卒 業 研 究 報 告 題 目

単語間の距離を用いた
クラスタ分析による文書の類似度計算

Calculating Document Similarity
by Cluster Analysis using Distance between Words

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

2018E39 武藤 昇吾

令 和 0 5 年（ 2 0 2 3 年 ）　　2 月 1 7 日 提 出

Abstract

In this study, a dimensionality reduction method using Word2vec and spectral clustering is proposed. This method reduces the dimensionality by multiplying the document matrix by the matrix created by the spectral clustering. In order to verify the effectiveness of this method, experiments of document classification are carried out using the following three dimensionality reduction methods.

1. Latent Semantic Analysis
2. Dimensionality reduction method using hierarchical clustering
3. Dimensionality reduction method using spectral clustering

The accuracy for each document classification is conpared. The second method is the dimensionality reduction method proposed in traditional research. The text data used in this study were synopsis of the novels. 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 methods. As a result of the experiment, the effectiveness of dimensionality reduction by spectral clustering was confirmed. It was because the dimensionality reduction by cluster analysis could distinguish the meaning of the words.

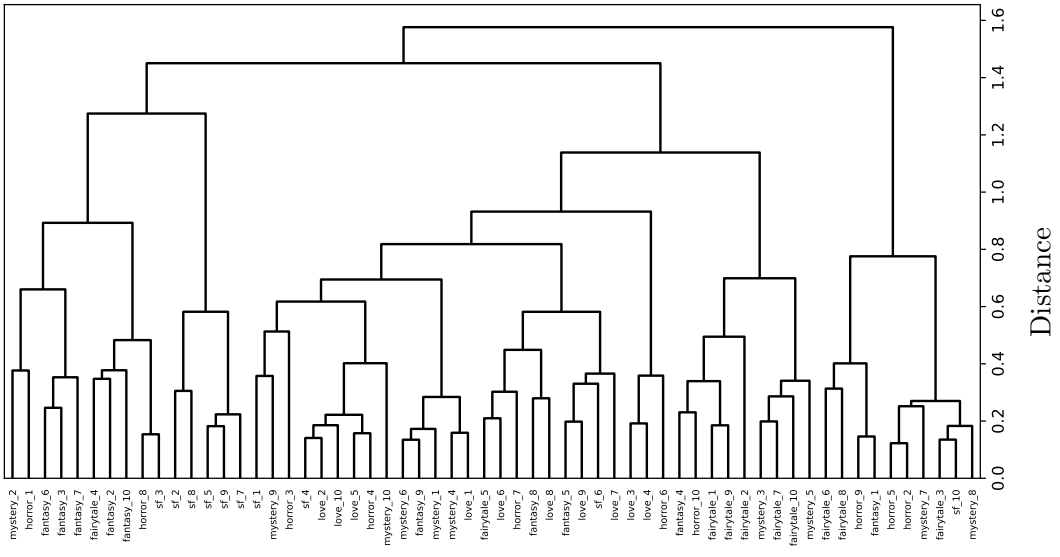

Figure 4.10 Result of dimension reduction by spectral clustering(6_60_80%)

目 次

第1章　序論

現在、コンピュータやスマートフォンといった情報端末の普及により、あらゆる情報が行き交う情報化社会と化している。それに伴い、膨大な量のデータが日々発信、蓄積されている。しかし、その膨大な量のデータの中には真偽のはっきりしないものもあり、すべてのデータが有益なものとは言えない。その膨大な量のデータから自分にとって有益な情報だけを抽出するための技術として生み出されたのがデータマイニングである。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、膨大なデータから有用なパターンやルールを発見する技術で、いまだ発展途上の分野である。

本研究では、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、所属研究室で提案された手法を実装し、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。提案する次元削減手法は、単語の分散表現法である Word2vec とスペクトラルクラスタリングを用いた次元削減法で、これにより単語の意味を考慮した次元削減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、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、あらすじ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、次元削減を下の 3 手法で実装し、それぞれで文書間の類似度を導出する。

- 潜在意味解析（LSA）
- 従来の研究手法（Word2vec+階層的クラスタリング）
- 本研究手法（Word2vec+スペクトラルクラスタリング）

潜在的意味解析は従来からの次元削減法で、従来の研究手法と合わせて比較することで本研究手法の有用性を検証する。手法の有用性の評価を出力された樹形図と分類タスクをこなす機械学習モデルに用いられる正解率という評価手法によって判断する。これらの検証結果を用いて、最終的に本研究の提案手法の有用性がどのようなものかを検討する。

第2章　テキストマイニング

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をいう。膨大な量のデータから有益な情報を得ることは人間には不可能である。そのためコンピュータを活用することによって、人間では発見できない法則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　テキストトマイニング

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。自然言語処理を用いて文章を単語列に分解し、それらの出現頻度や相関関係を分析することで有用なパターンやルールを発見することができる。

英語などの単語の境界が明確な言語に対して、日本語のような境界が曖昧で文法的な揺らぎが大きい言語は解析が困難だったが、自然言語処理の発展により実践的な水準の分析が可能となった。主にSNSのデータや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・分類などに活用されている。

2.2　自然言語処理

2.2.1　自然言語処理

自然言語とは、人間が日常的に使用し、コミュニケーションを行うのに用いる言語のことである。構文や単語の用法に厳格な規則が存在しないと考えられるものを指す。例として、「英語」「中国語」「日本語」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出された言語である人工言語が存在する。プログラミング言語や論理式など、構文や意味が厳格に定義された言語である。例として、「Python」「C言語」などが挙げられる。自然言語は規則が曖昧なため、単語や文が複数の意味で解釈可能であり、感情でも文章を制御しやすいため、様々な人とコミュニケーションが取れる。

2.2.2　自然言語の曖昧性

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、解釈が複数になることをいう。例として「厚い」という単語は、「厚い本」という文では、他方の面まで距離が大きいという意味を表しているが、「情に厚い」という文では、真心があるという意味を表している。このように「厚い」という単語には２つの解釈が存在することになる。

類犠牲とは、異なる単語が同じような意味を持っているということをいう。例として、「直す」「正す」というような、読み書きが異なる場合においても同じ意味を持つ単語が挙げられる。こうした曖昧性がテキストマイニングの大きな壁となっている。

2.2.3　自然言語処理

自然言語処理とは、自然言語をコンピュータで処理・分析する技術のことをいう。自然言語は日常的に人が使用するものであり、曖昧性を含んでいる。そのため、コンピュータで機械的に分析することで、大量のテキストデータの解析、非構造化データの処理などができるようにする。

2.2.4　形態素解析

形態素とは、単語において意味を持つ最小の単位であることをいう。

形態素解析とは、自然言語で書かれた分を言語上の最小単位である形態素に分解し、それぞれの品詞や変化などを割り出すことをいう。形態素解析をすることで、テキストデータをコンピュータで分析可能な形にしている。

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。本研究では形態素解析器として、MeCab というソフトウェアを使用した。

2.2.5　Mecab

MeCab は京都大学と日本電信電話株式会社コミュニケーション科学基礎研究所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。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 によって形態素解析をするときは、オプションを指定することで様々な結果を出力できる。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。

「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

第3章　実験で用いた技術・手法

3.1　単語のベクトル化と類似度計算

3.1.1　単語のベクトル化

テキストマイニングを行う上で、文字列である単語をベクトルに変換する処理が必要である。自然言語が数値ではないデータのため、自然言語処理はコンピュータにとって難しい処理の一つとされている。そのため、何らかの方法で文書の特徴量を抽出し、文書を特徴ベクトルに変換する必要がある。

特徴ベクトルはn個の特徴量 $x_i(i = 1, 2, 3, \cdots, n)$ を縦に並べたn次元ベクトルで定義される。文書 d_j の特徴ベクトル x_j の定義を以下に示す。

$$x_j = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} \tag{3.1}$$

x_i には文書の特徴量が入り、特徴量数 n はベクトル表現手法ごとで異なるが、主に単語数と等しくなる。また、プログラムによる解析の際は、各特徴ベクトルを更に横に並べた $n \times m$ 行列を作成することが多い。

3.1.2　Tf-Idf

Tf-Idfとは、文書内の単語の重要度を示す手法の一つである。全文章中の全単語と文書間の重要度を計算し、文書行列と呼ばれる単語数×文書数の行列を作成する。

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、文書における特定の単語の出現頻度を表す。出現頻度が高い単語ほど重要であると考えられ、出現頻度が低い単語ほどさほど重要ではないと考えられる。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、さほど重要ではない単

語と考えられ、Idf が低い値となる。逆にと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 はこれら二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。

$$Tf_{ij} = \text{文書 } d_i \text{ における単語 } t_{ij} \text{ の出現回数} \tag{3.2}$$

$$Idf_{ij} = \log \frac{\text{全文書数} + 1}{\text{単語 } t_{ij} \text{ が出現する文書数} + 1} + 1 \tag{3.3}$$

$$TfIdf_{ij} = Tf_{ij} \times Idf_{ij} \tag{3.4}$$

3.1.3 cos 類似度

cos 類似度とは、2 つのベクトルがどのくらい似ているかという類似性を表す尺度の一つである。cos 類似度は 2 つのベクトルがなす角のコサイン値のことで、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。

cos 類似度は以下の式で導出される。

$$\cos(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|} \tag{3.5}$$

3.2 次元削減

3.2.1 次元削減

次元削減とは、特徴ベクトルの次元数を減らすことである。文書を特徴ベクトルに変換する際、その特徴量数は膨大になることがほとんどである。前項の Tf-Idf も単語数だけ特徴量が存在することになるため、文書数、文字数が多いほど特徴量数が増加する。しかし、特徴量数の増大はテキストマイニングをするうえでの壁となり、解析の計算増大にもつながってしまう。そのため、次元削減によってなるべくデータの意味を保ったまま特徴量を減らすことで、計算量の削減や新たな特徴量の抽出が可能となる。

3.2.2 主成分分析 [1)]

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 (principal component analysis; 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、$N(N \leq P)$ 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。

$$z_n = \sum_{p=1}^{P} a_{pn} x_p \tag{3.6}$$

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。

$$\sum_{p=1}^{P} a_{pn}^2 = 1 \qquad (\forall n) \tag{3.7}$$

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 (3.6) 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 主成分で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

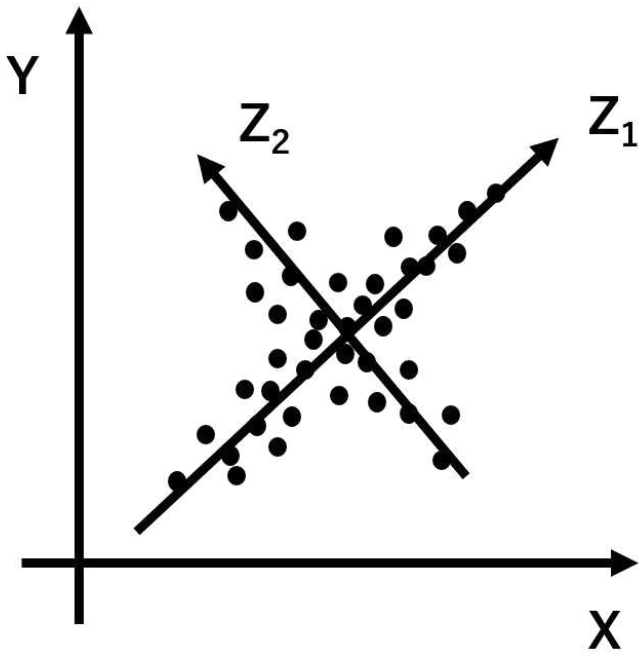

Figure 3.1　Example of two-dimensional data.

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。

3.2.3　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、ある主成分の表す情報は、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。

$$P_n = \frac{\lambda_n}{\sum_{p=1}^{P} \lambda_p} \tag{3.8}$$

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。

$$C_n = \sum_{i=1}^{N} P_i \tag{3.9}$$

3.2.4 潜在的意味解析

テキストマイニングにおいて、形態素解析後に単語-文書行列が生成される。単語-文書行列は式 (3.10) で表される。

$$TD = \left(\begin{array}{c|cccc} Term & doc_1 & doc_2 & \cdots & doc_N \\ \hline w_1 & I_{w_1,doc_1} & I_{w_1,doc_2} & \cdots & I_{w_1,doc_N} \\ w_2 & I_{w_2,doc_1} & I_{w_2,doc_2} & \cdots & I_{w_2,doc_N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_M & I_{w_M,doc_1} & I_{w_M,doc_2} & \cdots & I_{w_M,doc_N} \end{array}\right) \tag{3.10}$$

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 (singular value decomposition; 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。

$$TD = U\Sigma V^T \tag{3.11}$$

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 (3.12) 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。

$$TD_k = U_k^T\, TD \tag{3.12}$$

3.2.5 クラスター分析

クラスター分析とは、異なるものが混ざり合う集団から互いに似ているものを集めてクラスターと呼ばれる集団を作り、対象を分類することである。形や色などで分類する場合とは違い、クラスター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、教師なしの分類法である。クラスター分析の手法には、階層的クラスタリングと非階層的クラスタリングの 2 つが存在する。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ーに分類していく手法である。2 つの対象の距離を計算し、その距離が短い者同士から順にクラスターを 作成していく。クラスター同士の距離の計算方法はいくつか存在し、対象のデータに最も適した方法を選択する。代表的な計算方法は以下の 4 つである。

- 最短距離法（最も近いデータ同士の距離をクラスター間の距離とする）
- 最長距離法（最も遠いデータ同士の距離をクラスター間の距離とする）
- 重心法　　（クラスター内のデータの重心をクラスターの座標とする）
- ウォード法（仮に結合した際の分散が最も小さいクラスター同士が結合する）

また、階層的クラスタリングの分類過程は階層的な構造になるため、Figure 3.2 のようなデンドログラム（樹形図）で表すことができる。このデンドログラムの横に閾値で直線を引くことで、指定数のクラスターに分類することができる。

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ーに集まるよう分

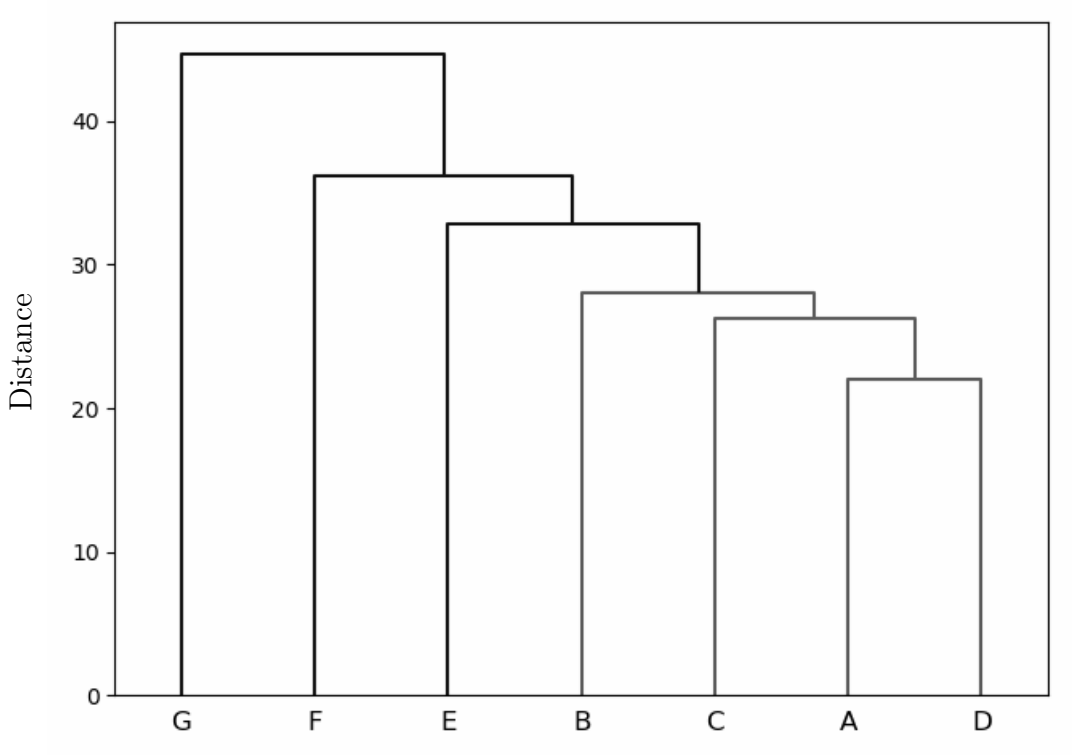

Figure 3.2　Dendrogram.

割する手法である。しかし階層的クラスタリングとは異なり階層的な構造を持たず、あらかじめいくつのクラスターに分類するかを決めておく必要がある。そのため事前に計算を行うことができず、また最適なクラスター数を自動的に計算する手法も存在しないため、分析者によって大きく結果が変わることがある。非階層的クラスタリングの手法として、スペクトラルクラスタリングがあるが、これについては次項に詳しく記載する。

本研究では次元削減に LSA、従来の研究手法の階層的クラスタリング、非階層的クラスタリングであるスペクトラルクラスタリングの３つを使用している。また、次元削減後クラスター分析を行うことで文章の分類を行なっている。階層的クラスタリングでのクラスター間距離の測定方法は最も使用されているウォード法を採用した。

3.2.6　スペクトラルクラスタリング [2)]

スペクトラルクラスタリングとは、固有値問題による連結成分分解を応用した非階層的クラスタリングのアルゴリズムである。具体的には、以下のようなアルゴリズムである。クラスター数 k が与えられているとする。

1. データからグラフを構築する。
2. グラフから隣接行列 W、degree matrix D を計算し、unnormalized graph Laplacian

$L(= D - W)$ を計算する。

3. unnormalized graph Laplacian L の固有値が小さい順に固有ベクトル K 個 $(u_1, \ldots, u_k)$ を計算する。
4. 固有ベクトルを並べて行列 $U = (u_1, u_2, \ldots, u_k)$ を作る。
5. 行列 U の行ベクトルを K-means などを用いて K 個にクラスタリングする。
6. K-means の結果、i 行目が入ったクラスターを頂点 v_i の入るクラスターとする。

スペクトラルクラスタリングの特徴としては、データからグラフを生成し、グラフの連結成分分解を応用してクラスタリングする。クラスタリングアルゴリズムとして古典的なものに、K-Means や Gaussian mixture model がある。K-Means や Gaussian mixture model はクラスターの中心点からの距離に基づいてクラスタリングを行うが、スペクトラルクラスタリングでは連結性に注目してクラスタリングを行うため、K-Means や Gaussian mixture ではうまくクラスタリングできなかったようなデータをうまくクラスタリングできることがある。

3.2.7 正解率 [3)]

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 (3.13) のように計算される。

$$\text{正解率} = \frac{\text{正解した数}}{\text{予測した全データ数}} \tag{3.13}$$

実際に例の混同行列を計算すると、対角成分にある正しく予測された値の合計値 24 を

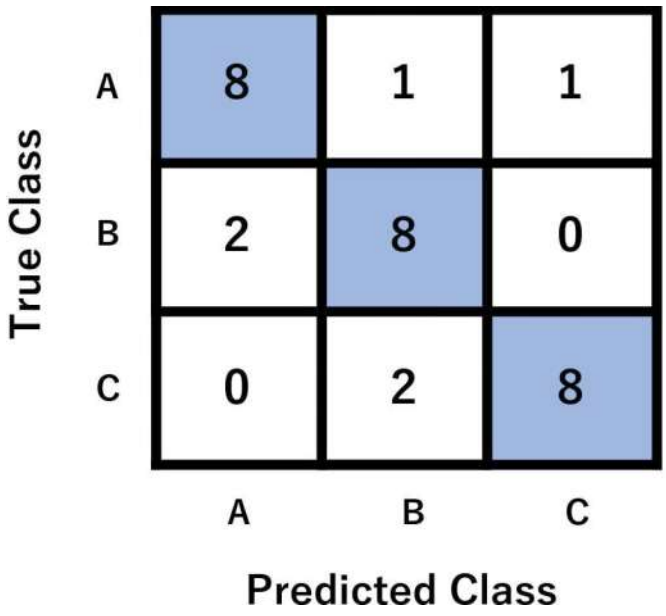

Figure 3.3　Confusion Matrix.

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 (3.13) 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。

3.3　WebAPI

3.3.1　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。

3.3.2 WebAPI[4)]

本研究で言うWebAPIとは、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のことである。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。有名な例として、Googleが提供する各種APIやAmazonのProduct Advertising API、Twitter社が提供するTwitter APIなどが挙げられる。近年ではWebAPIを公開することの重要度が高くなっており、企業によってはサービスの価値や収益を左右するケースまでもが確認されている。そのため活用事例も多く、特にSNSやECサイト等での利用が多くみられる。

3.3.3 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、作者、小説の評価などが挙げられる。本研究では、複数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。

3.4 Python

3.4.1 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、多くのユーザーから支持を得ている。

3.4.2 MeCab[7)]

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を採用した。

3.4.3 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。

3.4.4 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 (3.2)、(3.3) 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

3.4.5 cos 類似度

Python では scikit-learn ライブラリに用意されている cosine_similarity 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。

$$\cos\left(\vec{a},\vec{b}\right)_{distance} = 1 - \frac{\vec{a}\cdot\vec{b}}{|\vec{a}||\vec{b}|} \tag{3.14}$$

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。

3.4.6 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。

3.4.7 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。

3.4.8 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。

3.4.9　スペクトラルクラスタリング

Python では scikit-learn ライブラリに用意されている SpectralCluestering 関数でスペクトラルクラスタリングを行うことができる。

3.4.10　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。

3.5　Word2vec

3.5.1　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。

$$「\text{king}」 - 「\text{man}」 + 「\text{woman}」 = 「\text{queen}」 \tag{3.15}$$

式 (3.15) 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。

3.5.2 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。

3.5.3 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 (3.10) 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 (3.16) で表されるクラスターと名詞の行列と式 (3.10) にある元の文書行列との積で求められる。

$$CW = \left(\begin{array}{c|cccc} Term & w_1 & w_2 & \cdots & w_M \\ \hline C_1 & a_{1,1} & a_{1,2} & \cdots & a_{1,M} \\ C_2 & a_{2,1} & a_{2,2} & \cdots & a_{2,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_N & a_{N,1} & a_{N,2} & \cdots & a_{N,M} \end{array}\right) \tag{3.16}$$

各クラスターを $C_i(i = 1, 2, \ldots, N)$ 単語を $w_k(k = 1, 2, \ldots, M)$ と単語 w_j の cos 類似度を $S_{k,j}$ とする。このとき、式 (3.16) にある要素 $a_{i,j}$ は次の式で与えられる。

$$a_{i,j} = \begin{cases} \dfrac{1}{|C_i|} \displaystyle\sum_{k \in C_i} S_{k,j} & (j \in C_i) \\ 0 & (j \notin C_i) \end{cases} \tag{3.17}$$

第4章　実験

4.1　実験の概要

本実験では、以下の3種類の次元削減法で文書分類を行う。

- 潜在意味解析
- Word2vec＋階層的クラスタリング（ウォード法）
- Word2vec＋非階層的クラスタリング（スペクトラルクラスタリング）

潜在意味解析は、従来の方法との比較を行うために使用した。また、2つ目の次元削減手法は従来の研究で用いられた手法で、ウォード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。これらと提案手法であるスペクトラルクラスタリングを比較することで、提案手法の優位性を検証する。実験では、潜在意味解析の際の累積寄与率を基に次元の削減数を決定し、それぞれ同じ次元数まで次元削減を行う。その後クラスター分析による分類を行い、正解率の計算、比較を行う。また、その際に累積寄与率を変化させて実験することで、累積寄与率と精度の相関を検証する。

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。[8)9)] 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。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（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10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、Homebrew を使いバージョン 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 [10)]

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。そのため、本研究では東北大学の乾・岡崎研究室で作られた日本語モデルを使用した。このモデルは学習に Wikipedia の全記事が使われており、固有表現を考慮するように学習されている。モデルは複数あるが、その中でも 2019 年に学習された 300 次元のモデルを Gensim で読み込んで使用した。

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、3.3.3 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 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、ジャンルごとに 50 作品、計 300 作品取得した。

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 ジャンルとなっている。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 つとなる。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。実験パターン 3 と実験パターン 4 のジャンル数が 6 であるのは、上記の 6 つのジャンル全てを含んで解析を行うということになる。

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 と実験パターン 4 では選択した各ジャンルにおいて、取得した 50 作品すべてを含めたものということになる。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) … 1219 語
- 実験パターン 2 (単語種類数) … 3554 語
- 実験パターン 3 (単語種類数) … 1779 語
- 実験パターン 4 (単語種類数) … 5554 語

4.3.2 TfIdf の計算

Python を用いて TfIdf を計算する。計算は、3.4.4 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、TfIdf 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。3.4.6 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 3.2.3 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

3.4.7 項に記述があるように、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をもとに指定した。

4.4.3 スペクトラルクラスタリングによる次元削減

スペクトラルクラスタリング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 (3.5) 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、クラスターを分割する
4. クラスタ内の総単語数を計算する

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

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（3.17）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。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。

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　類似度による文書分類

4.5.1　文書間の cos 類似度

次元削減された行列に対して、3.4.5 項で記述したように cosine_similarity 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 cosine_similarity 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。3.4.8 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 3.2.5 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

3.4.8 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。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。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、3.4.10 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。
2. confusion_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
3. classification_report 関数で正解率を求める
4. 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
5. 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、階層的クラスタリングの結果による正解率、スペクトラルクラスタリングの結果による正解率、3 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　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。グラフ全体から、実験のパターンによって本研究の正解率が従来の研究手法の正解率よりも高くなることがわかった。

また、それぞれのグラフを比較して、以下の分析結果が得られた。

- パターン 1、パターン 2 のジャンル数が 3 つのグループはどの次元削減法でも正解率が高く、全体的に 0.6 付近の値をとっていることがわかる。また、ジャンル数が 3 つの場合は従来の研究手法の方が正解率が高いことがわかる。
- パターン 3、パターン 4 のジャンル数が 6 つのグループはジャンル数が 3 つのグループと比較して正解率が低くなっていることがわかる。次元削減法での比較はジャンル数が 3 つの場合と同様だが、正解率の値が全体的に 0.4 以下になっている。また、ジャンル数が 6 つの場合は、本研究手法の方が正解率が高いことがわかる。
- パターン 1、パターン 3 の文書数 10 のグループと、パターン 2、パターン 4 の文書数 50 のグループを比較すると、文書数が多くなっても正解率がほとんど変化しないことがわかる。

以上のことから、正解率と関係性があると考えられるのはジャンル数で、累積寄与率、文書数は関係性が低いと考えられる。

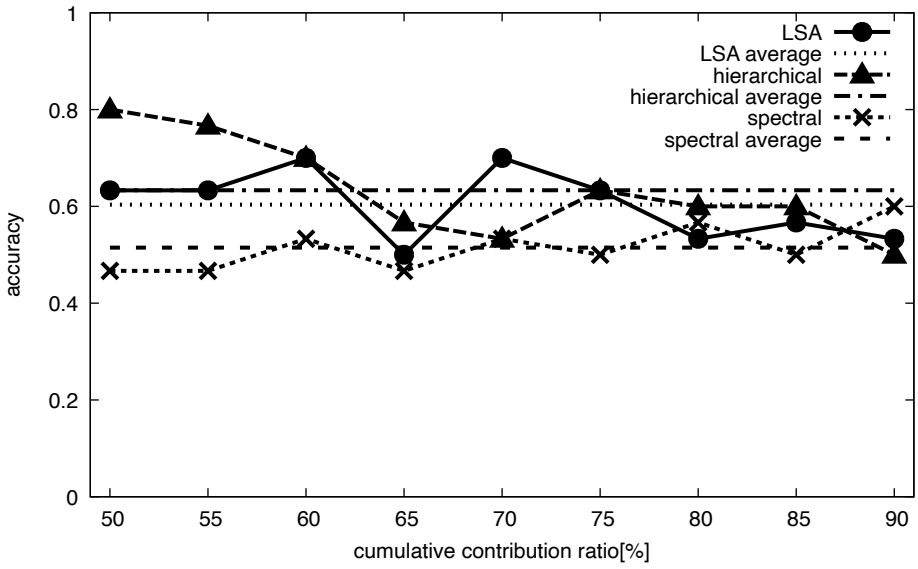

Figure 4.1　Accuracy of 3 genres and 30 works.

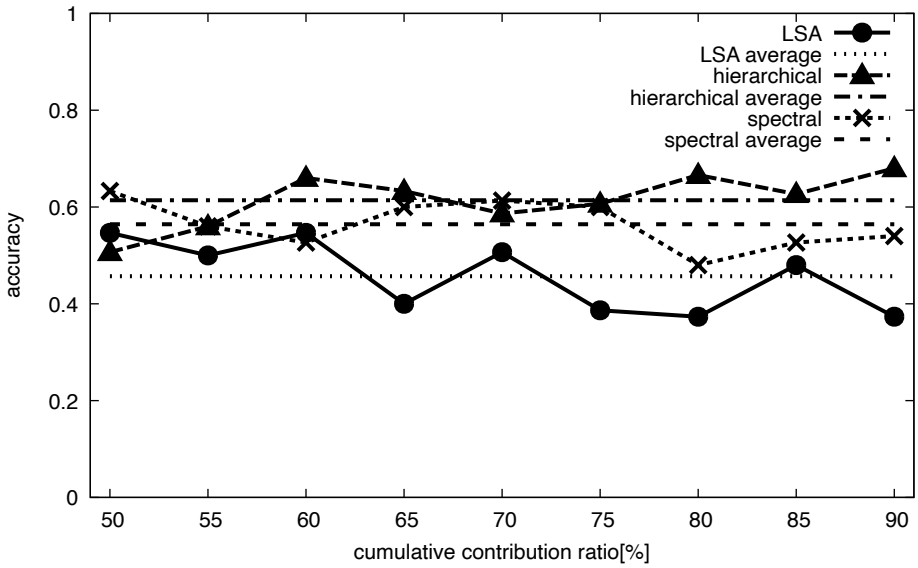

Figure 4.2　Accuracy of 3 genres and 150 works.

4.7.2　文書分類のデンドログラム

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。正解率の違いでどのようにデンドログラムの出力に違いが出るかを確認するために、階層的クラスタリングとスペクトラルクラスタリングの正解率の差がなく、潜在意味解析の正解率には差がある累積寄与率 80%においてのデンドログラムを出力する。その結果、Figure 4.5、Figure 4.6、Figure 4.7、Figure 4.8、Figure 4.9、Figure 4.10 のようなデンドログラムとなった。

Figure 4.5〜Figure 4.10 の図題の最後の括弧は (ジャンル数_文書数_累積寄与率) を表

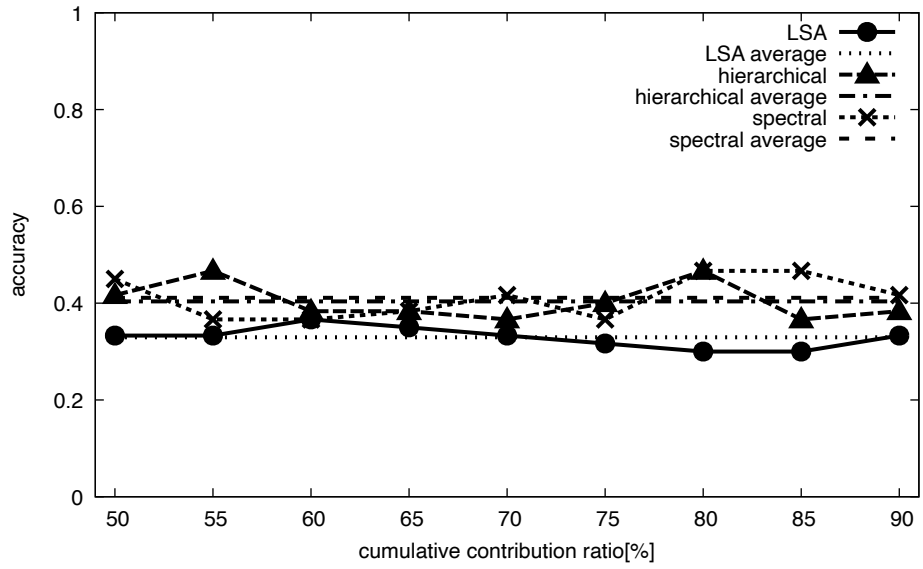

Figure 4.3　Accuracy of 6 genres and 60 works.

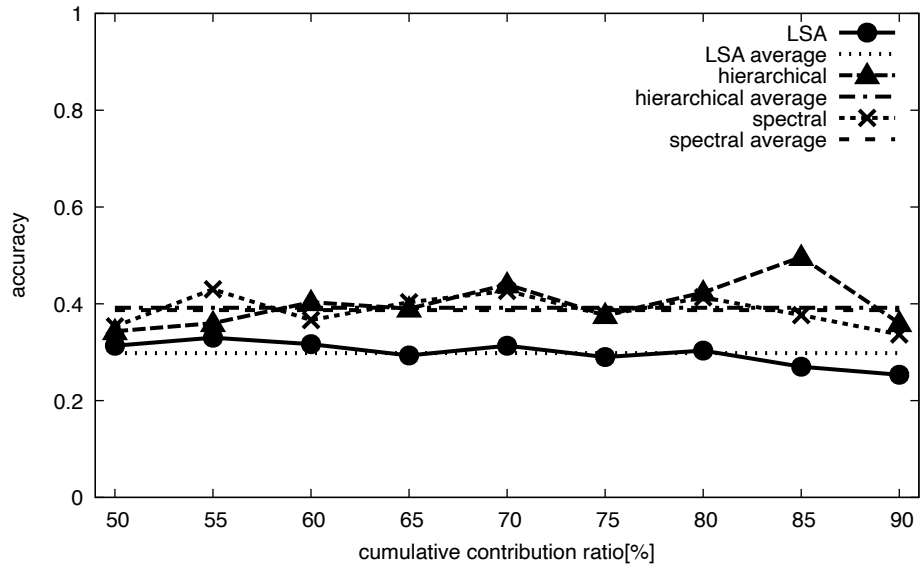

Figure 4.4　Accuracy of 6 genres and 300 works.

している。

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 4.5 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、上部ではファンタジーのクラスターができており、中部ではミステリーのクラスターができている。恋愛は比較的下部にまとまりながら分布している。次に Figure 4.6 の実験パターン 1 における階層的クラスタリングによる結果を確認するとそれぞれのジャンルが 4 つくらいの数でクラスターを形成している。しかし、そ

れ以上の大きさのクラスターでは、ジャンルごとにまとまっていないことがわかる。次に、Figure 4.7 の実験パターン 1 におけるスペクトラルクラスタリングによる結果を確認すると、下部のほうでファンタジーのクラスターができている。また、ミステリーも比較的同じクラスター内に分布している。しかし、恋愛だけは様々なクラスターに分布していることがわかる。以上のことから、実験パターン１ではデンドログラムを見ただけではほとんど違いがわからないことがわかった。

次に実験パターン 3 について比較を行う。Figure 4.8 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、各ジャンル３つくらいの数でクラスターが形成されているが、それ以上の大きなクラスターでは、違うジャンルでクラスターが形成されていないことがわかる。次に、Figure 4.9 の実験パターン 3 における階層的クラスタリングによる結果を確認すると、SF と童話がそれぞれまとまって同じクラスターに属していることがわかる。次に、Figure 4.10 の実験パターン 3 におけるスペクトラルクラスタリングによる結果を確認すると、上から、ファンタジー、SF、ミステリー、恋愛、童話、ホラーと多少のばらつきはあるもののほとんどこのように分布していることがわかる。以上のことから、実験パターン３では、３つの手法の中ではスペクトラルクラスタリングによる次元削減の分析結果がこちらの意図した通りになったものと考えられる。

4.8　考察

4.8.1　各手法の比較

4.7.1 項及び Figure 4.1〜Figure 4.4 より、本研究手法は文書のジャンル数が増えると正解率がほかの手法よりも高くなることがわかる。しかし、ジャンル数が少ないと潜在意味解析、階層的クラスタリングよりも正解率が低くなってしまうことがわかる。これは、スペクトラルクラスタリングのクラスタリング方法が関係していると考えられる。

スペクトラルクラスタリングはデータからグラフを生成し、グラフの連結成分分解を応用してクラスタリングしている。そのため、スペクトラルクラスタリングでは、データからグラフを生成し、最適なグラフの分割を評価関数としている。そのため、データ数が多くなってもクラスタリングの結果にはあまり影響ないと考えられる。しかし、階層的クラスタリングは、近い単語から順にクラスタリングしている関係上ジャンル数が少ないほど正解率が高く、ジャンル数が多くなると正解率が低くなると考えられる。このことから、スペクトラルクラスタリングは、ある程度のジャンル数があると最適なグ

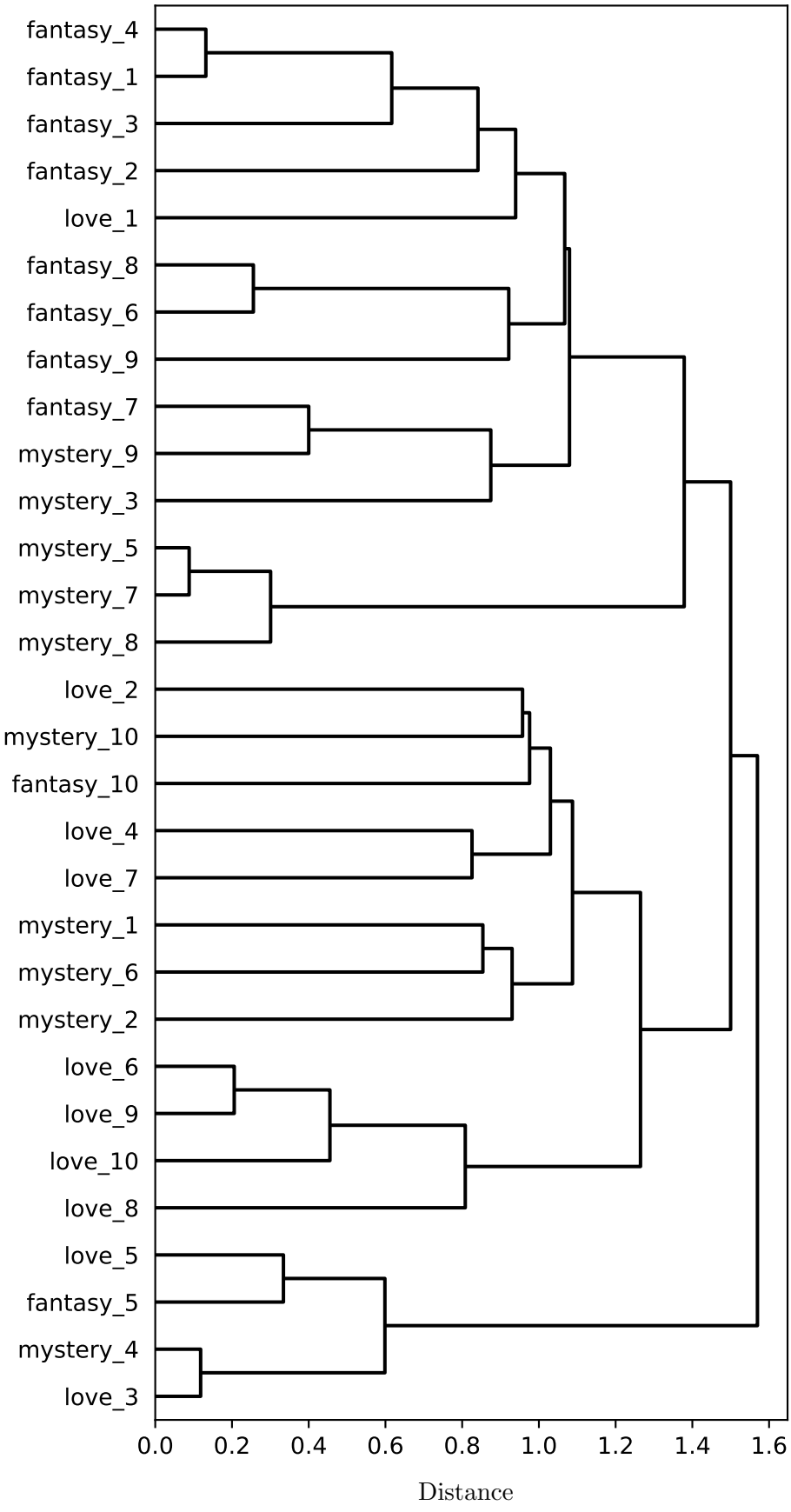

Figure 4.5 Result of dimension reduction by LSA(3_30_80%).

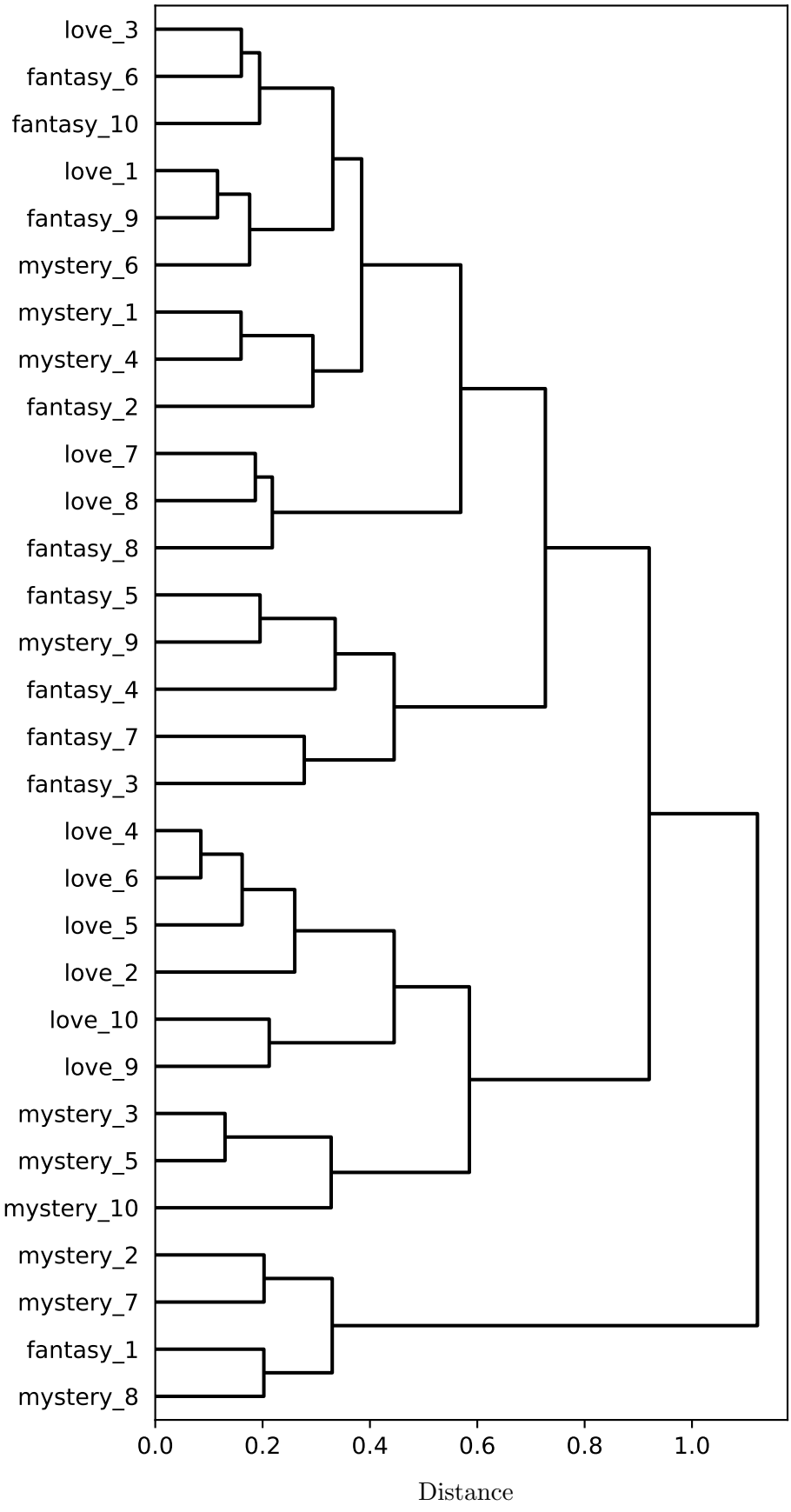

Figure 4.6 Result of dimension reduction by hierarchical clustering(3_30_80%).

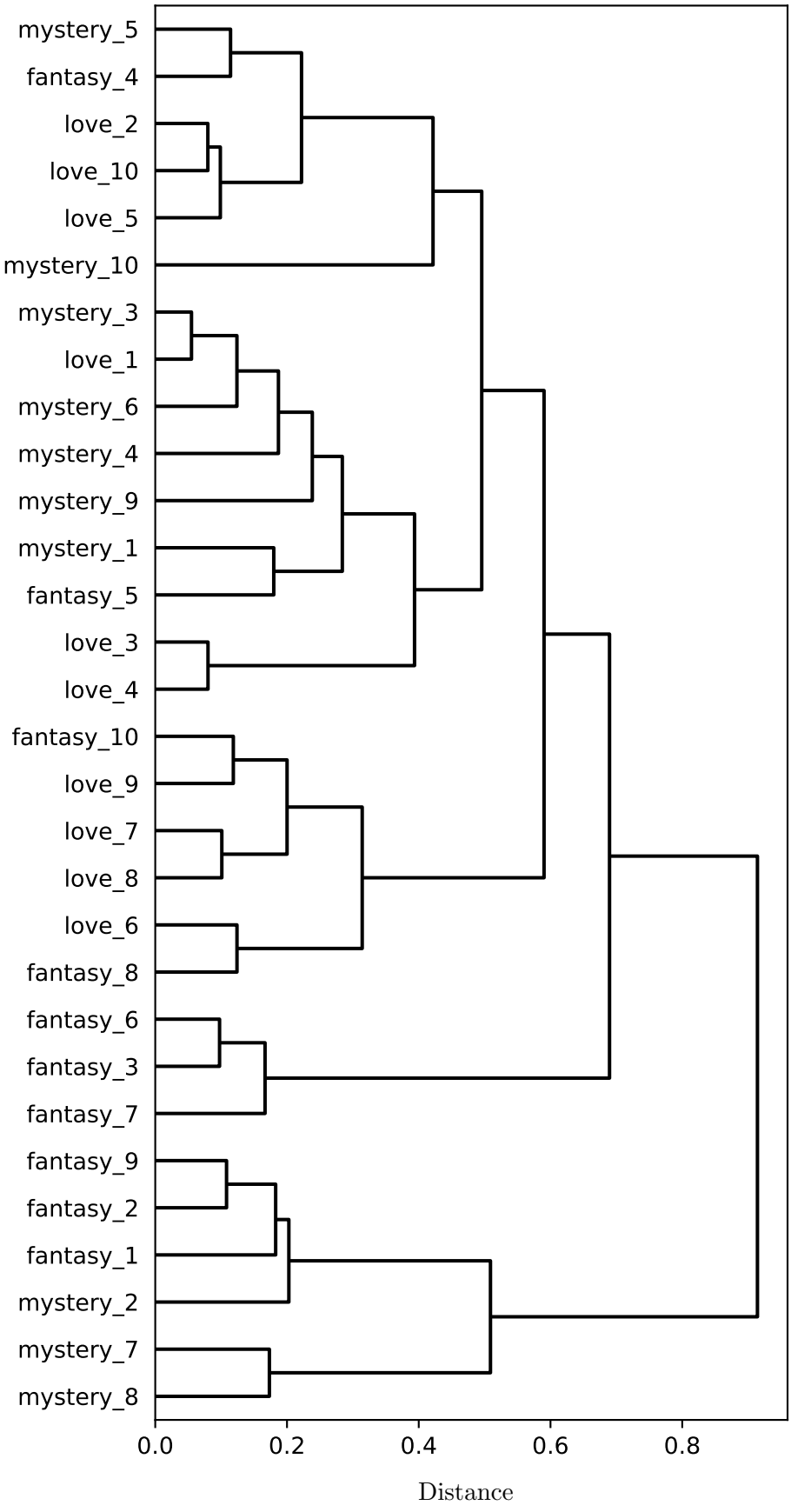

Figure 4.7 Result of dimension reduction by spectral clustering(3_30_80%).

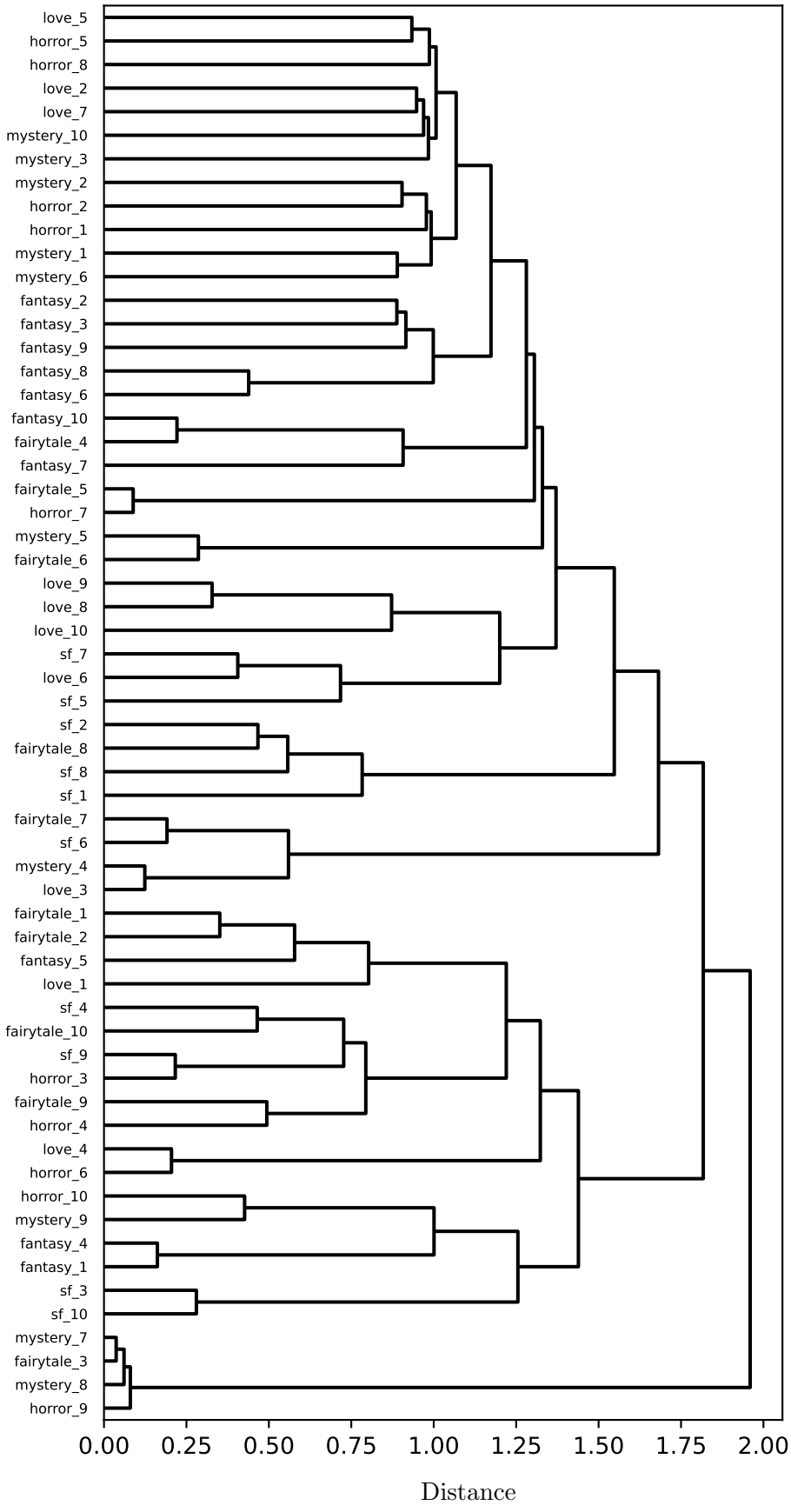

Figure 4.8 Result of dimension reduction by LSA(6_60_80%).

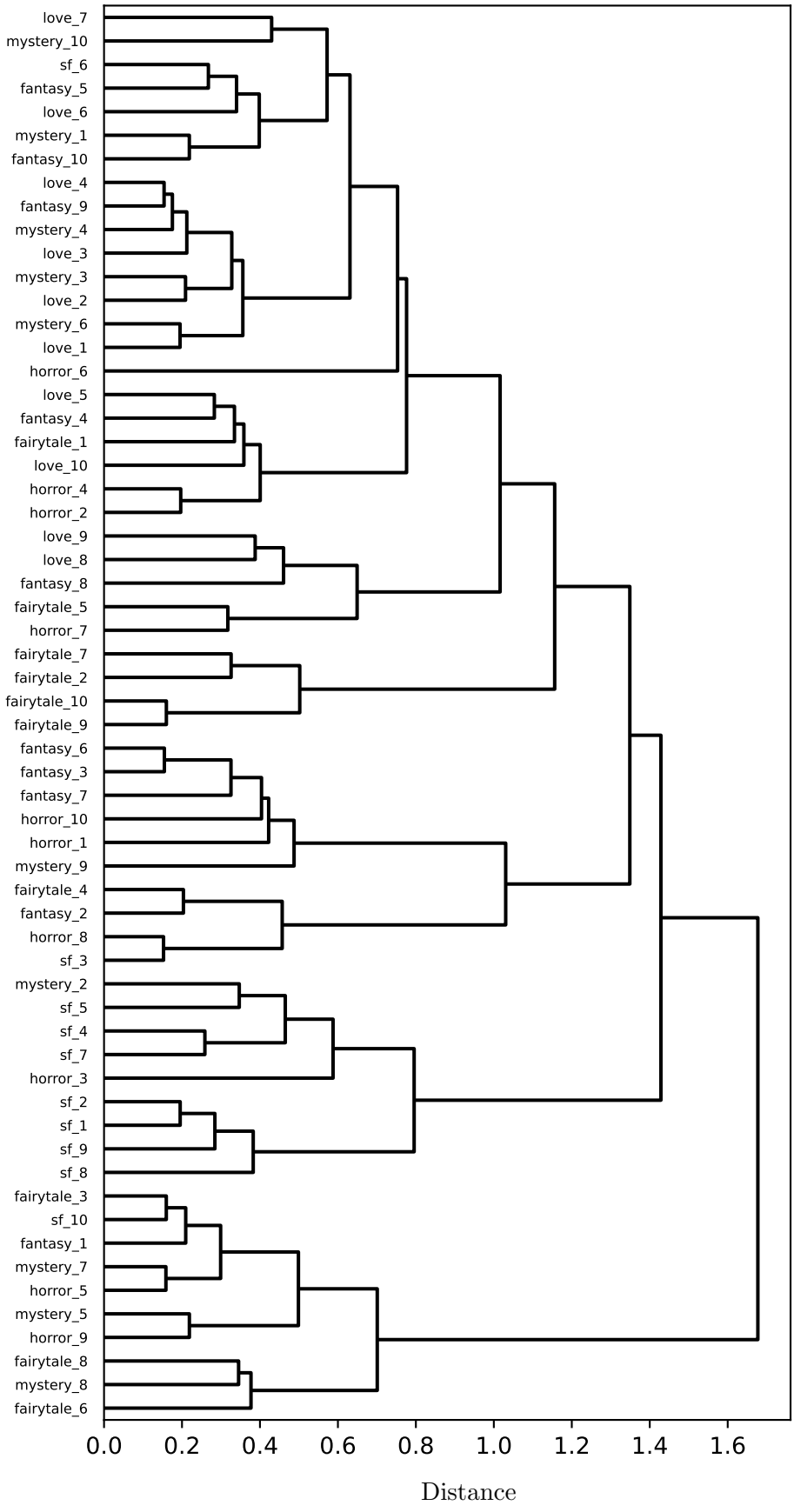

Figure 4.9 Result of dimension reduction by hierarchical clustering(6_60_80%).

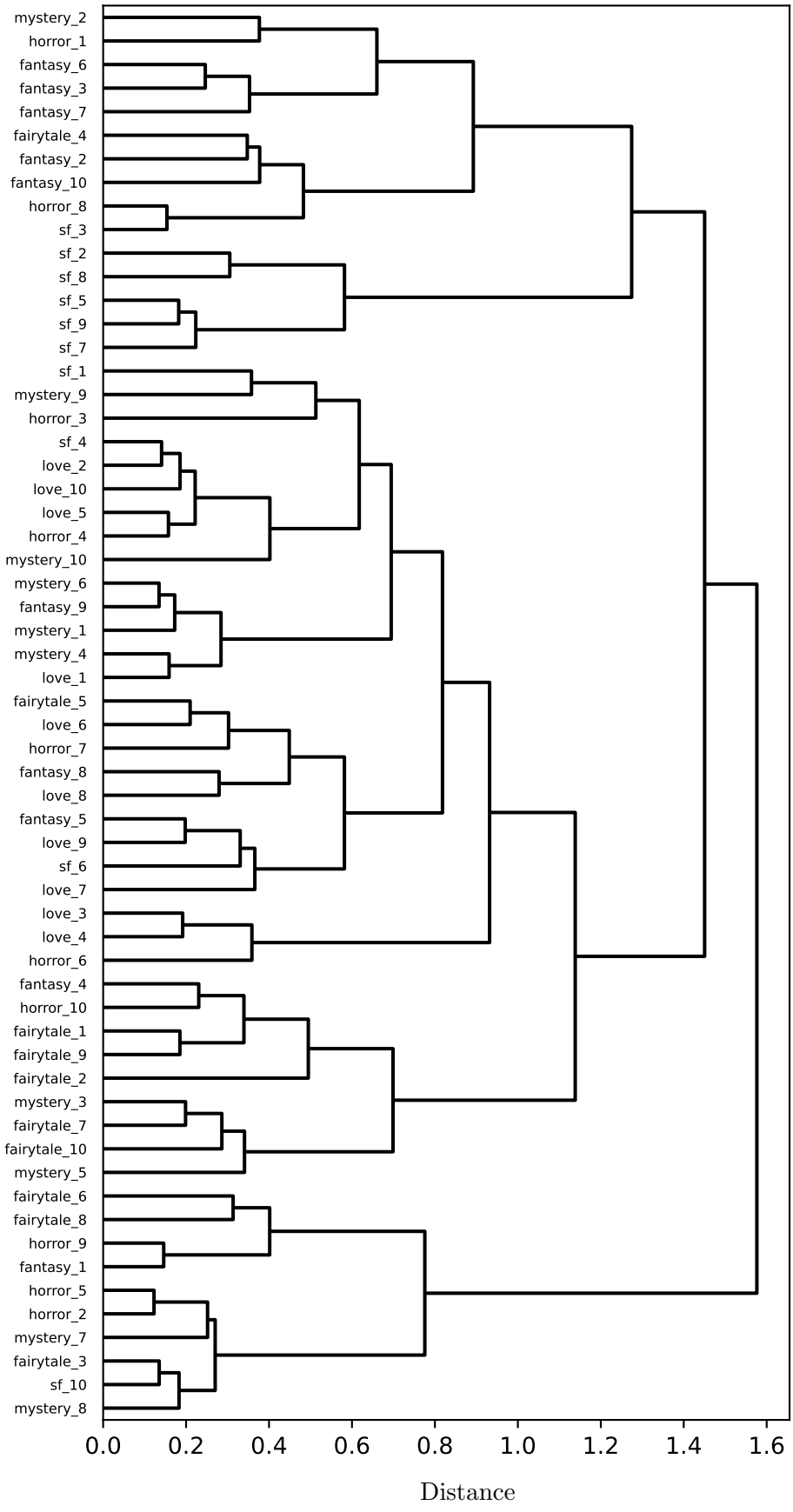

Figure 4.10 Result of dimension reduction by spectral clustering(6_60_80%).

ラフの分割の難しさが計算でき、精度が良くなると考えられる。

4.8.2　デンドログラムの比較

4.7.2 項及び Figure 4.5〜Figure 4.10 より、本研究手法はほかの研究手法に比べクラスター内ではぞれぞれのジャンルに分類されていることがわかる。これは、4.8.1 項の考察からもわかるように、ベクトル空間上で形成されたグラフをもとにクラスタリングしているからだと考えられる。潜在意味解析は単語を統計的に処理しているので、単語の意味を考慮せず次元削減をしているため違うジャンルでも近いクラスターに分類されることがある。また、階層的クラスタリングでも、近い距離の単語をクラスタリングしているだけなので、ほかのジャンルで意味の似ている単語が使われていると近いクラスターに分類されてしまう。単語のクラスタリングの結果から、階層的クラスタリングで文書の分類分けをするため、これらの単語を含む文書は同じクラスターに分類されやすくなる。そのため、潜在意味解析と階層的クラスタリングでは他のジャンルも同じクラスターに属してしまうと考えられる。しかし、スペクトラルクラスタリングは、グラフを生成しグラフの分割の難しさからクラスタリングしているので、クラスターよりも小さいところでの文書分類では同じジャンルの文書が比較的まとまっていると考えられる。

第5章　結論

本研究では、Word2vec とスペクトラルクラスタリングによる次元削減手法を提案し、昨年度研究手法や潜在意味解析との比較によってその有用性を検証した。

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、潜在的意味解析と階層的クラスタリング、スペクトラルクラスタリングによる次元削減を行い、分類問題に適用してから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、条件を変更した 4 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、スペクトラルクラスタリング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、次元削減手法のクラスタリング方法の違いが挙げられた。スペクトラルクラスタリングによる次元削減は階層的クラスタリングと比べ、グラフ分割に基づいたクラスタリング手法で、最適なグラフ分割を求めるための評価関数が設定されており、その評価関数の最適解に対応したある固有値問題の解を用いて分類を行う。それにより、良い結果が得られたのだと判断した。

しかし、本実験を通して 3 つの懸念点が浮かんできた。1 つはすべてのパターンで有用性が確認できなかったことである。デンドログラムと導出された正解率の比較により、ある程度の有用性は確認できたが、ジャンル数が少ない場合の時はクラスター分析のほうが正解率が高くなるという結果が得られた。そのため、すべてのパターンでスペクトラルクラスタリングを使用するのは有用だといえずほかの手法も併用することにより良い結果が得られると思われる。2 つ目は正解率の評価の妥当性に不安がある点である。本研究で使用した評価方法は昨年度の研究である程度の妥当性が確認されているが、それ以上に信頼性がある評価方法を検証できなかった。クラスター 内のデータ構成を考慮した評価指標を考案できれば、更に信頼性がある検証ができると思われる。 3 つ目は本研究手法は固有値問題を解くことから、計算量がデータ数の 3 乗と大きいため、データ数が大きい場合にはスペクトラルクラスタリングを行うと膨大な時間がかかるという問題点がある。

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のメンバーに厚く御礼申し上げます。

参考文献

1) 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

2) NHN TECHORUS Tech Blog, スペクトラルクラスタリング入門, 2017.
https://techblog.nhn-techorus.com/archives/5464（2022 年 10 月 28 日アクセス）.

3) 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.

4) 水野貴明 著, Web API: TheGood Parts, オライリージャパン, 2014.

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 10 月 28 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) Toshinori Sato, Neologism dictionary based on the language resources on the Web for Mecab, 2015.
https://github.com/neologd/mecab-ipadic-neologd（2022 年 10 月 28 日アクセス）.

8) 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.
https://www.gifu-nct.ac.jp/elec/deguchi/sotsuron/hasegawa

9) 瀬尾 慎介, K-means 法による次元削減を用いた文書分類, 岐阜工業高等専門学校電気情報工学科卒業研究報告, 2022
https://www.gifu-nct.ac.jp/elec/deguchi/sotsuron/seo

10) 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/（2022 年 10 月 28 日アクセス）.