卒 業 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　　出口　利憲　教授

岐 阜 工 業 高 等 専 門 学 校　電気情報工学科

2017E17 後藤　貴樹

令 和 0 4 年　(2 0 2 2 年)　　2 月 1 7 日

Abstract

The data mining is being used in modern society. The text mining is one of them and extracts what is required of documents. The purpose of this study is to confirm the effectiveness of dimensionality reduce by cluster analysis, by means of document classification, which is one of the data mining methods.The method is to convert the document to a vector using BERT. This is dimensionally reduced by three methods using cluster analysis, vector mean, and CLS tokens.The effectiveness of cluster analysis can be confirmed by classifying documents with them.

Python is used in this experiment. Clustering performed by cluster analysis uses Ward's method, and the characteristics of document classification are obtained by changing the number of clusters, the number of documents used, and the number of genres of the documents.

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情報端末が老若男女へ普及し、人々の多くがインターネットを用いている。これに伴いインターネット上に存在するデータは数を増し、これからも増えていくと考えられている。これらデータのジャンルは様々であり、ある人にとっては無益だが、ある人にとっては有益なものも多い。このように近年増加の一途を辿るインターネット上のデータの中から自身にとって有益なデータを抽出するには相応の労力が必要であり、データの母数が多いほどその難易度は高くなる。

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自然言語によって情報化されたもの、つまり文書データに絞ったものをテキストマイニングという。このテキストマイニングだが、対象が自然言語となるため単語や接続詞などの意味を捉える必要があるため他の数値を対象とするようなデータマイニングより難易度が高く、今現在でも多義語に対する人とコンピュータの解釈の不一致や語順を考慮したテキストマイニングなどの多くの課題が残っている。

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を対象としている。その中でもクラスタ分析という方法を実装し、それを用いて文書の次元削減を行い、それを用いて文章分類を行う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とを目的としている。

この研究ではBERTという自然言語処理モデルを使用することによって文書を複数のベクトルへと変換し、これに対してクラスタ分析を行うことでベクトルの次元削減を行い、それを用いて文書分類を実行する。今回文書分類の対象となる文書データは9ジャンル存在し、livedoorニュースコーパス[1)]から取得する。また、取得した文書のジャンル数の変化とクラスタ分析時に行う階層的クラスタリングのクラスタ数の変化による分類精度を確認することによってクラスタ分析の特性を求める。

今回はクラスタ分析の比較対象として、以下の2項目の次元削減による文章分類を行う。

- ベクトルの平均値を用いた次元削減
- BERTのCLSトークンを用いた次元削減

これら二種類の次元削減と本研究の主題であるクラスタ分析による次元削減を比較する

ことで、BERT を用いた文書の分散表現を利用した文章分類におけるクラスタ分析の有効性を検討する。

第2章　基礎知識

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、大量のデータを統計学や人工知能を駆使することによって情報の傾向を見出す技術である。つまり、これを用いることによって膨大な情報から自身にとって有益な情報、知識を抽出することができるということである。対象となる情報が少数ならば私たち人間が情報をチェックすることでこれを行うことができる。しかし、対象となるデータが膨大であると作業量も増し、人の手で行うには厳しいものとなる。一方、コンピュータを用いた場合は処理が早く、人的なミスも存在しない。このような理由で現在データマイニングは使用されている。

2.1.2　テキストマイニング

テキストマイニングとはデータマイニングの一種であり、対象が文書データに限られている。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るのに使用されている。対象が自然言語であり、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度が高いとされている。

2.1.3　形態素解析

形態素とは、いわゆる単語や接続詞といった意味を持つ最小単位のことである。形態素解析は文書データから文法や品詞、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。これを用いることによって単語や品詞の抽出が可能になる。

2.1.4　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を用いて、それ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。例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、「赤いリンゴの描かれたコップ」というものに対しても「赤い」「リンゴの描かれたコップ」なのか「赤いリンゴの描かれた」「コップ」なのか判別がつきにくいといったように、自然言語の使い方には厳格なルールがないためこのような問題が発生する。

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技術である。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問題を解決する手法のことを指す。この過程として学習と推論の二つが存在する。

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で学習を行う手法である。

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データを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機械学習の技術の一つである。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、現代の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[4)] 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。

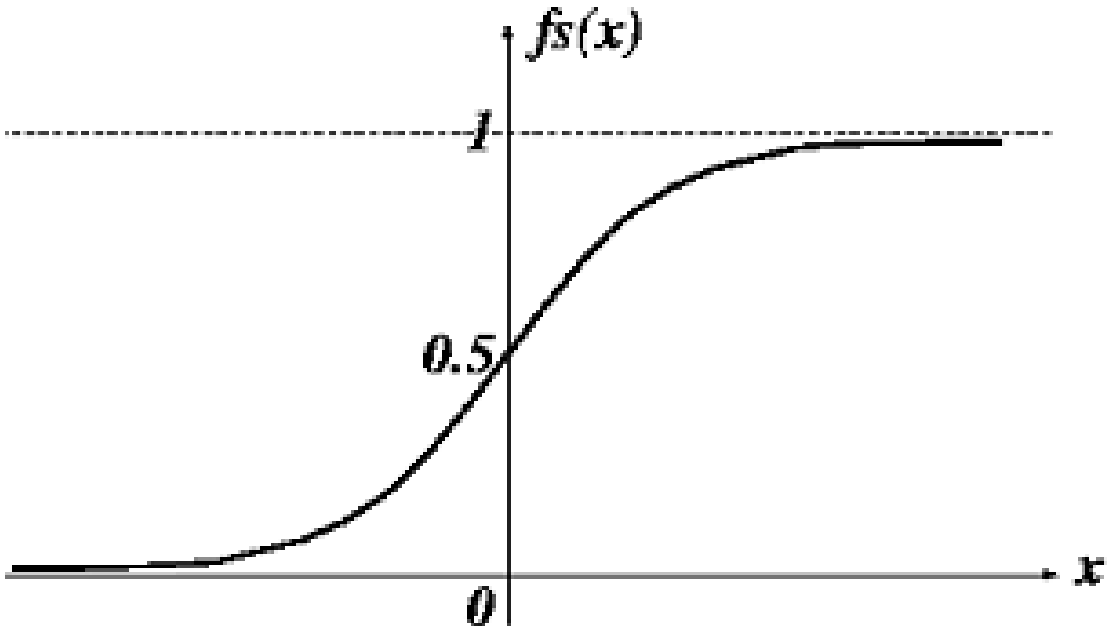

Figure 2.1　Sigmoid function.

2.3.7　言語モデル

言語モデルとは、言語を文章の出現しやすさの確率を用いてモデル化したものである。例えば、「私はパンを食べた」と「私は釘を食べた」の二つの文章に対して確率を与えるとする。この場合、前者は一般的にもよく使用されるが、後者は誤飲事故などでしか使用される事のない文であるため、前者の方に後者よりも高い確率を与える。これを行うことにより文章の自然さを確率を用いて表現することができ、文章誤りの訂正などに用いることができる。

2.3.8　分散表現

分散表現とは単語を高次元のベクトルとして表す技術である。コンピュータでは自然言語をそのままデータとして与えても処理できないため、コンピュータが処理できる数値として与えることによってコンピュータの疑似的な自然言語の理解を助ける。これに関連するものとして Word2vec が挙げられる。

2.3.9　Word2vec

Word2vec とは 2013 年に発表された自然言語処理の手法である。単語をベクトルで表現することによって単語の意味を数学的に表現することを目的にしている。同じ意味を持つ単語や同じタイミングで出現する単語を類似したベクトルで表現する。また、ベク

トルを加算、減算することで単語に意味を足したり引いたりすることを可能としている。

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、[5] 再帰型ニューラルネットワークがベースになっている（Figure 3.1）。このモデルの特徴は注意機構を有していることである。これは一つのトークンの出力を処理するに際、入力トークン全てを参照し、それらに重みをつけて出力の値を決定する。これによって文章を入力に与えた場合は前後の単語だけではなく文脈を考慮した出力が可能である。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理解した分散表現が可能になった。

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 [2] 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」

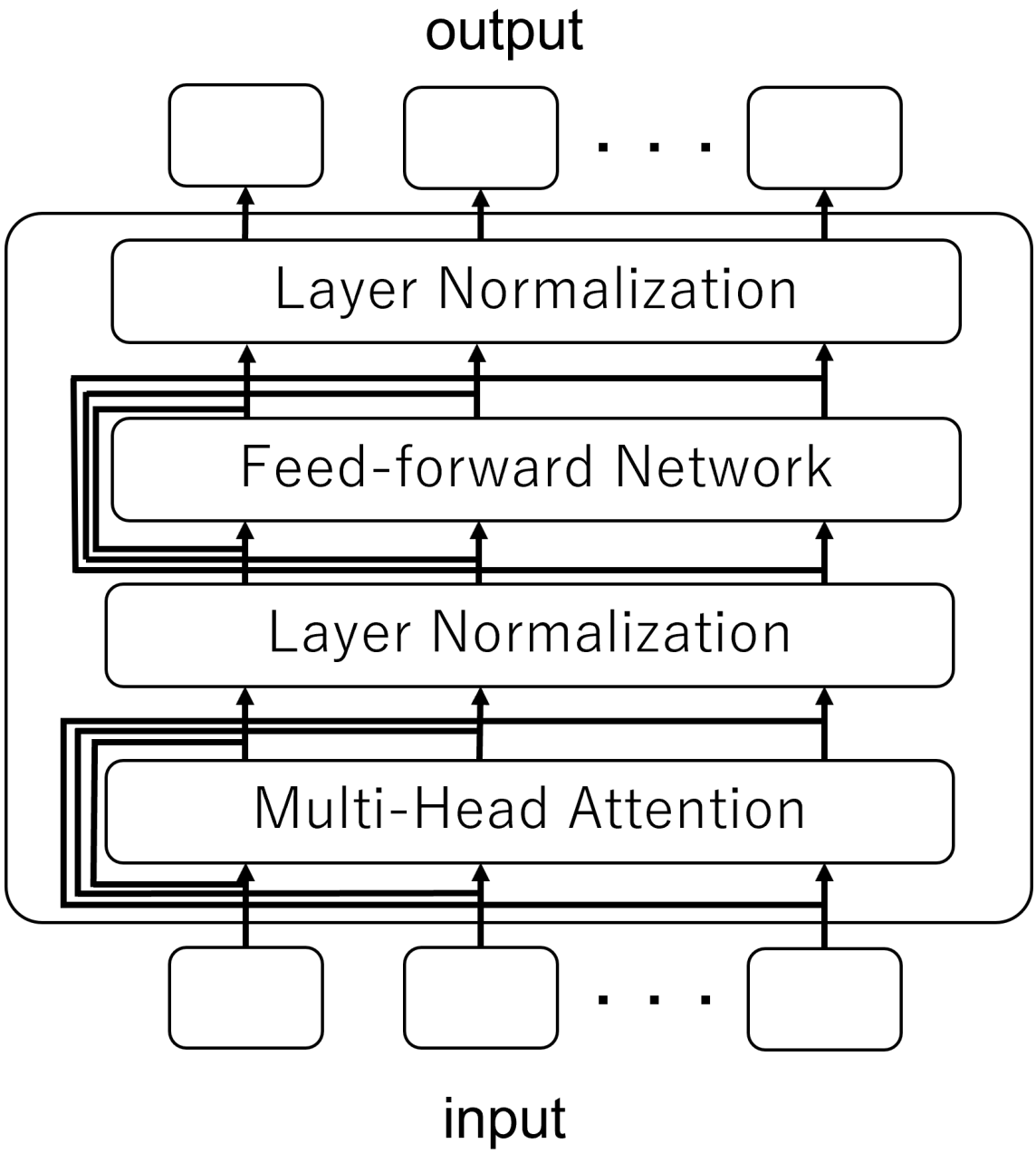

Figure 3.1　Structure of BERT.

「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。これに対する BERT の出力はトークンの分散表現ではなく文章その

ものの分散表現としての使用が可能である。

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 Python

3.2.1 Python

Python とはコンピュータ言語の一つであり、グイド・ヴァン・ロッサム氏によって開発された。この特徴として以下の点が挙げられる。

- インタプリタ形式
- オブジェクト指向
- 移植が容易
- オープンソースで運営されている
- シンプルな記述で可読性が高い
- ライブラリが豊富

このような特徴から現在根強い人気を博しており、何百万ものユーザーがいるといわれている。今回は主に以下の 3 つのライブラリをインポートして使用した。

- Transformers
- Matplotlib
- Numpy
- SciPy

3.2.2 Transformers

Transformers とは Huggingface 社が提供するオープンソースのライブラリである。BERT を含むニューラルネットワークを用いたさまざまなモデルを提供しており、さまざまな言語に対する事前学習モデルを保有している。今回はその中の一つである東北大学研究チームにより作られた BERT の日本語モデルを使用する。

3.2.3 Matplotlib

Matplotlib とは Python のグラフ描画のためのライブラリである。折れ線グラフやヒストグラムなど様々な種類のグラフの描画を可能とするオブジェクト指向の API を提供している。今回はこれを用いて結果を折れ線グラフで表示した。

3.2.4 Numpy

Numpy とは Numeric を起源に持つ Python の拡張モジュールであり、数値計算の効率化を目的といている。Python は動的型付け言語であるため数値計算が C 言語や Java といった静的型付け言語よりも計算時間が長いという欠点を持つが、Numpy では ndarray という独自の多次元配列オブジェクトとそれに対する関数を提供することにより、計算を高速化する。

3.2.5 SciPy

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

3.3 Google Colaboratory

Google Colaboratory

Google Colaboratory とは Google が提供するサービスであり、ブラウザから Python を実行できる。環境構築が不要であり、多くのライブラリがインストール済みである。また、GPU を使用することができる。今回はこれを利用して実験を行う。

3.3.1 GPU

GPU とは Graphics Processing Unit の略称である。画像処理に特化した演算装置であるが今回は画像処理で用いるわけではない。GPU は大量の演算を並列かつ高速に処理することができる性能を持っているため、今回の実験ではベクトルの演算を行う際にこの GPU を使用した。

3.4　階層的クラスタ分析

階層的クラスタ分析とはクラスタリングの手法の一つである。Figure 3.2 のように与えられたデータからデータ間の距離などの情報を用いて類似したデータをクラスタとしてまとめる。これを繰り返すことによってデータをクラスタリングすることができる。特徴として Figure 3.3 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。デンドログラムではデータ間の距離を縦方向で表しており、最下方で連結した 2 つのデータの距離が最も近い。

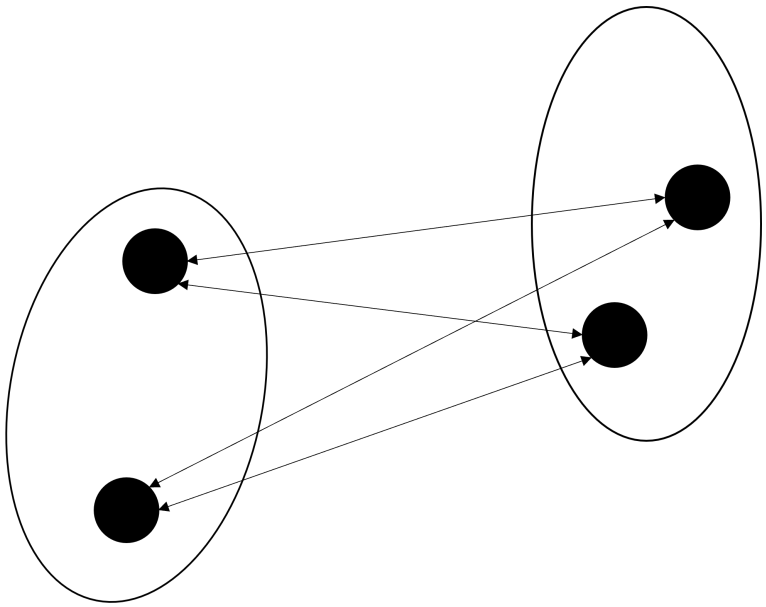

Figure 3.2 Cluster analysis.

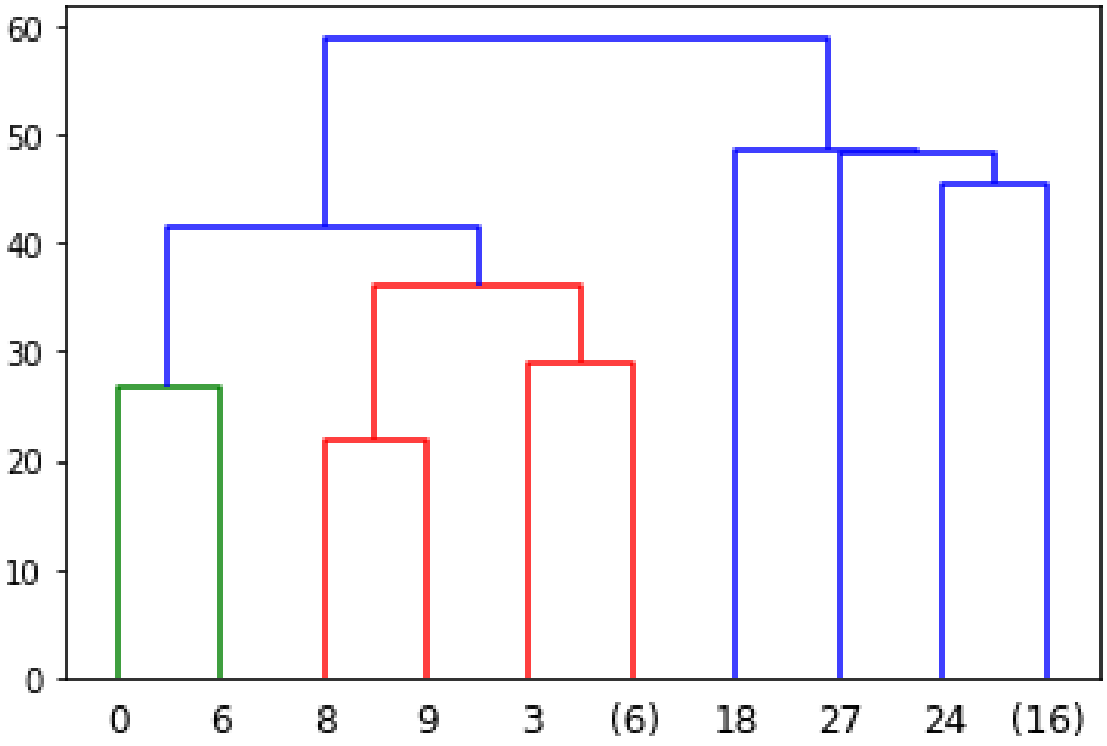

Figure 3.3 Dendrogram.

第4章　実験

4.1　実験の概要

本実験ではlivedoorニュースコーパスを用いてダウンロードした9ジャンルのニュース記事を文章分類の対象とする。文章分類は次元削減が行われた文章ベクトルをクラスタリングすることで行う。文章ベクトルは文章内の全てのトークンの平均ベクトル、BERTのCLSトークンのベクトル、そしてクラスタ分析で次元削減された文書クラスタ行列を用いる。また、クラスタ分析においては単語のクラスタリング時のクラスタの個数を10から300まで10刻みで変化させ、それによる変化を比較する。これら分類の結果の正答率を比較することによってクラスタ分析による次元削減がBERT従来の方法である平均ベクトルやCLSトークンと比べてどのような差別化ができるかを確認することがこの実験の目的である。

また、この実験では以下の8種類の文章分類を行う。

- ジャンル数3：1ジャンルにつき10文書
- ジャンル数5：1ジャンルにつき10文書
- ジャンル数7：1ジャンルにつき10文書
- ジャンル数9：1ジャンルにつき10文書
- ジャンル数3：1ジャンルにつき20文書
- ジャンル数5：1ジャンルにつき20文書
- ジャンル数7：1ジャンルにつき20文書
- ジャンル数9：1ジャンルにつき20文書

ジャンル数と文書数を変化させて行うことによってクラスタ分析におけるジャンル数と文書数の関係を調べる。

4.2　実験準備

4.2.1　実験環境の構築

本実験で使用したプログラムはGoogleColaboratoryで実行させた。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。今回はこれの本文の最初から 128 トークン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

今回の実験の目的はクラスタ分析による次元削減の有効性を求めることであるため、対象となる文書データにジャンルが存在すると正解率を求めやすくなるとともにクラスタリングが容易になると考え、これらの文書データを用いる。これらジャンルからそれぞれ 10 個ずつもしくは 20 個の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 [1)] から入手した 9 ジャンルの文書を文章分類の対象とするが、これら文章にはそれぞれタイトル、日時などの情報が含まれているため、それら情報を抜いて使用した。今回は後述する BERT モデルのトークナイザを用いて 1 文書につき 128 トークンまで分割し、それら先頭に CLS トークン、末尾に SEP トークンがついた計 130 トークンを用いて文章分類を行った。

```
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.

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 [2] 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. 全ての文書をトークン化する
2. 全てのトークンをベクトル化する

3. 2. でできた全てのベクトルを linkage と fcluster 関数で階層的なクラスタリングする
4. Table 4.1 のように単語がそのクラスタに属するかを 0 と 1 で表すトークンクラスタ行列を作成する
5. Table 4.2 のように単語がその分に入っているか否かを 0 と 1 で表す文書トークン行列を作成する
6. 4. の行列と 5. の行列の積をとり、クラスタ文書行列を作成する
7. 6. の行列を転置した文書クラスタ行列を作成する
8. 7. の行列をクラスタリングする

次元削減前は 1 文書の分散表現は 128 個 768 次元のベクトルにより表されたが、この処理を行うことによって 1 文書あたり設定したクラスタ数個にまで数値が削減される。

Table 4.1　Word-cluster matrix.

	word1	word2	word3	word4	word5	word6
cluster 1	0	1	0	0	1	0
cluster 2	0	0	1	0	0	0
cluster 3	1	0	0	0	0	1
cluster 4	0	0	0	1	0	0

Table 4.2　Document-word matrix.

	document 1	document 2	document 3	document 4	document 5	document 6
word 1	0	1	0	0	0	0
word 2	0	0	1	0	0	0
word 3	0	0	0	0	0	1
word 4	0	0	0	1	0	0

4.3.2　ベクトルの平均値による次元削減

BERT モデルにより livedoor ニュースコーパスから取得した文書ごとに 128 トークンに分割し、それに付加される CLS トークンのベクトルを除いたものを使用する。これら

ベクトルの平均値を計算し、その 1 つの 768 次元のベクトルをその文書における代表ベクトルとして次元削減を行った。この方法を用いることにより、1 文書につき 128 個で表されていたベクトルの数を 1 つのベクトルで表現できる。

4.3.3 CLS トークンによる次元削減

取得した BERT モデルに取得した文書データを与え、128 トークンに分割する。この際先頭に付加する CLS トークンのみを次元削減に用いる。これの分散表現を文書の代表ベクトルとして用い、次元削減を行った。ベクトルの平均値を用いた方法と同様に、1 文書につき 1 つのベクトルで表される。

4.4 正解率の計算

4.4.1 クラスタの割り当て

ベクトルの平均値、CLS トークンベクトル、クラスタ分析で作成した文書クラスタ行列をそれぞれクラスタ数が実験で用いたジャンル数と等しくなるようクラスタリングする。これにより割り振られたクラスタ番号をジャンルの番号としてみなすことで、Table 4.3 のように 1 ジャンルにつき与えられた 10 もしくは 20 個の文書にクラスタ番号が与えられる。これらからジャンルごとの最頻値をそのジャンルにおけるジャンル番号として扱う。

4.4.2 重複の削除

クラスタリングによりジャンルが違う文書が同じクラスタ番号を割り振られた場合、以下の処理を行う。

1. 重複したクラスタ番号の数を数える
2. 1. の数が小さいほうの次に多いクラスタ番号をそのジャンルの番号とする
3. 1. 3. を重複がなくなるまで繰り返す

これによりジャンルに対応するジャンル番号から重複をなくすことでジャンルとクラスタ番号に 1 対 1 の関係を与える。

Table 4.3　Allocation of number.

	Allocated cluster number									
Genre0	5	7	7	7	7	7	7	7	7	7
Genre1	3	5	3	6	3	8	5	3	1	5
Genre2	2	3	8	5	1	3	3	5	6	3
Genre3	5	2	2	7	1	6	5	6	0	6
Genre4	0	1	1	1	1	1	5	2	3	5
Genre5	1	5	5	5	5	2	2	6	5	5
Genre6	8	6	4	4	3	8	8	4	2	8
Genre7	2	2	2	2	2	2	2	2	2	2
Genre8	1	7	2	1	2	2	2	5	5	2

4.4.3　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、4.4.2で重複をなくした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した文書数として使用して、全文書数で除算することで式 4.1 のように正解率を算出した。

$$accuracy = \frac{\text{正解した文書数}}{\text{全文書数}} \tag{4.1}$$

4.5　実験結果

4.5.1　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とベクトルの平均値、CLS トークンを用いた次元削減による文章分類を行い、それらの正解率を比較することでクラスタ分析によるアプローチの有効性を評価する。これを対象となる文書のジャンル数を 3 から 9 まで 2 刻みで行い、クラスタ分析の場合はクラスタ数を 10 から 300 まで 10 刻みで変化させ、使用する文書も各 10 個と各 20 個の二種類で行う。これによりクラスタ分析による次元削減の特性と平均値ベクトルと CLS トークンを用いた次元削減を確認し、それらと比較する。

4.5.2　クラスタ数による正解率の推移

対象となった文書のジャンル数を 9、文書数を 1 ジャンル 10 個の計 90 個の文書を用いてクラスタ分析による次元削減を行った。Table 4.2 にその正答率の推移を示す。Figure 4.2 からクラスタ数が 50 に到るまでは著しく文書分類の精度が低く、クラスタ数が 150 を超えたあたりから正解率が 0.40 を超えることが多くなった。しかし、クラスタ数の変化による正解率の変化は微々たるものである。

4.5.3　ジャンル数による正解率の推移

4.3 にジャンル数が 7 の時の正解率の推移を、Figure 4.4 にジャンル数が 5 の時の推移を、Figure 4.5 にジャンル数が 3 の時の推移をそれぞれ示す。Figure 4.2 と Figure 4.3 を比較するとジャンル数が 2 減ったことによって全体を通して正解率が大きく上がったことがわかる。同様に Figure 4.2 と Figure 4.4、Figure 4.5 を比較しても、ジャンル数が減少するほど正答率が上昇している。その一方、Figure 4.4 と Figure 4.5 ではクラスタ数が多くなるにつれ正解率に減少傾向があることがわかる。

4.5.4　文書数による正解率の推移

ジャンル数 9 でそれぞれのジャンルから 10 文書ずつ使用して文章分類を行なったものを Figure 4.2 に 20 文書ずつ使用したものを Figure 4.6 に示す。文書数を各ジャンル 10 個で文章分類した Figure 4.2 と Figure 4.6 を比較したところ、全体の正解率に大きな影響は出ていないが、文書数が各ジャンルにつき 20 個にした場合のほうはクラスタ数を増加させたときの正解率が上昇傾向にあり、180 を超えた付近から正解率が多くの場合 0.45 より高くなっている。

同様の実験でジャンル数を 7 にした場合の結果を Figure 4.7、5 にした場合を Figure 4.8、3 にした場合の結果を Figure 4.9 に示す。ジャンル数が 7 の場合での各ジャンルの文書数が 10 のものと 20 のもので比較してみると、両者共にクラスタ数が 170 の時に大きく正解率が落ちているが、文書数を 20 にしたときはクラスタ数が 150 付近から正解率がピークを迎えており多くの場合に 0.45 を超える正解率を出している。一方ジャンル数が 5 の場合はクラスタ数が 70 付近から正解率が 0.50 を超え続けている。ジャンル数が 3 の場合は 9 ジャンルの時と同様にクラスタ数が 170 付近で正答率が上昇し、最も高いもので 0.80 に届くほどであった。それに加え、文書数が各ジャンル 10 の時と比較し、クラスタ

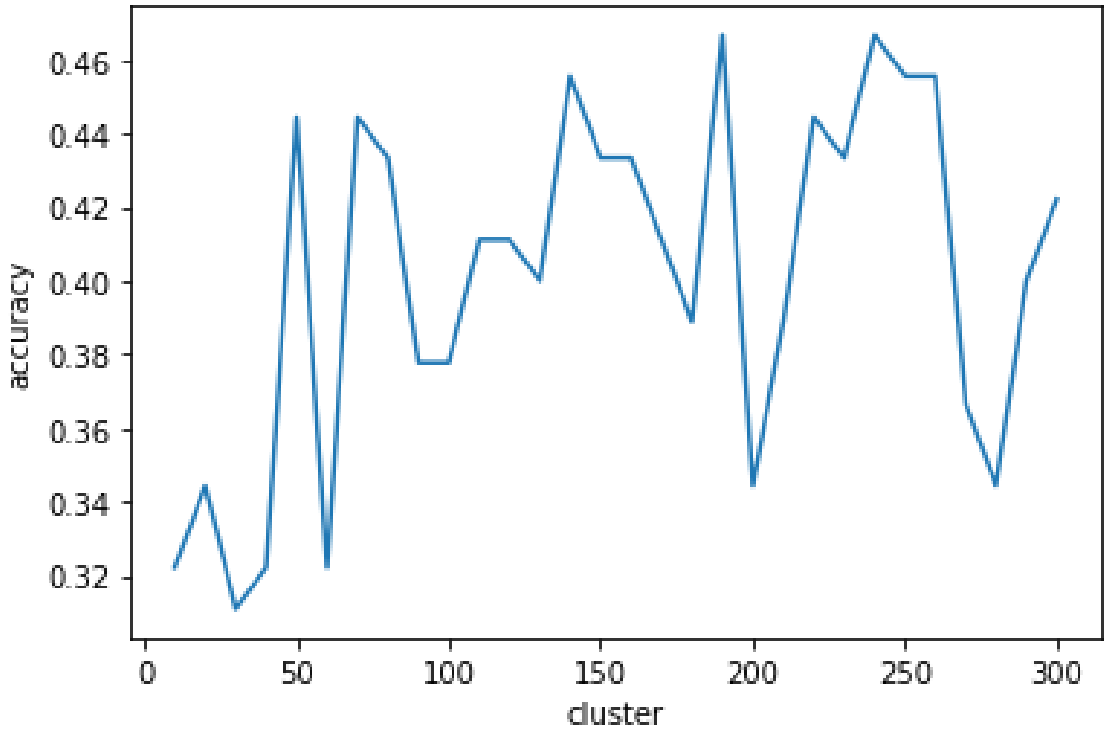

Figure 4.2 Accuracy of 9 genres.

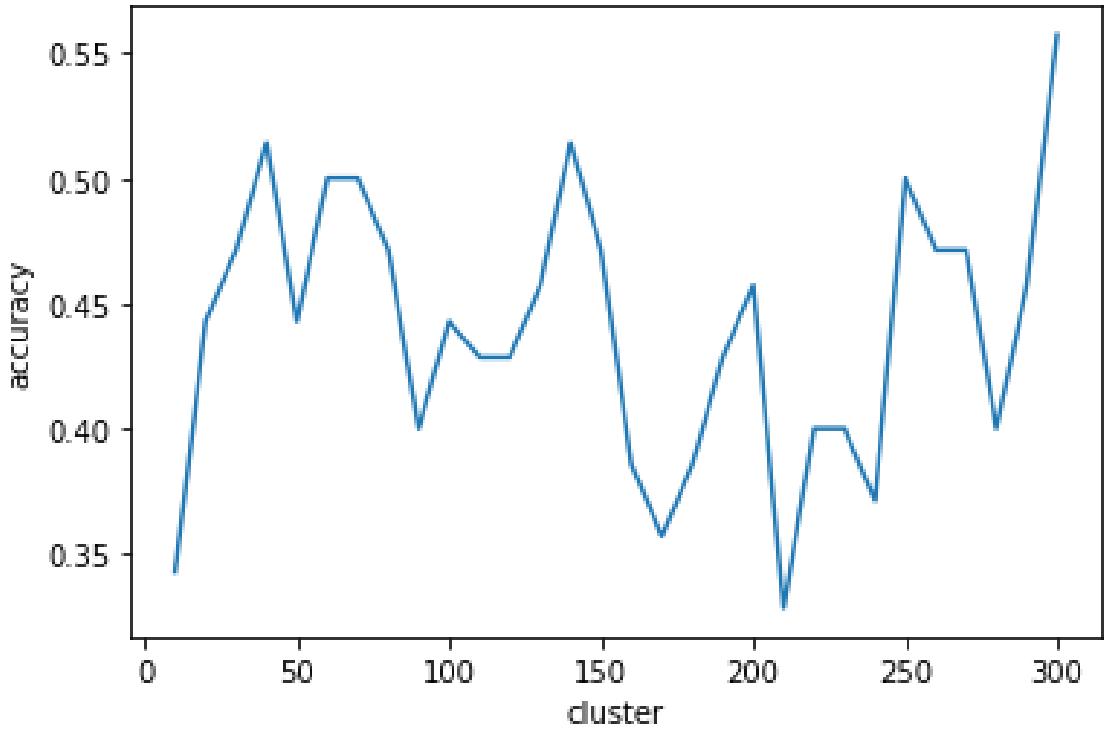

Figure 4.3 Accuracy of 7 genres.

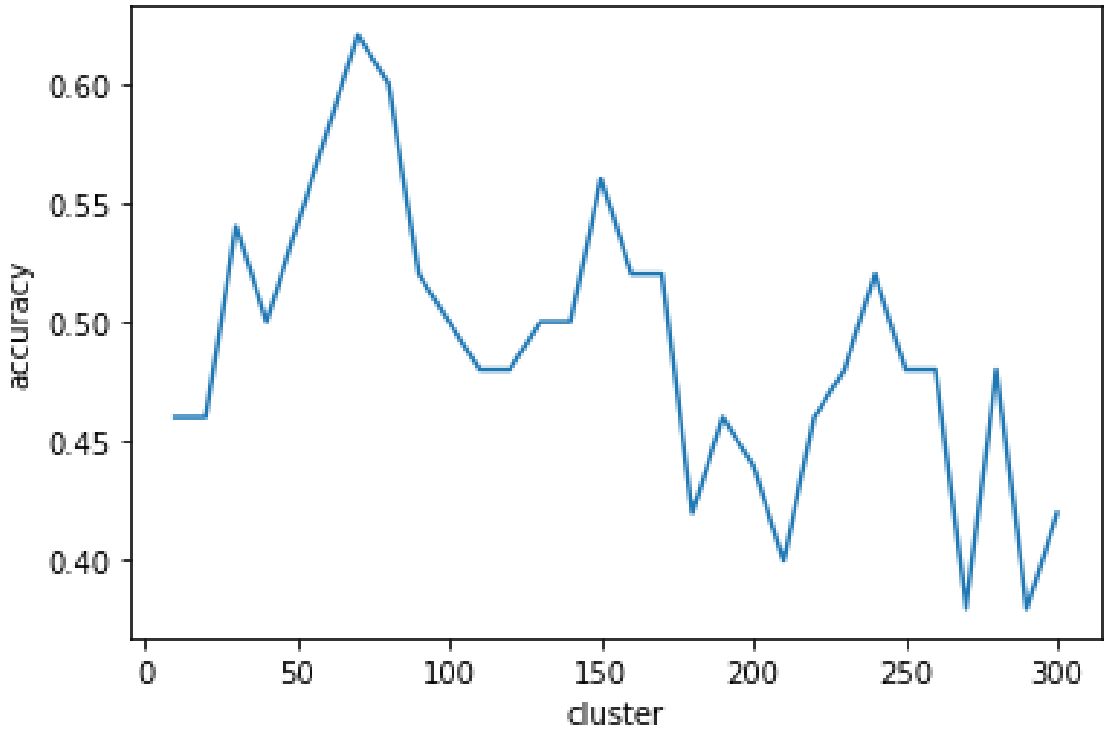

Figure 4.4 Accuracy of 5 genres.

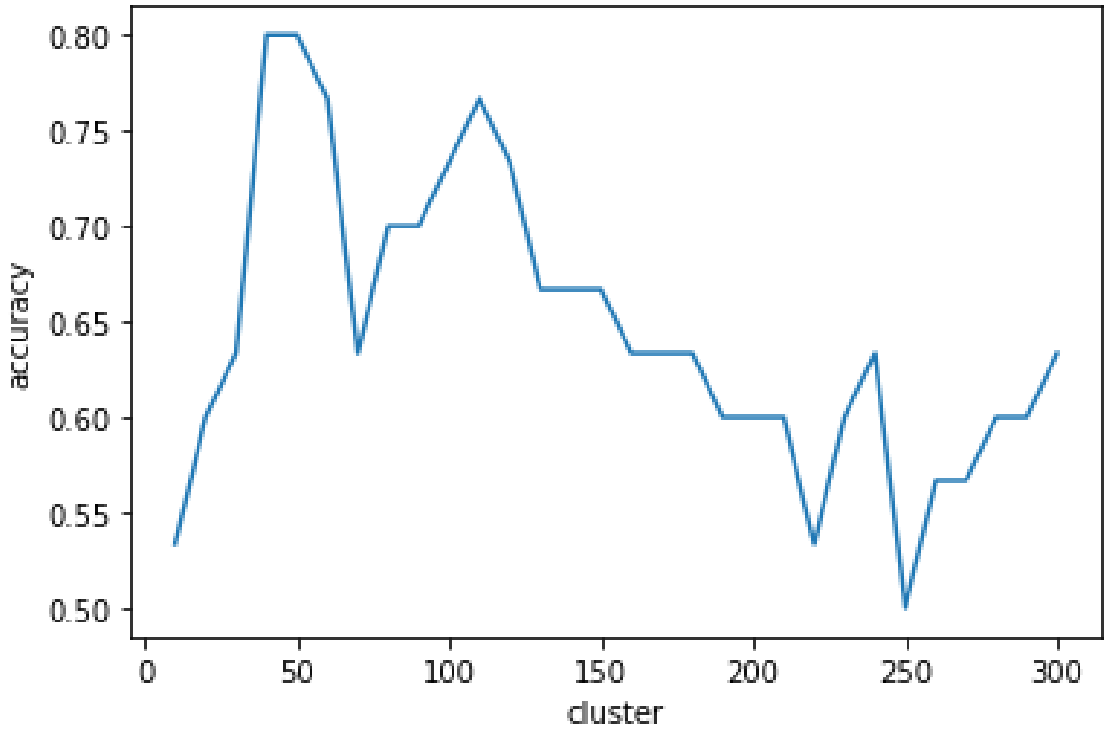

Figure 4.5 Accuracy of 3 genres.

数が増えたときに大きく正解率が下がっていた 5 ジャンルと 3 ジャンルの場合の正解率が各ジャンル 20 文書使用している場合では正解率が下がり始めることなく、ほぼ一定であることがわかる。

4.6 考察

4.6.1 クラスタ分析におけるクラスタ数

今回の実験で得られた結果ではクラスタ分析においてクラスタ数が 10 や 20 などの場合にはあまり高い正解率が出ないことがわかった。これは動詞や品詞、名詞といった文章を構成する要素の配置が自然言語の特性上同じパターンであることが多く、ベクトル化した際もこれら要素は大きく似通うため、クラスタリングの数が少ない場合では動詞、品詞、名詞といったようにクラスタがわかれてしまい、多くの文書が似通った構成のクラスタ番号列になってしまうからと考えられる。実際に 9 ジャンル 10 文書ずつでクラスタ分析時の分割クラスタ数を変化させ、クラスタ数とジャンルごとに割り振られたクラスタ番号の関係を Table 4.4 に示す。

Table 4.4 Relation of Genres and Cluster(9 genres).

	10 cluster	50 cluster	100 cluster	200 cluster	300 cluster
genre 0	5	8	3	2	2
genre 1	5	7	7	1	4
genre 2	0	5	0	7	1
genre 3	0	5	0	0	5
genre 4	3	3	1	5	7
genre 5	2	5	3	5	2
genre 6	6	2	4	1	4
genre 7	0	3	6	6	0
genre 8	0	6	1	3	0

この表からわかる通り、クラスタの数が 10 や 50 と少ない場合において、クラスタリングによってジャンルに割り振られるクラスタ番号が複数ジャンルと重複することが多く、それぞれが別のジャンルとして分けられていないことがわかる。一方クラスタ数を

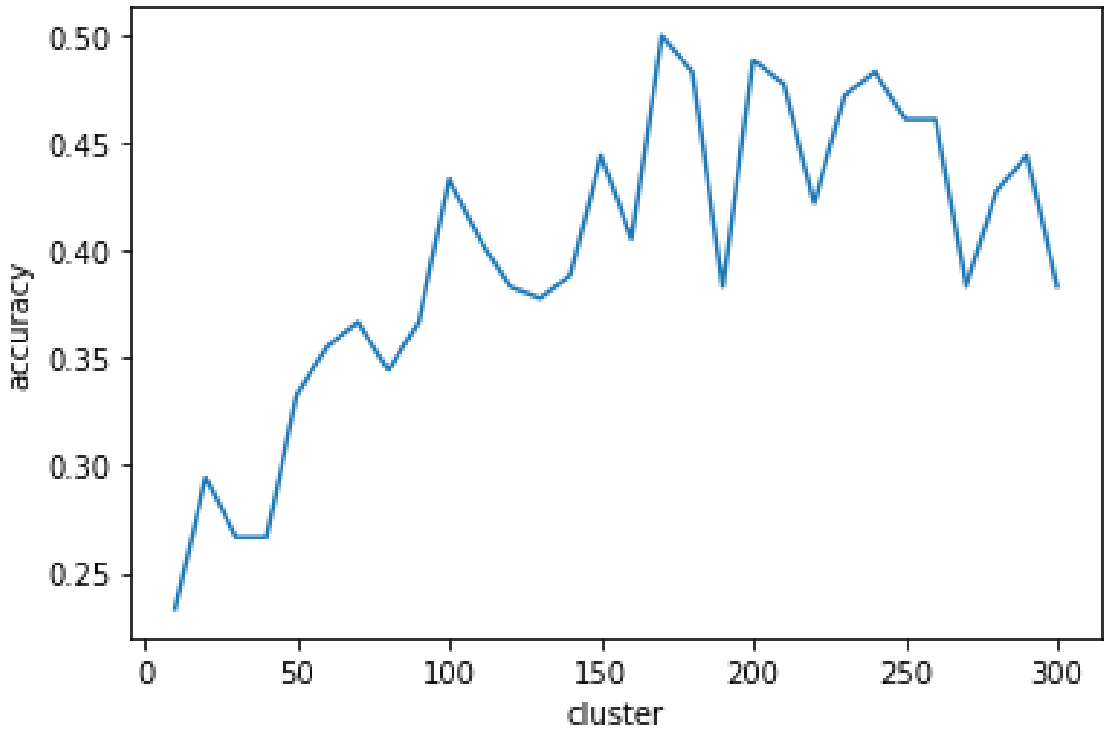

Figure 4.6 9 genres each 20 documents.

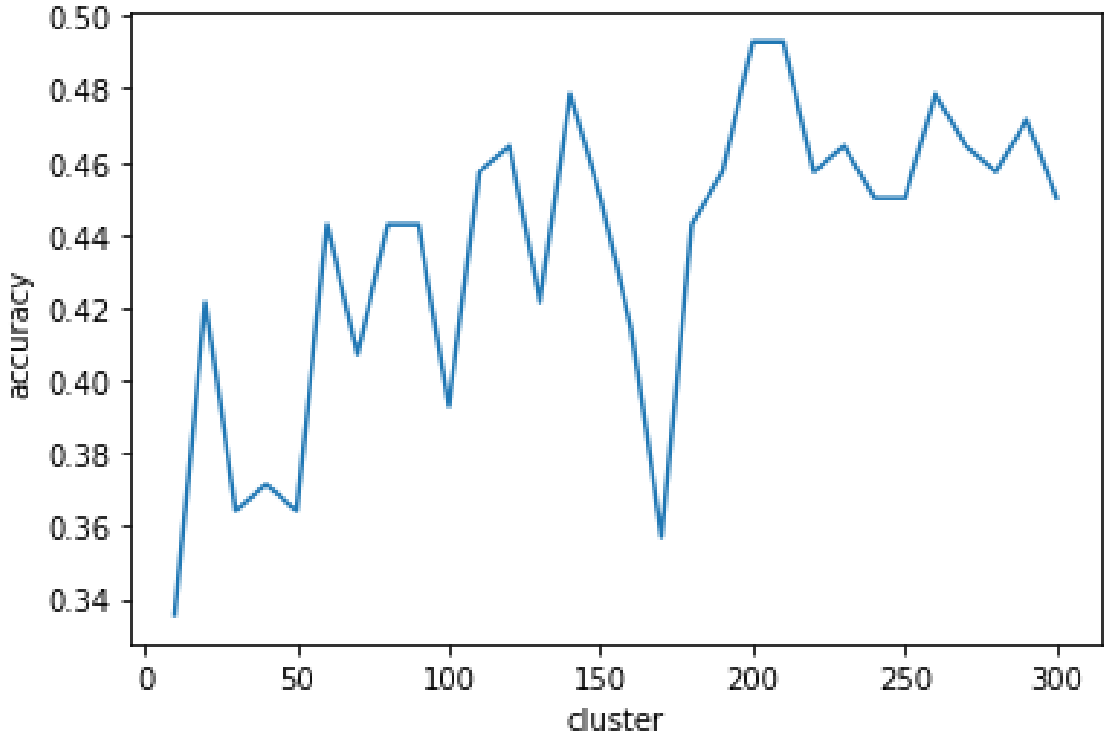

Figure 4.7 7 genres each 20 documents.

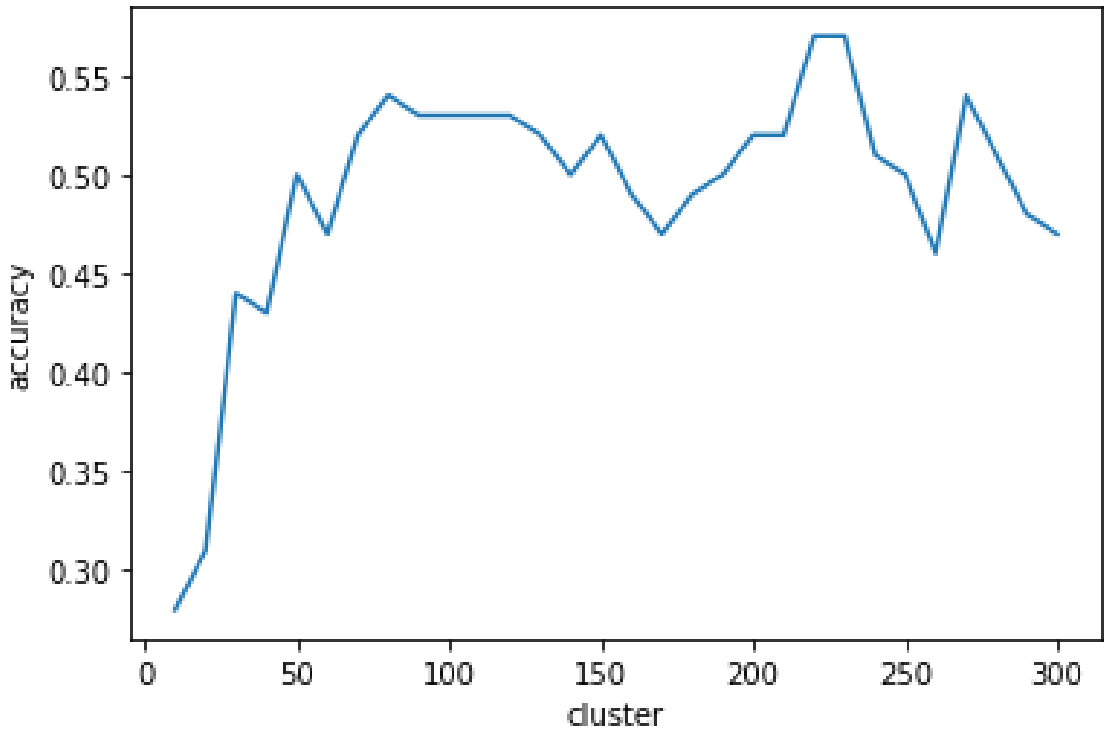

Figure 4.8 5 genres each 20 documents.

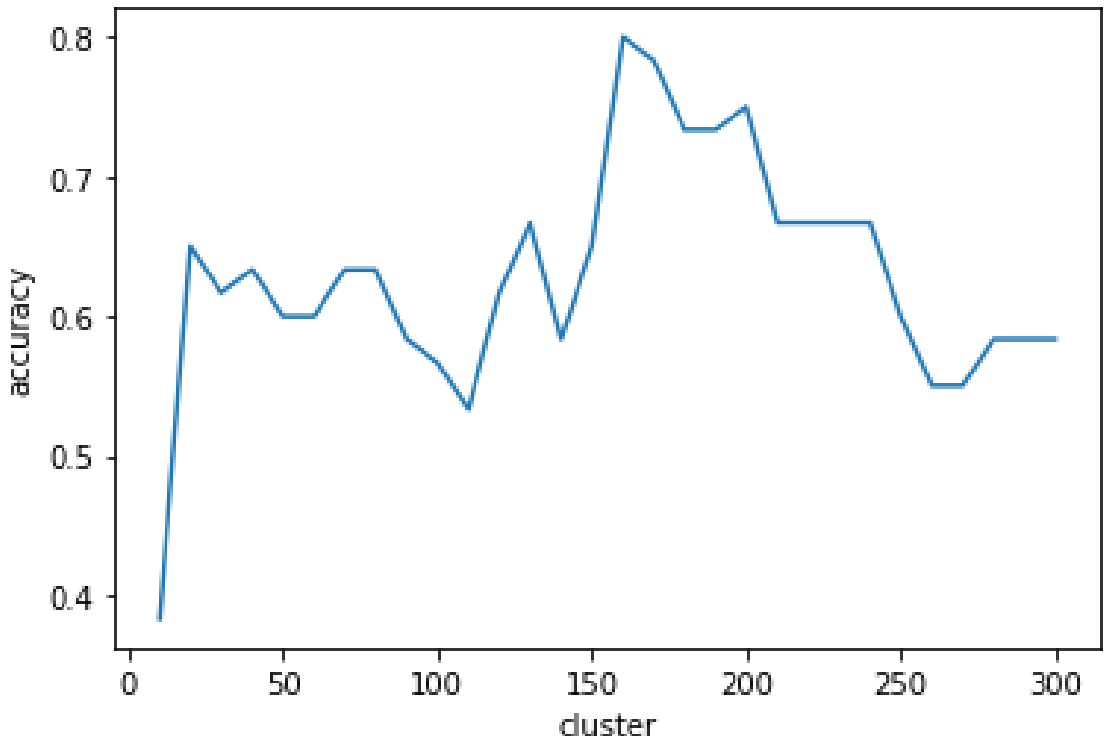

Figure 4.9 3 genres each 20 documents.

増加させ、200 や 300 などにした場合では、重複が無くなるわけではないがしたとしても 2 ジャンルほどであり、ジャンルごとに分かれやすくなっていることがわかる。

4.6.2　クラスタ分析におけるジャンル数

今回行った実験ではジャンル数を減らすほど正答率が高くなるという結果が得られた。この一因として、ジャンル数が減ったことによってクラスタリングされた文書が正解となる確率が増加したということがある。それとは別にジャンル数が減ったことによってデータ数が少なくなり、対象となるベクトルの特徴量が比較的見分けやすくなっているのではないかと考えられる。また、今回はジャンル数ごとの文書を同じ数にして実験したが、これにより 3 ジャンルの場合は 30 文書、5 ジャンルの場合は 50 文書といったようにジャンルの数によって総文書量が変化してしまったのも一つの原因かと考えられる。

また、4.6.1 で行ったもののジャンル数を 5 に変更したものの結果を Table 4.5 に示す。

.

Table 4.5　Relation of Genres and Cluster(5 genres).

	10 cluster	50 cluster	100 cluster	200 cluster	300 cluster
genre 0	0	0	1	3	0
genre 1	2	1	3	4	3
genre 2	3	4	4	1	3
genre 3	2	2	0	0	3
genre 4	1	3	2	0	0

9 ジャンルで行った結果である Table 4.4 と比較して、こちらではクラスタ数が少ないときのクラスタ番号の重複が少なく、一方クラスタ数が増えるにつれてジャンルに対するクラスタ番号が重複するようになった。この重複の理由としては、ジャンル数が少なく、なおかつ用いた文書が各ジャンルにつき 10 文書であることだと考えられる。ジャンルが減ることによって総文書量も減少してしまい、5 ジャンルの場合は 50 文書しか用いていないことになる。これは総文書が 30 文書しか用いていない Figure 4.5 では特に顕著であり、これはトークン数に対してクラスタ数が多くなりすぎたためだと考えられる。3 ジャンル 10 文書ずつの場合トークン数は $3 \times 10 \times 128$=3840 トークンとなるが、日本

語の特徴上そのうち多くのトークンが「が」、「に」、「を」、「は」などの助詞になると考えられる。そのため、区別しやすい名詞などのトークンの数は実際にはその半数近くになってしまい、クラスタ数が 250 や 300 になってしまうと多少似通った分散表現のものも別のクラスタに割り当てられてしまい正解率が落ちてしまうと考えられる。

4.6.3 クラスタ分析における文書数

今回の実験では使用する文書を各ジャンル 10 個ずつのものと 20 個ずつの 2 種類で文章分類を行った。特に 20 文書で行った Figure 4.8 と Figure 4.9 はクラスタ数が増えるごとに正解率が上がっていく傾向が顕著に出現している。その一方、10 文書ずつ用いた実験の特徴であったクラスタ数 250 あたりからのクラスタ数の増加による正解率の低下が見られなかった。この要因としては、文書数を 10 から 20 に増やしたことにより出現する単語が 2 倍ほどになったため、クラスタ数が 250 や 300 程になってもクラスタリングで似通ったベクトルが別のクラスタに割り当てられることがなくなったためだと考えられる。

4.6.4 他の次元削減との比較

今回の実験ではクラスタ分析による次元削減以外にもベクトルの平均値を用いた次元削減と CLS トークンを用いた次元削減も行った。それぞれ同じ条件で行った実験の正解率で 1 ジャンルの 10 のものを Table 4.6、20 のものを Table 4.7 に示す。

Table 4.6 Average and CLstoken Approach(10docs).

	9 genres	7genres	5genres	3genres
Vector Average	0.44	0.49	0.54	0.63
CLS Token	0.53	0.56	0.58	0.67

Table 4.7 Average and CLstoken Approach(20docs).

	9 genres	7genres	5genres	3genres
Vector Average	0.53	0.51	0.54	0.62
CLS Token	0.58	0.50	0.57	0.70

クラスタ分析で 9 ジャンル 10 文書ずつ用いた文章分類では正解率の最大値が 0.47 であった。これと Table 4.6 を比較するとベクトルの平均値より正解率が高いが、CLS トークンを用いた方法よりは低いことがわかる。同様に文書数を 20 にして行った場合でもクラスタ分析を用いた次元削減での正解率が CLS トークンを用いた次元削減よりも高い正解率を出すことはなかった。

一方、ジャンル数を減らし、5 ジャンルで文章分類を行った際は両条件において平均値を用いた方法より高い正解率を出すことに成功しており、CLS トークンを用いた方法と同等もしくはそれ以上の正解率である。これよりさらにジャンル数を減らした 3 ジャンルでは 10 文書の場合も 20 文書の場合も正解率が 0.80 に達しており、CLS トークンを用いた方法よりも高い正解率だった。

最も高い正解率を出すためには最適なクラスタ数を定める必要がある。このクラスタ数は今回の実験によりジャンル数と使用する文書数によって変動することがわかったので、Figure 4.10 に各ジャンル 10 文書の時のジャンル数と正解率が最高の時のクラスタ数の相関を示す。このグラフを見ると、ジャンルとクラスタ分析における各ジャンルで最も高い正解率を出したときのクラスタ数は比例関係を持っているように見える。

各ジャンルにつき 20 文書用いた場合の相関も Figure 4.11 に示す。Figure 4.10 とは違い、7 ジャンル目から最高正解率を出したときのクラスタ数が急激に下がり始める。

これら二つの違いとして挙げられるものは使用する文書数なのだが、文書数が多い場合かつ 7 ジャンル以上の分類を行う場合は同文書量で 5 ジャンルの場合の理想のクラスタ数より少ないクラスタ数が最も正解率が高くなるという結果が得られた。

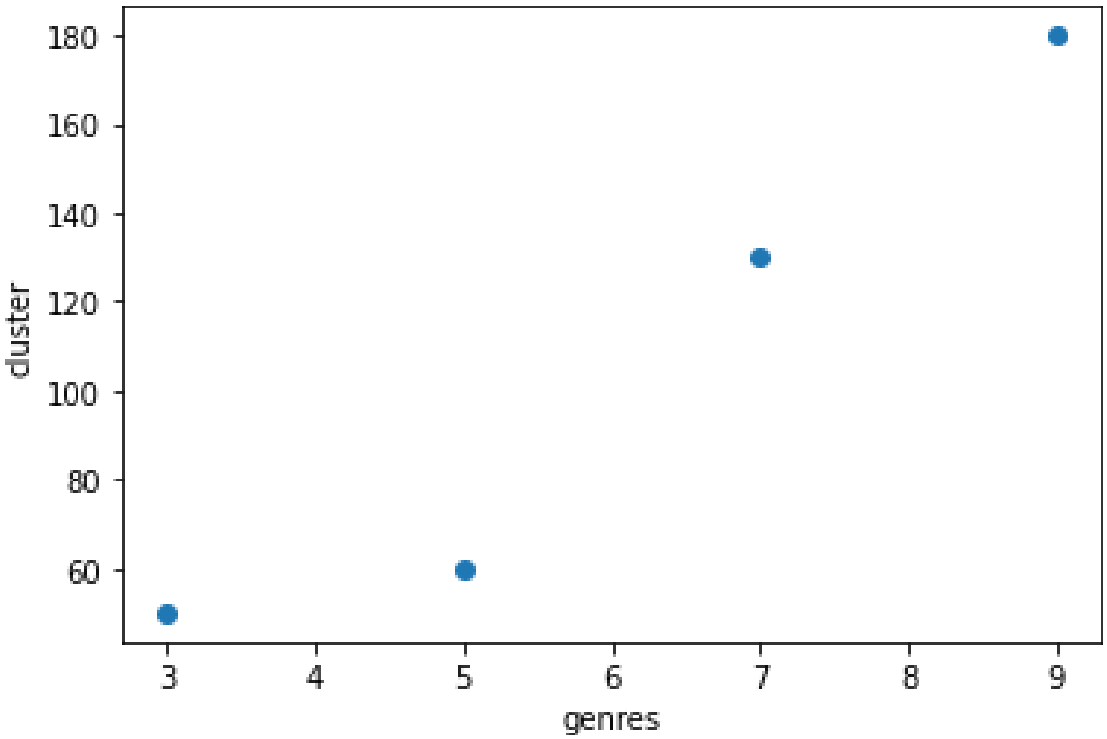

Figure 4.10 Relation of genres and cluster (10 docs).

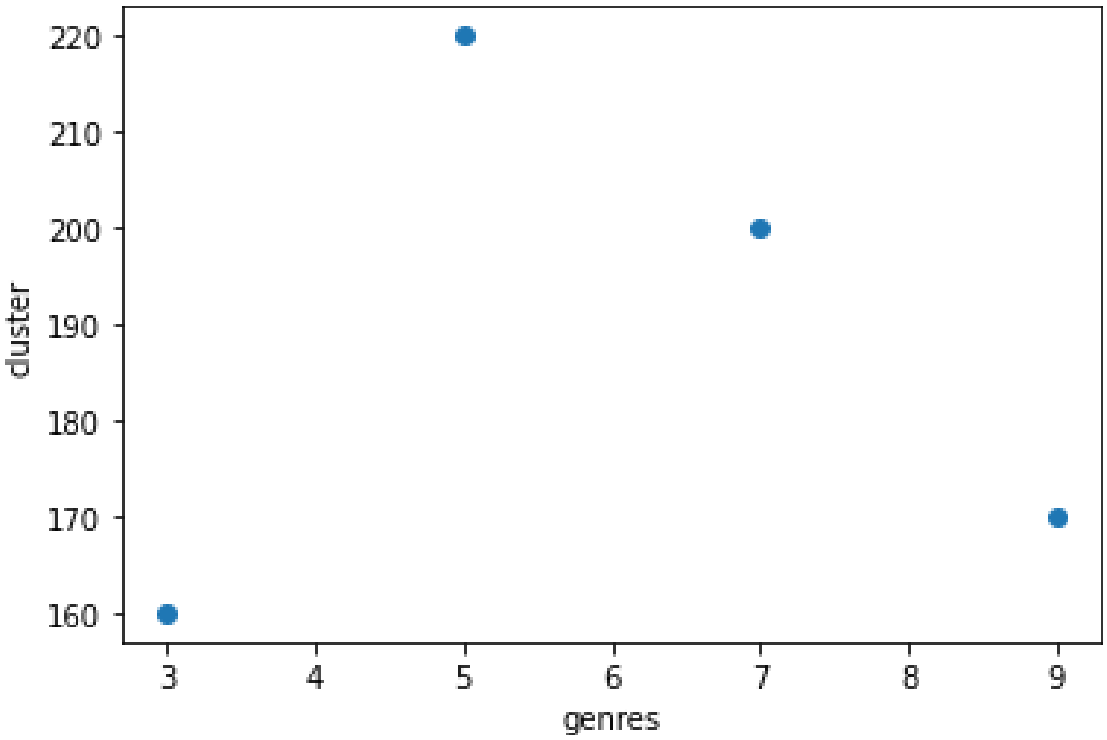

Figure 4.11 Relation of genres and cluster (20 docs).

第5章　結論

本研究では、最大9つのジャンルを持つ文書に対してBERTの文書分類で使用される手法であるベクトルの平均値とCLSトークンによる次元削減、そしてクラスタ分析による次元削減を行った。これらの文章分類の精度の指標として正解率を用いた。

この研究を行うに際し、データはlivedoorニュースコーパスを用いてlivedoorニュースとして掲載されていた9ジャンルの文書データで取得した。これを東北大学研究チームにより作成された学習済みBERTモデルのトークナイザを用いて形態素解析を行った。その後、今回の主題であるクラスタ分析、ベクトルの平均値、CLSトークンを用いたそれぞれの次元削減、文章分類を行った。また、クラスタ分析では行うクラスタリングのクラスタ数の変化による正解率の推移を調べるためクラスタ数を10から300まで10ずつ区切って実験を行い、ジャンル数と文書数についても8パターン行い、結果を比較した。

今回の実験結果からは、クラスタ分析におけるクラスタ数はジャンル数と文書量が増えるほど多くしたほうが良い結果が得られ、ジャンル数が少ないときはBERTで従来使われてるCLSトークンを用いた方法よりも正確な次元削減ができる、この条件下においてクラスタ分析による文章分類が有効であることがわかった。

一方、ジャンル数が多い場合は他の次元削減方法よりも正解率が低く、他と比べたクラスタ分析の有効性を感じることはできなかった。また、一つの問題点として文書数とジャンル数により精度のいい結果を出せるクラスタ数が変動するため、それを求めない限りは使用しにくいとも感じられた。

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) livedoor ニュースコーパス, ロンウィット,[アクセス日:2022/1/13]
https://www.rondhuit.com/download.html

2) cl-tohoku/bert-japanese,[アクセス日:2022/2/12]
https://github.com/cl-tohoku/bert-japanese

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) シグモイド関数を理解してみる,Yaju3D,[アクセス日:2022/2/4]
https://yaju3d.hatenablog.jp/entry/2018/10/31/013702

5) 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,[アクセス日:2022/2/4]
https://ledge.ai/bert/

6) 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,[アクセス日:2022/2/13]
http://www.gifu-nct.ac.jp/elec/deguchi/sotsuron/hattori/

7) 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,[アクセス日:2022/2/13]
http://www.gifu-nct.ac.jp/elec/deguchi/sotsuron/nagao/

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,[アクセス日:2022/2/13]
http://www.gifu-nct.ac.jp/elec/deguchi/intro2020.html