特 別 研 究 報 告 題 目

BERTを用いたクラスタ分析による文章分類

Classification of Documents
by Cluster Analysis using BERT

指 導 教 員　　主査 出口 利憲 教授
副査 田島 孝治 准教授

岐阜工業高等専門学校 専攻科 先端融合開発専攻

2022Y12 後藤 貴樹

令 和　06　年　(2024　年)　2　月　01　日

Abstract

The data amount is increasing in proportion to development of technology. Recently, data analysis is important technology in the world. But big data analysis is demands long time. The purpose of this study is to investigate the effectiveness of dimensionality reduction by cluster analysis. This is done by BERT distributed representation and Ward's method clustering. This study uses documents classification as an indicator. The documents used for dimensionality reduction are in nine genres from the livedoor news corpus. Also the distance between documents after dimensionality reduction is used for validation through documents classification. These results show that this method significantly reflects information on proper nouns such as people's names. This feature classifies from a different perspective than latent semantic analysis and may be a more effective dimensionality reduction in some situations.

目 次

第1章　序論

近年、コンピュータやスマートフォンといったインターネットに接続可能な情報端末が老若男女へ普及したことにより、インターネット上のデータは膨大となった。これにより様々な情報を手に入れることが可能になった一方、自身の求めるデータを探し、手に入れることは難しくなった。

これを解決する方法の一つとして、コンピュータを用いてそれら有益なデータのみを抽出するための技術が存在する。そのような技術をデータマイニングといい、それらの中でも対象を日本語や英語など自然言語によって記述された文書などに絞ったものをテキストマイニングという。

本研究はこのテキストマイニングで使用される方法の一つである文書分類において、文書間の類似度を求める技術の効率化を図るために使用される次元削減における手法について提案し、その性質を求めるものである。この研究ではBERTの形態素解析と分散表現を用いて文書中の単語を数値として表現し、それらをクラスタ分析により分類することによって次元削減を行う。本手法での次元削減の有効性、特徴の検討のためそれによる次元削減後の行列での文書分類を行い、それによるデンドログラム、正解率を求める。また、他手法として潜在的意味解析による次元削減を用いた場合の正解率、デンドログラムを比較し、本手法の特徴などを考察する。

第2章　基礎知識

2.1　テキストマイニング

2.1.1　テキストマイニング

テキストマイニングは文書データから情報を取り出すことの総称である。昨今では市場のトレンドを分析するためにSNSなどで製品に対する良い点や悪い点の収集すを行うために利用されている。対象が日本語や英語といった自然言語であり、単語や接続詞には意味や対象が存在するため、数値などに対するデータマイニングより難しいとされている。

2.1.2　形態素解析

形態素は単語や接続詞といった意味を持つ言語の最小単位のことである。形態素解析は文書から品詞、単語などの情報をもとに、それを形態素に分割する処理のことを指す。

2.1.3　MeCab

MeCabとはオープンソースの形態素解析エンジンであり、日本語に対応している。品詞などの情報が記録された辞書に基づき単語を抽出し、形態素解析を行う。今回使用したBERTモデルのトークナイザはMeCabを用いて文章を単語に分割した後にトークン化を行っている。

2.2　自然言語

2.2.1　自然言語

自然言語とは日本語や英語、中国語のように人間が日常的に意思疎通を目的として使用する言語である。対の概念としてプログラミング言語などの人工言語が存在する。これら二つの違いは人工言語がコンピュータを対象としているためコンピュータに解釈しやすいような構造になっており厳格に解釈が定められているのに対し、自然言語は「上手（じょうず)」や「上手（かみて)」などの多義語などの解釈に幅のあるものが存在するという点である。

2.2.2　自然言語の曖昧性

テキストマイニングにおける問題の原因となっているものの一つとして自然言語の曖昧性が挙げられる。その一つとして多義語が挙げられ、これは同じ単語であってもそれぞれ持つ意味が異なることを指している。例として「生物（せいぶつ）」と「生物（なまもの）」などが挙げられる。このように単語が二つ以上の意味を持つ別の単語を同じものだと判断してしまうという課題がテキストマイニングに存在する。また、「赤い屋根の大きい家」というものに対しても「赤い」「屋根の大きい家」なのか「赤い屋根の」「大きい家」なのか判別がつきにくいといったような問題が生じる。これは自然言語の解釈の幅の広さに起因した問題である。

2.3　機械学習

2.3.1　機械学習

機械学習とは、マシンラーニングとも呼ばれるデータを分析する技術の一つである。この技術の一つとしてニューラルネットワークが挙げられる。大量のデータを用いて機械にデータから出力のパターンを学習させることによって重みなどの要素を最適化し、問題を解決する手法のことを指す。この技術には学習と推論の二つの過程が存在する。

2.3.2　学習

学習は機械学習において入力されたデータから望ましい出力を得るために行われる過程の一つであり、教師あり学習と教師なし学習が存在する。学習を行うことでモデルが入力に対する出力のパターンを導くことができ、問題として与えられたものに対する出力の精度が上昇する。

2.3.3　教師あり学習

教師あり学習とは機械学習における学習方法の一つである。入力データとそのデータに対応する正解のデータを大量に用意し、モデルに与える。これを用いてモデルは入力データと正解データの相関を導くことにより学習を進める。

2.3.4　教師なし学習

教師なし学習は教師あり学習と異なり、学習データに正解のデータを与えない状態で学習させる方法である。教師あり学習は正解のデータと入力データの相関を求めることにより学習を行うが、教師なし学習では与えられたデータを比較することによって傾向を求めることによって機械自身がデータの法則を学習する。

2.3.5　特徴量抽出

機械学習で問題を解決する際に必要となるのが特徴量抽出である。これは、数値として表現されていない自然言語などのデータを数値に変換する処理である。これらを取得することによってモデルにその情報をデータとして与えることが可能になる。また、モデルの性能の向上を目的として数値化したデータに次元削減を行い、データを圧縮するといったアプローチを行うこともある。教師あり学習では人間がこれを与えることが多く、教師なし学習ではモデルがこれを見出すことが多い。

2.3.6　ニューラルネットワーク

ニューラルネットワークとは人間の脳を数式により模したモデルである。これは人間の神経回路のモデル化に起源を持つ数理モデルであり、機械学習で用いられることが多い。このモデルは複数のレイヤーの組み合わせにより構成され、その層の一つ一つは何らかの変換を行う。各層はそれぞれ入力中に対し線形化を行い、例として Figure 2.1 に示すシグモイド関数[1)]の様な活性化関数による出力を返すようになっている。これは人間のニューロンの発火現象を擬似的に再現したものである。昨今機械学習の分野で話題になっているディープラーニングはこのニューラルネットワークの層を多くしたものであるため、日本語では深層学習と言われている。

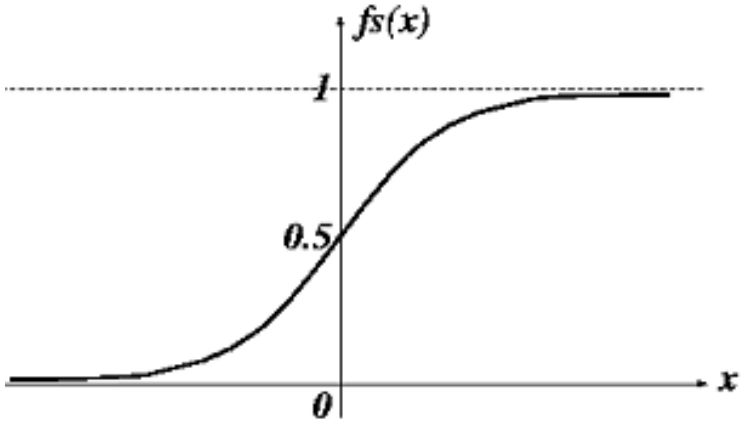


Figure 2.1: Sigmoid function.

第3章　実験で使用した技法

3.1　BERT

3.1.1　BERT

BERT とは Bidirectional Encoder Representations from Transformers の略称であり、2018 年に Google によって発表された自然言語処理モデルであり [2]、再帰型ニューラルネットワークがベースになっている。再帰型ニューラルネットワークと比較して Figure 3.1 のような注意機構を有しており、これは入力の一つであるトークンを処理する際に他のトークンを直接参照する。また、特徴として事前学習モデルであることが挙げられ、トークンの一部を隠して予測を行う Masked Language Model と Next Sentence Prediction という二つの文章がつながっているか否かを予測する方法の二つで学習が行われることが挙げられる [3]。これにより従来では不可能であった文脈を考慮したトークンの分散表現が可能になっている。

3.1.2　事前学習

事前学習とは機械学習における学習の段階の前に行う学習のことであり、大量のラベルなしデータを用いてモデルを学習させることが多い。今回用いた東北大学研究チームが作成した BERT モデル [4] は日本語 Wikipedia の全ての記事を用いた事前学習が施されており、これによって汎用的な日本語のパターンが既に学習されている。これによりデータの特徴量の抽出が良好に行われる。

3.1.3　ファインチューニング [3]

ファインチューニングとは特定のタスクに対する精度を向上させるための処理である。少数のラベル付き学習データを用いて BERT を学習させると共に分類器についても学習させることでその特定のタスクのみ精度が向上する。

3.1.4　トークン化 [3]

BERT は複数の言語タスクに対応出来るような入力形式で設計されている。例として「明日は自然言語処理の勉強をしよう。」という文章は「明日」「は」「自然」「言語」「処理」「の」「勉強」「を」「しよ」「う」「。」と言ったように分割される。そしてこのとき

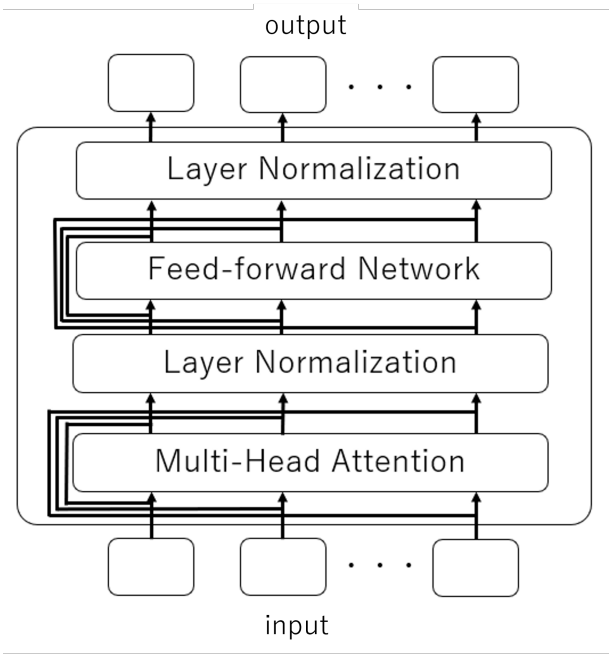


Figure 3.1: Structure of BERT

先頭と末尾に特殊なトークンが追加される。先頭に追加されるものを CLS トークンといい、末尾に追加されるものを SEP トークンという。

3.1.5 CLS トークン[3)]

CLS トークンとは BERT によってトークン化された文章の先頭に配置されている特殊なトークンである。これは BERT の事前学習時に次文予測で利用されるトークンであり、これに対する BERT の出力はトークンの分散表現ではなく文章の分散表現として用いられる。

3.1.6 SEP トークン[3)]

SEP トークンは BERT によってトークン化された文章の末尾に配置される特殊なトークンである。これは末尾のみではなく、文章が二つ連続する場合も付加される。例として「今日は雨だ。家にいよう。」と言う文章なら「今日」「は」「雨」「だ」「。」「SEP」「家」「に」「いよ」「う」「。」というように文章の継ぎ目にも配置される。

3.2 計算手法

3.2.1 Tf-Idf

Tf-Idf は文書内にある単語の重要度を示したものである。単語の出現頻度と単語の情報量の積によって求められる。N 個の文書中、各文書 $d_i(i = 1, 2, ..., N)$ にて文書 d_i において M 種類の単語が現れたとき、各単語を $t_{ij}(j = 1, 2, ..., M)$、t_{ij} が含まれた文書数を n_t とする。その時、単語の出現頻度 Tf、単語の情報量 Idf、$TfIdf$ は次の式で示される。

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

$$Idf_j = \log \frac{N}{n_t} \tag{3.2}$$

$$TfIdf_{ij} = Tf_{ij} \times Idf_j \tag{3.3}$$

3.2.2 cos 類似度

cos 類似度はベクトル空間モデルにおいて、文書の比較に用いられる計算法である。二つのベクトル $\vec{a}$ と $\vec{b}$ の cos 類似度は次の式で表される。

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

3.2.3 特異値分解

特異値分解は Figure 3.2 のように任意の $m \times n$ の実行列 A が $m \times m$ 行列 U と $m \times n$ の対角格行列 Σ と $n \times n$ 行列 V の積に分解する手法である。この時、Σ は実行列 A の特異値行列であり、特異値が重要度順に並んでいる。

3.2.4 潜在的意味解析

潜在的意味解析は特異値分解を用いて文書行列を統計学的な次元圧縮を行う手法である。Figure 3.2 で示したようように行列は 3 つの行列によって表現される。その中の一つである特異値行列 Σ の特異値を参照し、その影響度が低い順に U、Σ、V を任意の値まで次元削減することによって元行列 A を次元削減する方法である。これにより重要な情報を保持したまま次元削減が可能となる。このように k 次元まで削減した左特異ベクトルである U_k を式 (3.5) のように元の行列 A と乗じることによって次元削減後行列 A_k を

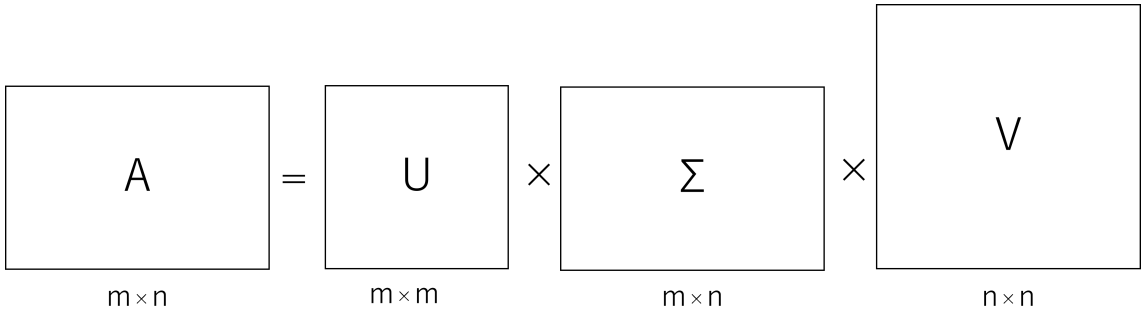


Figure 3.2: Singular value decomposition

算出する。

$$A_k = U_k A \tag{3.5}$$

3.2.5 階層的クラスタ分析

クラスタ分析は与えられたデータをいくつかの集合に分類するデータ解析手法である。その中でも階層的クラスタ分析は Figure 3.3 のようにデータ間の距離などの類似度に基づいてグループ分けを行うことによって順次クラスタを形成するものを指す。順次クラスタを形成していく特性上、Figure 3.4 のようなクラスタリングの過程を確認できるデンドログラムといわれる樹形図を作成することができる。

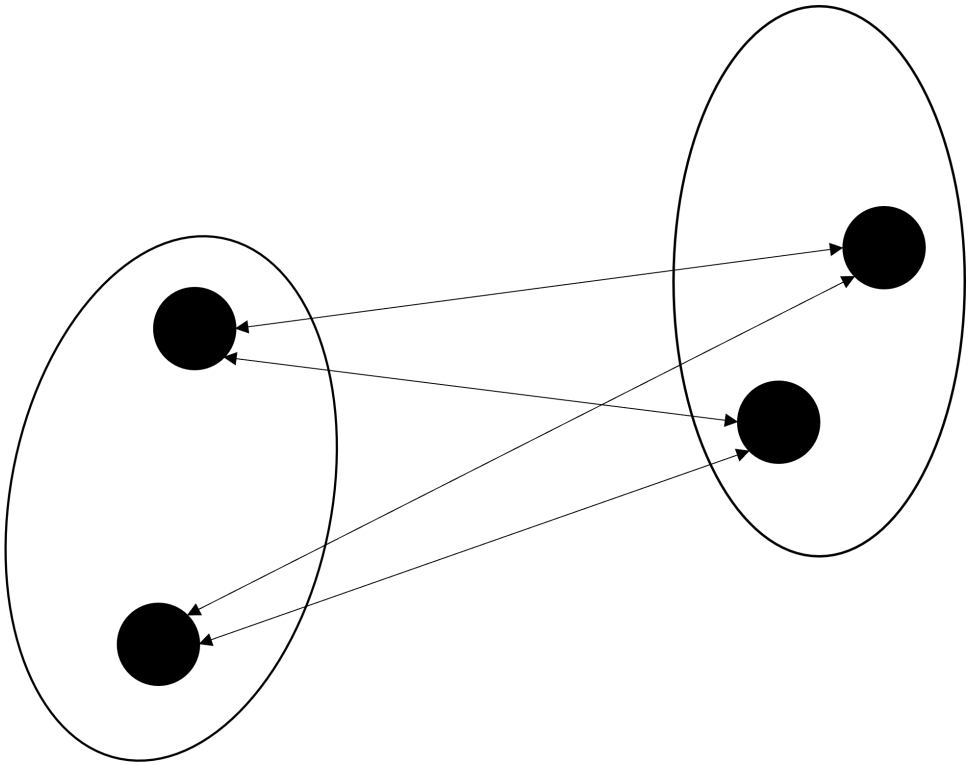


Figure 3.3: Cluster analysis

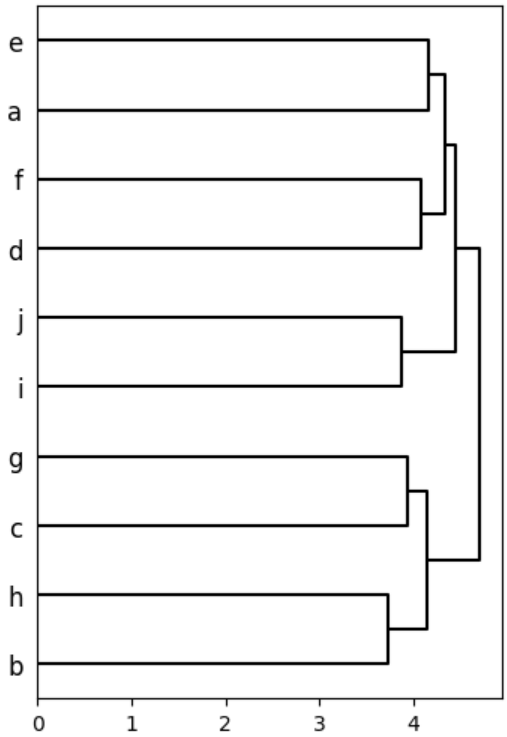


Figure 3.4: Dendrogram

第4章　実験方法

4.1　実験の概要

本研究はBERTを用いた埋め込み表現とクラスタ分析を用いた次元削減の有効性を検討することを目的としている。そのためクラスタ分析を用いた次元削減の他、Tf-Idfと潜在的意味解析による次元削減と比較する。

この研究で用いる文書はlivedoorニュースコーパス[5)]を用いて取得した9ジャンルのニュース記事である。あらかじめ付与されているジャンルをもとに文章分類後の結果と照らし合わせることによって正解率とし、これとデンドログラムを次元削減の有効性の指標とする。

この研究ではlivedoorニュースコーパスを用いて9ジャンルの文書データをそれぞれ10文書ずつ抽出して用いる。

4.2　実験準備

4.2.1　実験環境の構築

本実験はGoogleColaboratory上で実行した。また、その環境の構築のために以下の項目を行った。

- livedoorニュースコーパスからの文書データの取得
- 文書データの前処理
- 学習済みBERTモデルの取得

4.2.2　livedoorニュースコーパスを用いた文書データの取得

今回文章分類の対象となるデータはlivedoorのニュースとして掲載されていた文書であり、株式会社ロンウィットが収集し配布したデータである。今回はこれの本文を抜き出して使用する。この文書データは以下の9ジャンルに分類されており、これとクラスタリング結果を比較することで正解率を算出する。

- dokujo-tsushin
- it-life-hack
- kaden-channel
- livedoor-homme

- movie-enter
- peachy
- smax
- sports-watch
- topic-news

クラスタ分析による次元削減の有効性を検討するために文章分類を行うため、このようにあらかじめジャンル分けされた文書データを利用する。これらジャンルからそれぞれ10文書を抽出して用いる。

4.2.3 文書データの前処理

今回はlivedoorニュースコーパスから入手した9ジャンルの文書を分類の対象とするが、これら文書には日時や著者名など文書のジャンルに関連しない情報が含まれているため、それら情報を除外して本文のみを使用した。

4.2.4 学習済みBERTモデルの取得

Transformersをインポートして東北大学研究チームが作成したBERTの日本語モデル[4)]を取得した。取得したBERTモデルの詳細をFigure 4.1に示す。 このBERTモデルで出力されるベクトルは768次元であり、入力として与えることのできる文書のトークンは最大で512個であることがわかる。また、このBERTモデルは事前に日本語のwikipediaの全ての記事を用いて事前学習されたものである。

4.3 次元削減

4.3.1 クラスタ分析による次元削減

以下の手順でクラスタ分析による次元削減を行った。

1. BERTトークナイザを用いトークンに分割する
2. 東北大学BERTモデルを用いてトークンの分散表現を取得する
3. 取得した分散表現にウォード法を用いたクラスタ分析を行う
4. 文書D_Nとクラスタ番号C_Mを用いて式(4.1)のようにクラスタ分析によって得たトークンそれぞれのクラスタ番号を基に、文書内に存在するトークンの種類の割合を示した行列を作る

```
BertConfig {
  "_name_or_path": "cl-tohoku/bert-base-japanese-whole-word-masking",
  "architectures": [
    "BertForMaskedLM"
  ],
  "attention_probs_dropout_prob": 0.1,
  "classifier_dropout": null,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pad_token_id": 0,
  "position_embedding_type": "absolute",
  "tokenizer_class": "BertJapaneseTokenizer",
  "transformers_version": "4.16.2",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 32000
}
```


Figure 4.1: BERT config

式 (4.1) における割合 r_{ij} は文書 D_i 内のトークン数を α_i とし、D_i 内に含まれ、なおかつクラスタ C_j に含まれるトークン数を α_{ij} としたとき式 (4.2) により示される。次元削減前は文書の分散表現は文書数 × トークン数 ×768 個の数値で表されたが、この処理を行うことによって文書数 × クラスタ数個にまで数値を削減される。本研究では文書内のすべてのトークンを用いた場合と動詞、名詞トークンのみを用いた場合の二通りを行う。東北大学のトークナイザには品詞の付加情報が存在しないため、品詞情報を持つ MeCab を用いて同様の文書をトークンに分割し、その結果をもとにして BERT のトークンに品詞情報を付加した。本研究では使用文書のジャンル数が 9 であるため、少なくとも 10 クラスタ以上に分けるべきだと考えた。一方、クラスタ数の上限は比較対象である潜在的意味解析の上限である文書数、今回は 90 次元に設定した。

$$CD = \left(\begin{array}{c|cccccc} & C_1 & C_2 & C_3 & C_4 & ... & C_M \\ \hline D_1 & r_{11} & r_{12} & r_{13} & r_{14} & \cdots & r_{1M} \\ D_2 & r_{21} & r_{22} & r_{23} & r_{24} & \cdots & r_{2M} \\ D_3 & r_{31} & r_{32} & r_{33} & r_{34} & \cdots & r_{3M} \\ D_4 & r_{41} & r_{42} & r_{43} & r_{44} & \cdots & r_{4M} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ D_N & r_{N1} & r_{N2} & r_{N3} & r_{N4} & \cdots & r_{NM} \end{array}\right) \tag{4.1}$$

$$r_{ij} = \frac{\alpha_{ij}}{\alpha_i} \tag{4.2}$$

4.3.2 Tf-Idf と潜在的意味解析による次元削減

使用した文書ごとにトークンに分割し、それを基に Tf-Idf を算出した。この Tf-Idf に対して潜在的意味解析を行い、文書数 × 指定した数にまで次元削減を行った。

4.4 正解率の計算

4.4.1 クラスタの割り当て

次元削減の有効性や特性を検討するため、文書分類として BERT のトークンベクトルとクラスタ分析を組み合わせた次元削減した文書、Tf-Idf と LSA で次元削減した文書をそれぞれ実験で用いた文書のジャンル数 L とクラスタ数が等しくなるようクラスタリングする。これにより一つの文書につき一つのクラスタ番号が付与される。使用文書のジャンルを G_i、一ジャンルごとに用いた文書数 S ($S = \frac{N}{L}$) としたとき、ジャンル G_i、j 文書目に付与されたクラスタ番号 c_{ij} は式 (4.3) の行列によって表すことができる。このジャンル G_i の行のクラスタ番号の最頻値をその元ジャンルの番号として扱い、重複を無くすことによって元のジャンルとクラスタ番号に全射の関係を作る。

$$GD = \left(\begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & \dots & S \\ \hline G_1 & c_{11} & c_{12} & c_{13} & c_{14} & \dots & c_{1S} \\ G_2 & c_{21} & c_{22} & c_{23} & c_{24} & \dots & c_{2S} \\ G_3 & c_{31} & c_{32} & c_{33} & c_{34} & \dots & c_{3S} \\ G_4 & c_{41} & c_{42} & c_{43} & c_{44} & \dots & c_{4S} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ G_L & c_{L1} & c_{L2} & c_{L3} & c_{L4} & \dots & c_{LS} \end{array}\right) \tag{4.3}$$

4.4.2 正解率の計算

クラスタリングによって文書ごとに与えられたクラスタ番号とジャンルに対するクラスタ番号を比較し、番号が一致したものを正解した文書として扱う。これをカウントしたものを正解文書数 n として使用して、全文書数 N で除算することで式 (4.4) のように正解率を算出した。

$$accuracy = \frac{n}{N} \tag{4.4}$$

第5章　実験結果

5.1　実験結果の評価法

本実験ではクラスタ分析による次元削減に加え Tf-Idf と潜在的意味解析を用いた次元削減による文書分類を行い、それら正解率を比較することでクラスタ分析による次元削減の有効性を評価する。これによりクラスタ分析による次元削減の特性と他の方法を用いた次元削減を確認し、それらと比較する。

5.1.1　クラスタ数による正解率の推移

9 ジャンルの文書をそれぞれ 10 文書ずつ使用し、クラスタ分析による次元削減を用いて文書分類を行ったときの正解率の推移を Figure 5.1 に示す。わずかだがクラスタ分析時のクラスタ数を増やすほど正解率もそれに伴い上昇することがわかる。クラスタ数 30 での文書ごとの類似度をデンドログラムにしたものを Figure 5.2 に示す。

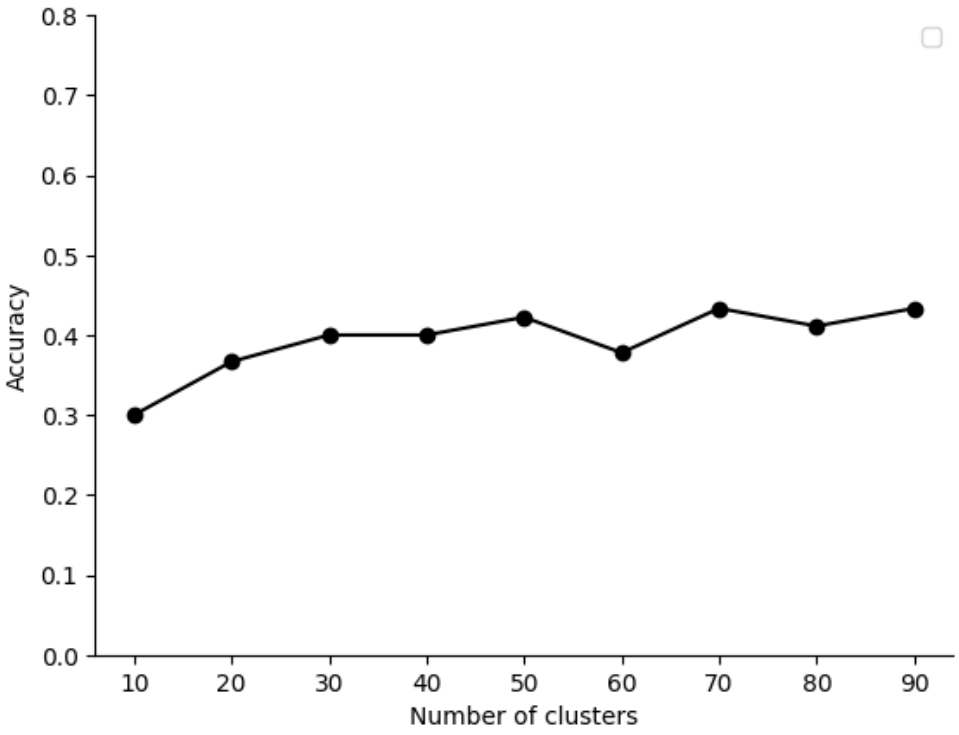


Figure 5.1: Accuracy of cluster analysis

Figure 5.2 から、この時の分類結果では smax、it-life-hack と kaden-channel の文書類似度が高いとされていることがわかる。ではなぜこれら 3 ジャンルの文書類似度が高くなったのか、クラスタ分析時に分類した 30 クラスタを Figure 5.3 に示す。

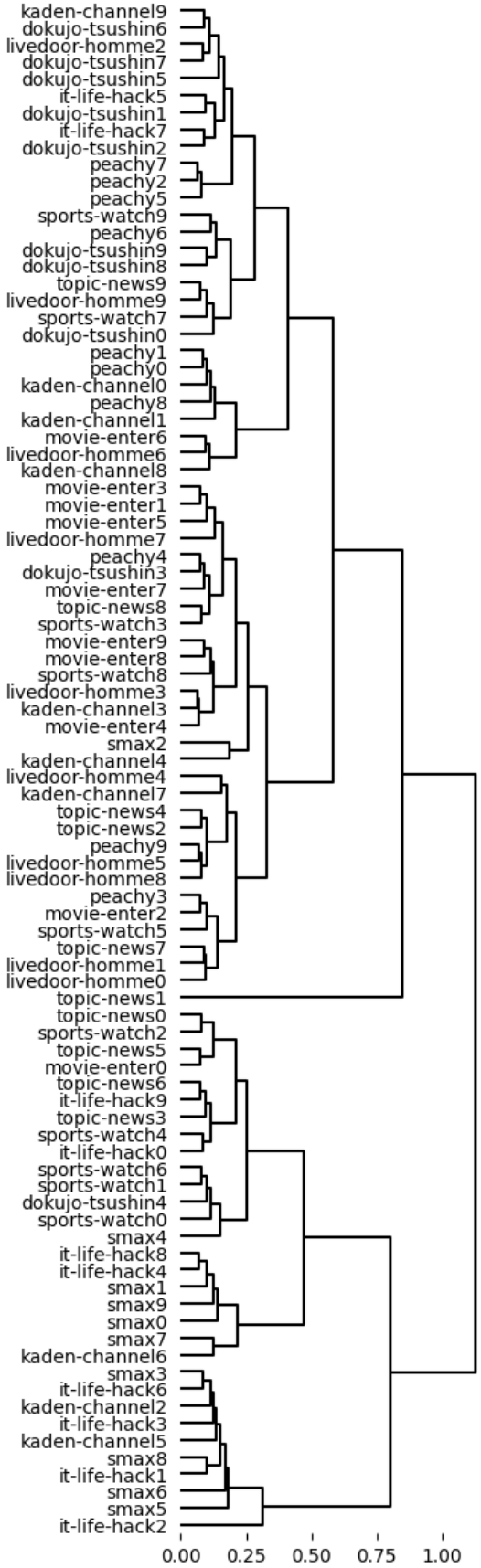


Figure 5.2: Similarity among documents by cluster analysis(30 clusters)

```
tokens_num0 ['。', '。', '。', '[SEP]', '。', '。', '。', '。', '。', '。', '。', '。'
tokens_num1 ['も', 'も', 'も', 'でも', 'も', 'も', 'は', 'も', 'も', 'も', 'も', 'に',
tokens_num2 ['情報', '番組', '雑誌', 'など', '注目', '始め', '専門', '店', '増え', 'か
tokens_num3 ['この', '先', '##日', 'この', '世界一', 'ふ', '##くら', '柔らか', '優しい
tokens_num4 ['##it', '##プティ', '##y', 'new', '##s', 'jp', 'OS', 'Office', 'ソフト',
tokens_num5 ['トー', '##スト', 'Pa', '##in', 'Pet', 'Pa', '##s', 'パン', '##パ', '各地
tokens_num6 ['[CLS]', 'ところ', '朝', '夕', 'フレンチ', '江ノ島', 'LO', '##N', '##CA',
tokens_num7 ['の', 'の', 'の', 'の', 'の', 'の', 'の', 'の', 'の', 'の', 'の', 'の', '
tokens_num8 ['、', '、', '、', '、', '、', '、', '、', '、', '、', '、', '、', '、', '
tokens_num9 ['に', 'に', 'に', 'に', '中', 'に', 'に', 'に', 'に', 'に', 'に', 'に', '
tokens_num10 ['が', 'が', 'が', 'が', 'が', 'が', 'が', 'が', 'が', 'が', 'が', 'が',
tokens_num11 ['です', 'でし', 'でし', 'まし', 'です', 'まし', 'ます', 'まし', 'ます',
tokens_num12 ['。', '。', '。', 'だ', '。', 'だ', 'だ', '。', '。', '。', '。', '。',
tokens_num13 ['で', 'は', 'だろ', 'う', 'で', 'は', 'でしょ', 'う', 'でしょ', 'う', 'た
tokens_num14 ['ん', 'か', '?', 'ん', 'けど', 'か', '?」', 'か', '?', 'ので', 'か', 'の
tokens_num15 ['た', 'た', 'た', 'た', 'た', 'た', '生まれ', 'た', 'た', 'た', 'た', '#
tokens_num16 ['や', 'や', 'な', 'な', 'な', 'や', 'な', 'という', 'こんな', 'そんな',
tokens_num17 ['よう', 'よう', 'あり', 'いえ', 'ない', 'ない', 'こと', 'ない', 'ある',
tokens_num18 ['さ', 'れ', 'て', 'いる', 'いる', 'さ', 'れ', 'き', '##っ', '##く', 'れる
tokens_num19 ['は', 'は', 'は', 'は', 'は', 'は', 'は', 'は', 'は', 'は', 'は', 'は',
tokens_num20 ['「', '「', '「', '「', '「', '「', '「', '「', '「', '「', '「', '「',
tokens_num21 ['と', 'と', 'し', 'と', 'と', 'と', 'と', 'と', 'と', 'と', 'と', 'と',
tokens_num22 ['28', '6', '33', '36', '30', '10', '30', '20', '三', '20', '30', '76', '
tokens_num23 ['」', ')」', 'を', '」', '」', '」', '」', '」', '」', '」', '」', '」',
tokens_num24 ['・', '・', '・', '/', '・', '・', '##・', '##ダ', '##・', '・', '・', '
tokens_num25 ['(', '(', '歳', ')', '月', '人', '(', '歳', ')', '人', '(', '歳', ')', '
tokens_num26 ['で', 'で', 'で', 'で', 'で', 'で', 'で', 'で', 'で', 'で', 'で', 'で',
tokens_num27 ['て', 'て', 'て', 'て', 'て', 'て', 'て', 'て', 'て', 'て', 'て', 'て',
tokens_num28 ['を', 'を', 'を', 'を', 'を', 'を', 'を', 'を', 'を', 'を', 'を', 'を',
tokens_num29 ['し', 'て', 'する', 'し', 'し', 'し', 'する', 'し', 'する', 'し', 'し',
```

Figure 5.3: Classification of cluster analysis(30 clusters)

Figure 5.3 からわかるようにクラスタ分析によってトークンはそれぞれ品詞、句読点、数字に分類される。このような多くの文書で用いられるもののクラスタを生成し、それを基に文書分類をおこなったところでジャンルがあまり関係しない分類になってしまう。その一方、tokens_num4 のように「OS」や「Office」といったモバイル機器やガジェットに関連するトークンがが多く含まれるものがある。it-life-hack や smax はパソコンやスマートフォン、モバイルガジェットなどの記事を多く掲載している。そのためこれら文書はこの観点からの比率が高くなり、文書、クラスタ間の距離が近くなったと考えられる。クラスタ分析によるトークン分類の理想としては tokens_num4 のようにジャンルごとにトークンが分類されることである。

全てのトークンを用いてクラスタ分析を行ったときに SEP トークンや句読点など文書ならば必ず使用するものが集まったクラスタができてしまうという欠点を解消するため、本文文書から動詞と名詞のみを抽出してクラスタ分析を行った。その結果を Figure 5.4 に示す。文書から動詞と名詞を抜き出して行った場合ではクラスタ数 40 の時に正解率が下がるが、全体的にはすべてのトークンを用いたときと正解率の差はほぼないといえる。しかし、句読点や SEP トークンといったクラスタが発生していないのに正解率は上昇しない。この時のデンドログラムを Figure 5.5 に示す。

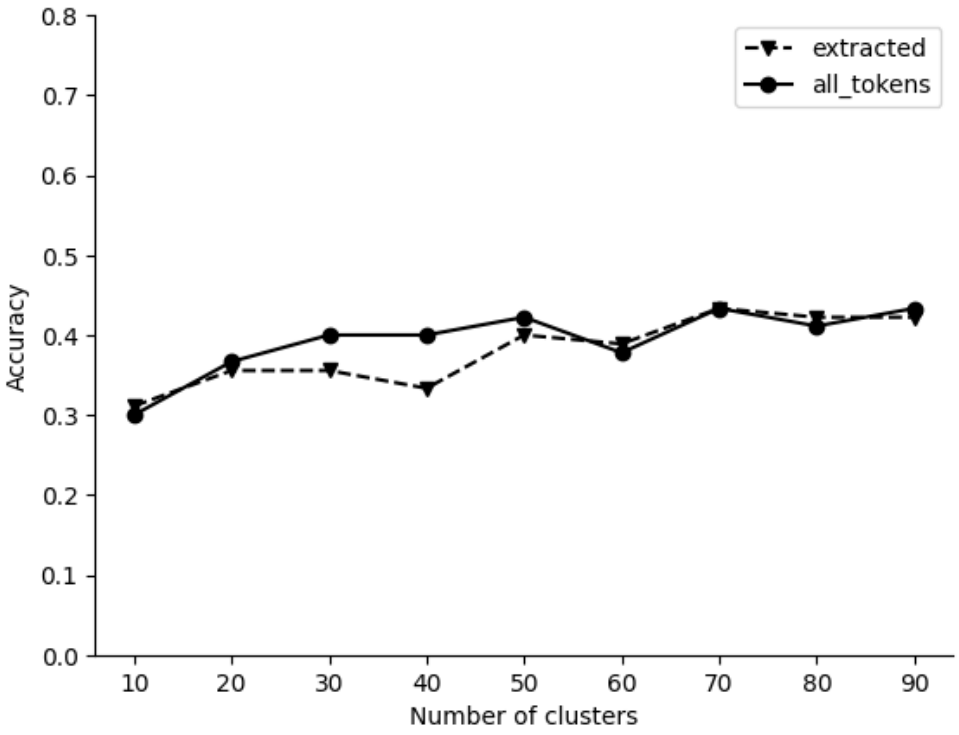


Figure 5.4: Accuracy of cluster analysis(extracted verb and noun)

Figure 5.5 のデンドログラムからはすべてのトークンを用いたとき同様に it-life-hack、kaden-channel、smax は一つのクラスタとして扱われるように、次元削減後のそれら文書は他の文書に比べて類似していることが読み取れる。一方、すべてのトークンを利用したものとの差として、映画のニュースを幅広く扱う movie-enter と男性用の Web マガジンである livedoor-homme の次元削減後行列は一つにまとまっている。ではなぜこのような差が現れたのかを調べるため、Figure 5.6 にクラスタ分析時にどのようにトークンが分類されたかを示す。Figure 5.6 からは Figure 5.2 と同様に「OS」、「Office」が含まれたクラスタがある。そして句読点などを除いたことによって動詞や名詞といったものの分類を行ってはいるのだが、「する」やその変化形のクラスタ、「よう」、「こと」などのクラスタが生成されてしまい、やはり大きく意味を持たないクラスタができてしまうことがわかる。そのため動詞と名詞を抽出した場合もほぼ同様の正解率になってしまったと考えられる。

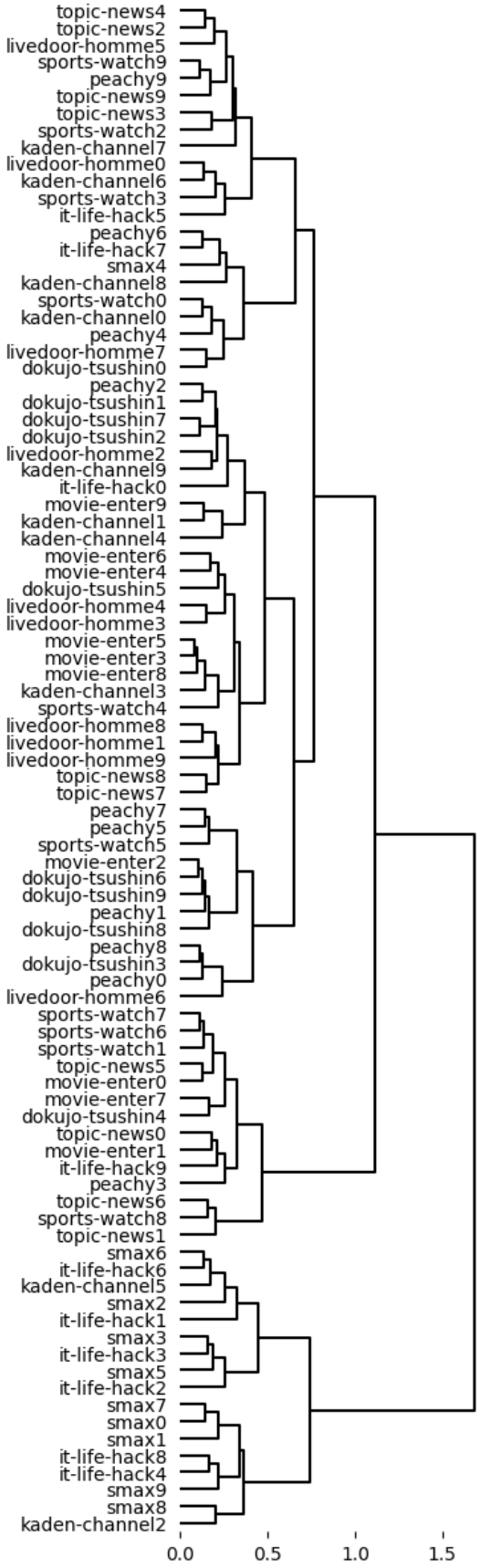


Figure 5.5: Similarity of documents by cluster analysis(30 clusters)

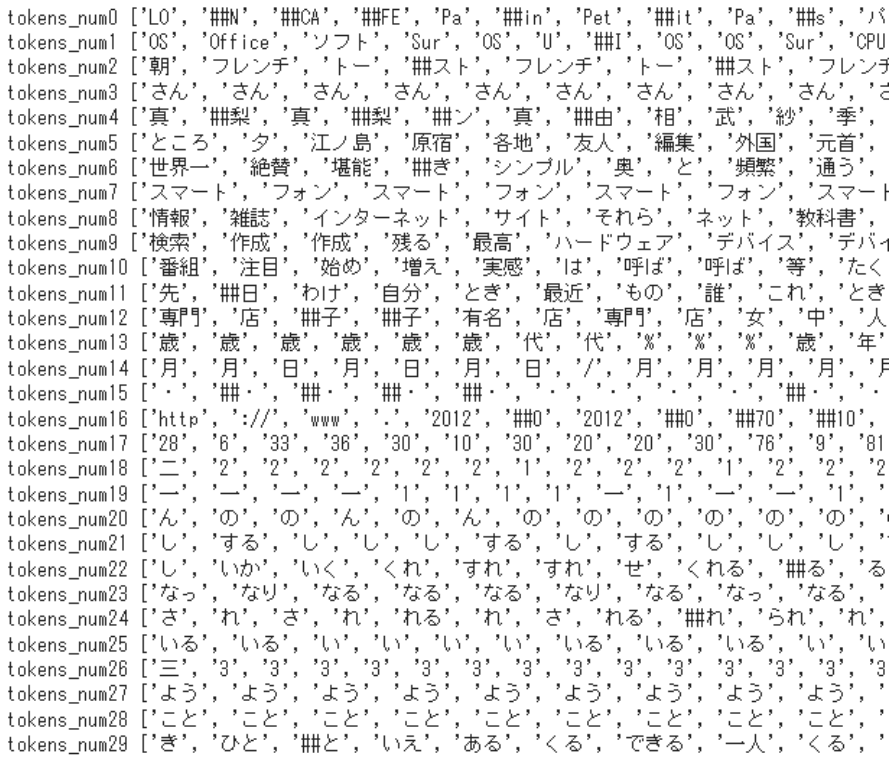


Figure 5.6: Classification of cluster analysis(30 clusters,extracted verb and noun)

5.1.2 ジャンル数による正解率の推移

Figure 5.7 にジャンル数が 5、7、9 の時の正解率の推移を示す。このグラフから、5 ジャンルで行った場合はクラスタ数に関係なく 0.4 から 0.5 ほどの正解率を維持していた。その一方、7 ジャンルと 9 ジャンルを用いて行ったものは正解率にほとんど差が無く、共にクラスタ数が増加するほどに正解率が上昇している。

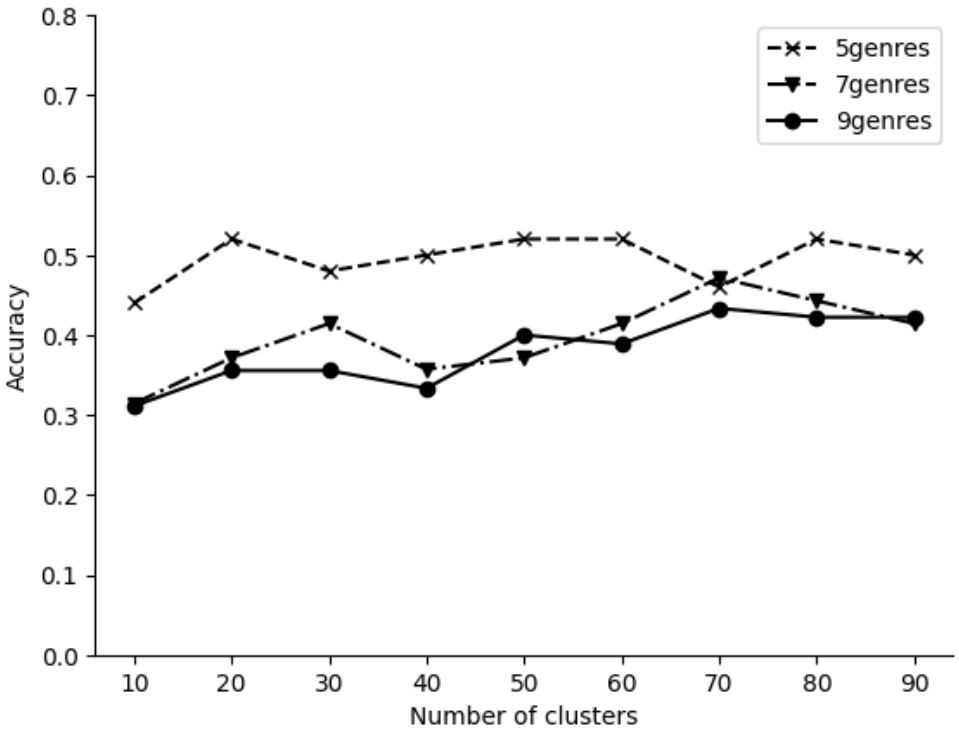


Figure 5.7: Accuracy difference depending on the number of genres

5.1.3 文書数による正解率の推移

ジャンル数 7 で 1 ジャンルごとの文書数を 10、20、30 に変化させ文章分類を行なった正解率の推移を Figure 5.8 に示す。この結果から最も使用文書が少ない 10 文書の正解率が他に比べ少し高いように思えるが、ジャンルごとのサンプルが少ないことによる誤差の範疇だと考えられる。そのため 30 文書用いた場合の 60 クラスタ以降での正解率とほぼ等しい結果となっている。

5.2 考察

5.2.1 クラスタ分析におけるクラスタ数

Figure 5.4 の結果から、すべてのトークンを用いたクラスタ分析と動詞や名詞を抽出して行ったクラスタ分析の双方でクラスタ数の増加に伴い正解率が上昇していることがわかる。これはクラスタ数が増加するほど特徴的なトークンの集まりがさらに細分化されることで、例えば smax や it-life-hack などの似たジャンルの文書の区別が可能になっていくためだと考える。

ここで潜在的意味解析と正解率を比較する。潜在的意味解析の次元数をクラスタ分析と同じく 10 から 90 まで変化させ、正解率を求めたものを Figure 5.9 に示す。

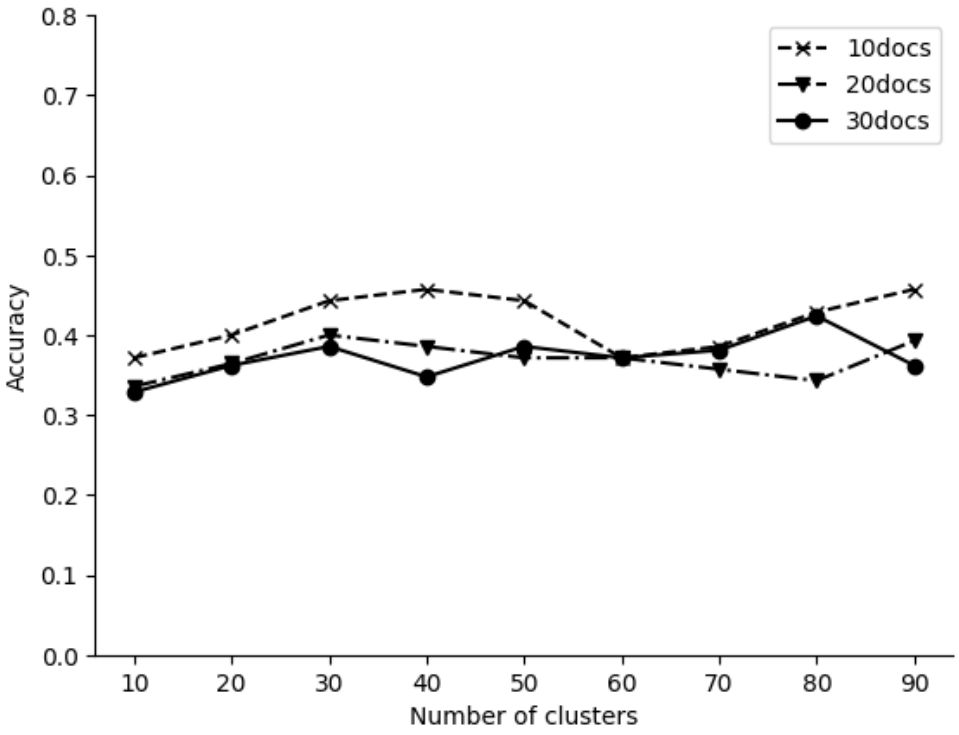


Figure 5.8: Difference in accuracy depending on documents number

この比較から、次元数を 10 から 50 ほどに削減した場合、潜在的意味解析を用いて次元削減を用いた場合の方が正解率が高いことがわかる。一方、それ以降は正解率に大きな差はない。この二つの方法で 90 次元に削減したときのデンドログラムを Figure 5.10 に示す。この二つのデンドログラムを比較すると topic-news に関する違いが見受けられる。topic-news は政治家や芸能人の発言を多く取り上げるニュースサイトである。このジャンルについて、クラスタ分析は特に sports-watch というスポーツ関連の記事とクラスタ間の距離が近い結果を示しているが、潜在的意味解析ではそうではなく、dokujo-tsushin や peachy といった文書が多いクラスタとの距離が近くなっている。sports-watch と topic-news の記事の共通点としてはスポーツ選手、芸能人などの固有名詞が頻繁に出現するという特徴があり、クラスタ分析時に固有名詞トークンのクラスタが生成されたためと思われる。実際に「孫正義」、「千原ジュニア」など有名人と「落合監督」、「田中将大」は同じクラスタに属した。また、「記者」、「インタビュー」といった人物へのインタビュー形式の記事に用いられる単語も一つのクラスタにまとめられていた。

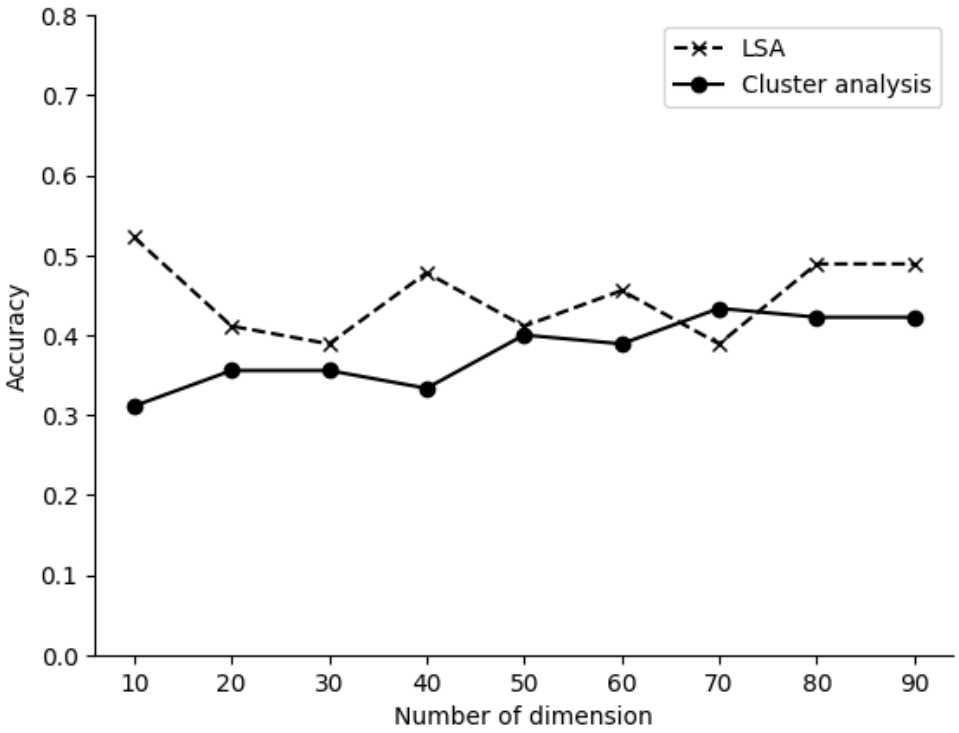


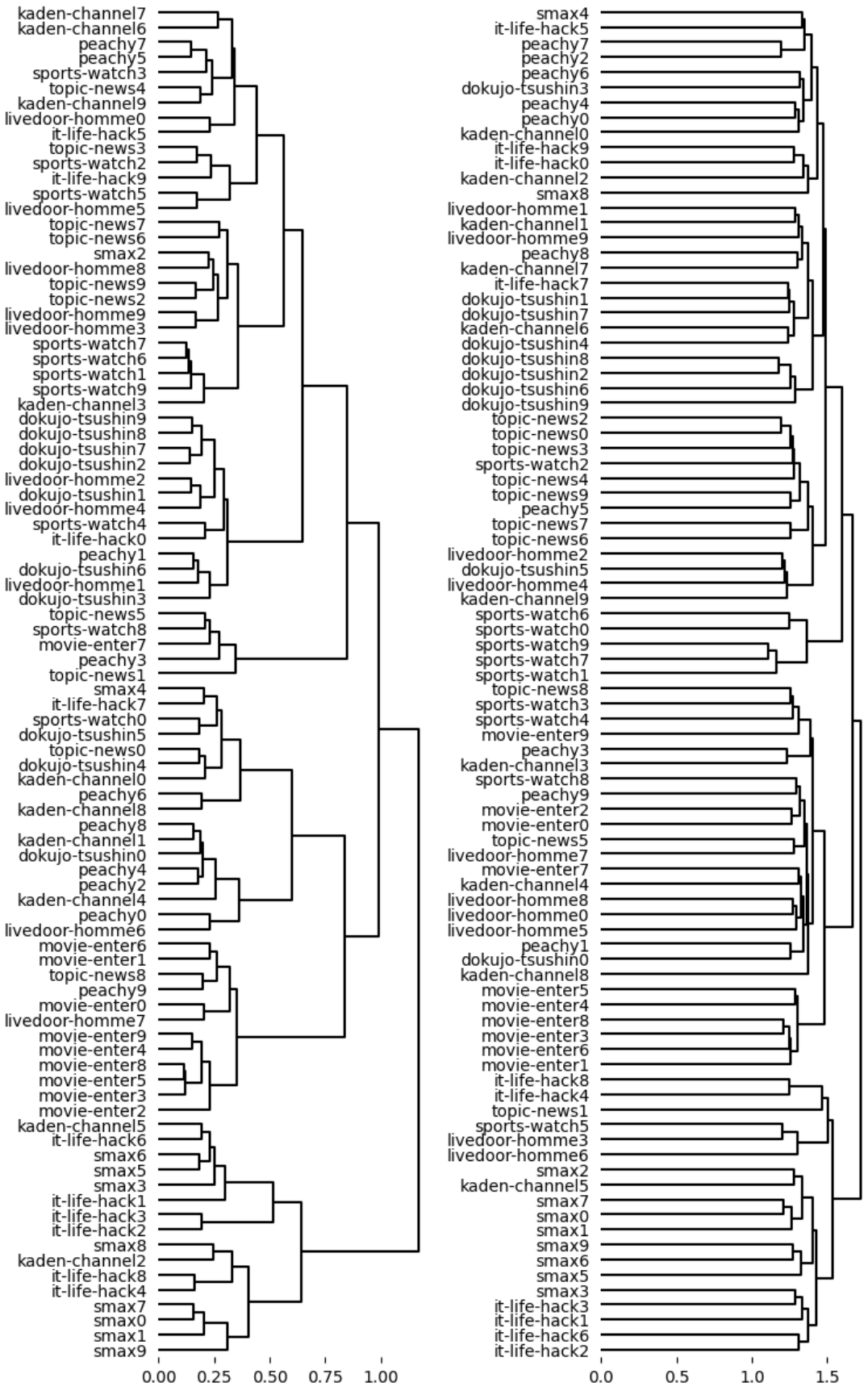
Figure 5.9: Compare accuracies by cluster analysis and latent semantic analysis

5.2.2 クラスタ分析におけるジャンル数

本実験では Figure 5.7 のように、7 ジャンルと 9 ジャンルにおいてほぼ同様かつクラスタ数と比例した正解率を持つことがわかった。しかし、5 ジャンルで行ったときはクラスタ数の増加に伴い正解率は右肩下がりになることがうかがえる。この実験で使用したジャンルは 4.2.2 で上から順に用いている。5 ジャンルでの次元削減は movie-enter まで、7 ジャンルでは smax までである。この結果からは 5 ジャンルと 7 ジャンルで用いた文書の違いに原因があると考えられる。特に smax は Figure 5.5 で示したように it-life-hack と類似していると考えられる。そのためそれを用いた 7 ジャンルと 9 ジャンルでは少ないクラスタ数の時にその分類がうまくできず、正解率が低いと考える。

5.2.3 クラスタ分析における文書数

文書数を変化させて行った実験の結果である Figure 5.8 からは正解率に関して大きな変化は見られなかった。このことから文書数の増加に関係なく、クラスタ分析による次元削減では類似した文書は類似した形に次元削減されると考えられる。



(a) Cluster analysis

(b) Latent semantic analysis

Figure 5.10: Similarity difference among cluster analysis and latent semantic analysis

第6章　結論

本研究では東北大学の日本語 BERT モデルによる分散表現とクラスタ分析を組み合わせた次元削減を行い、用いるトークンや文書数、ジャンル数を変更して行うことによって本手法による次元削減の特徴を調べた。次元削減後の行列を分類したときの正解率、デンドログラム、他手法との比較によって推測されたクラスタ分析による次元削減の性質は以下の通りである。

- クラスタ数を増やすと特徴的なトークンがさらに細分化されることで文書の特徴をさらに深くとらえた次元削減が可能
- 文書内すべてのトークンを用いた場合と名詞、動詞を抽出して行った場合の次元削減に大きな差は発生しない
- 文書に登場した固有名詞、英字などの影響が大きい

また、次元削減において一般的な方法である Tf-Idf を用いた潜在的意味解析による次元削減と比較したときに類似した文書に違いがあった。潜在的意味解析は topic-news に対して dokujo-tsushin が類似したジャンルだとしているが、クラスタ分析を用いた次元削減では sports-watch が類似しているとされている。livedoor ニュースコーパスにより正解が与えられていた今回の実験では潜在的意味解析を用いた次元削減が正解率においてクラスタ分析を上回っていたが、有名人の動向や発言などといった観点で分類を行う場合があるのならクラスタ分析を用いることが適する場合があると考えられる。

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) シグモイド関数を理解してみる
https://yaju3d.hatenablog.jp/entry/2018/10/31/013702
2) BERT とは — Google が誇る自然言語処理モデルの仕組み、特徴を解説,Ledge.ai
https://ledge.ai/bert/
3) 近江崇宏・金田健太郎・森長誠・江間見亜利　著,BERT による自然言語処理入門, オーム社,2021
4) cl-tohoku/bert-japanese
https://github.com/cl-tohoku/bert-japanese
5) livedoor ニュースコーパス, ロンウィット
https://www.rondhuit.com/download.html
6) BERT を使った文章分類
https://qiita.com/tetz1/items/17ae12627c6c2d4c0231
7) 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/
8) 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/
9) 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/intro2020.html