。それにより文書分類を行い、分析結果をデンドログラムと正解率で表した。以上の作業を、作品数、Word2Vec モデルを変更した 6 つのパターンで行い、結果を比較した。出力された結果を比較したところ、特に作品数が多い実験パターンにおいて、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、テキストデータ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、単語間の関係を付加価値として与えられるメリットがある。そのため、本研究の分析対象である小説のあらすじは、その付加価値が良く効くテキストデータであったと考えられ、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、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。そのうえで、同じ作品数において、いくつかの異なる Word2Vec モデルを適用した文書分類の比較より、Word2Vec モデルの次元数が高くなるほど、正解率もそれに比例して高くなることが判明した。この結果が得られた理由として、高い次元数の Word2Vec モデルの方が、低い次元数のモデルよりも、より深く単語の意味を理解していることが挙げられた。しかし、本実験を通して 2 つの懸念点が浮かんできた。1 つ目は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。2 つ目は、Word2Vec の次元数についてである。現在日本語に対応している Word2Vec の学習済みモデルは 50 300 次元までである。また、50 次元の Word2Vec は東北大学が提供しているものではない。本研究では Word2Vec の学習プロセスは対象としていないため、開発元が異なる 50 次元のモデルについては採用しなかった。しかし、Word2Vec の次元数とク

ラスタリングの精度との正の相関関係を明言するには、より多くの作品を対象都市、さらに幅広い次元数で行う必要があると感じた。ただし、クラスター分析の利点が見られたパターン 3、4, 5 においても、100 次元と 200 次元ではわずかな差しか見られなかったため、50 次元のように低い次元数ではなく、より高い次元数の Word2Vec を導入する必要があると考えられる。ただし、次元数を増やすとデータ数が増え、計算量も増加するため、次元削減が困難になる点が懸念される。

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) テキストマイニングとは？解析方法や活用事例、無料・有料ツール解説, 水落 絵理香,HubSpot Japan,2023.
https://blog.hubspot.jp/text-mining

2) 日本語形態素解析の裏側を覗く！MeCab はどのように形態素解析しているか, 齋藤貴生, クックパッド株式会社,2017.
https://techlife.cookpad.com/entry/2016/05/11/170000

3) tf-idf（term frequency - inverse document frequency）とは？,ITmedia Inc.,2021.
https://atmarkit.itmedia.co.jp/ait/articles/2112/23/news028.html

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) 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/（2022 年 12 月 1 日アクセス).

6) 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.

7) scikit-learn の歩き方, 岩通システム株式会社,2017.
https://www.iwass.co.jp/column/column-11.html

8) scipy.cluster.hierarchy.dendrogram,The SciPy community.
https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.dendrogram.html

9) 長谷川翔海　　Word2vec を利用したクラスター分析による文書分類に関する研究, 岐阜工業高等専門学校電気情報工学科卒業研究報告, 2021.
https://www.gifu-nct.ac.jp/elec/deguchi/sotsuron/hasegawa/

10) 藤井康太　　Word2vec を利用した単語のクラスター分析による文書分類に関する研究, 岐阜工業高等専門学校電気情報工学科卒業研究報告, 2022.
https://www.gifu-nct.ac.jp/elec/deguchi/sotsuron/fujii/