レビューを用いた映画の特徴量化と学習方法による言語モデル間の性質比較

宮下　天祥†　　荘司　慶行††　　藤田　澄男†††

† 青山学院大学大学院 理工学研究科　〒 252 – 5258 神奈川県 相模原市 中央区 淵野辺
†† 青山学院大学 理工学部 情報テクノロジー学科　〒 252 – 5258 神奈川県 相模原市 中央区 淵野辺
††† ヤフー株式会社　〒 102–8282 東京都 千代田区 紀尾井町 東京ガーデンテラス紀尾井町 紀尾井タワー
E-mail: †tensho@sw.it.aoyama.ac.jp, ††shoji@it.aoyama.ac.jp, †††sufujita@yahoo-corp.jp

あらまし　本研究では，映画情報サイトで利用可能なデータ，ベクトル化の手法によって生成される映画の特徴量が，異なるタスクにおいてどのような性質を表すかを明らかにする．そのために，様々な方法で生成したベクトルを，様々なタスクで網羅的に評価した．この際，ベクトル化に用いるデータによる違いを検証するために，映画のメタデータを用いた場合と，ユーザ投稿のレビューを用いた場合で比較した．ベクトル化の手法による違いを検証するために，プーリング，ベクトルの選択，要約的手法それぞれで，レビューとメタデータから映画をベクトル化した．最後に，生成されたベクトルの違いを明らかにするために，分類，回帰などの複数のタスクで，客観的，主観的の 2 つの側面から評価した．

キーワード　情報検索，映画レビュー，BERT，特徴量生成

1　はじめに

現代の日本人は，インターネット上に溢れた，多量の動画に囲まれた状態で生活している．近年，ストリーミングによる配信サービスやサブスクリプション（定期購読）による見放題サービスなどの流行を受けて，誰もがインターネット上で映像コンテンツを見るようになってきている．総務省の調査によると，全人口のおよそ 30 パーセント弱は，有料動画配信サービスを日常的に利用しているとされる [1]．

このような映像配信サービスの隆盛を受けて，ウェブ上の映像メディアの検索や推薦のニーズは増加傾向にある．従来のように映画館で映画を視聴する場合，自分が次に見る映画を選ぶには，その時点で上映されている数本の映画の中から，興味のある 1 本を選び出せればよかった．一方で，近年の動画のオンライン配信では，昔の映画から最新の映画まで，数多くの映画から，次に見るべき映画を決定する必要がある．現代のユーザは日々，膨大な数の映画の中から，自分の観たい映画を選んでいる．そのため，映画の推薦や検索のためのアルゴリズムは，より高度で便利なものが求められるようになってきている．

このような流れの中で，機械学習技術の発展は，より高度なアルゴリズムを用いた，ユーザのニーズを満たす映画の検索や，次に見るべき映画の推薦を可能にしている．こうした中で，とくに重要性が高まっているのが，映画をベクトルで表すエンベッディング技術である．機械学習を用いたアイテムの検索や推薦などのタスクにおいて，アイテムをベクトルとして扱うことは，現代ではもっとも一般的なアプローチのひとつである．例えば，オンラインショッピングサイトでは，ユーザの購買履歴などからそのユーザの興味がありそうな商品の推薦を行っている．この際，商品説明などの情報や，ユーザの購買行動をベクトル化し，ユーザの興味と近い特徴を持つ商品のマッチングをする．また，Learning to Rank においては，まず，ランキング対象となるアイテムをベクトル化してからランキング化する．

しかし，映画は映像作品であるため，映画の特徴を表すような情報を扱いづらい．このようなアイテムをどのような手法でベクトル化すべきかは未知数である．

本研究では，映画のあらすじや解説，映画につけられたメタデータなどのテキストデータを用いて，アイテムを特徴ベクトルとして表現することについて議論する．具体的には，どのようなデータを使って，どのような手法でベクトル化したときに，どのようなタスクで使えるかを明らかにする．

そのために，

- データ，
- ベクトル化手法，
- タスク

について，それぞれの組み合わせごとに，どの程度の精度で計算が行えるかを分析する．

具体的なデータの比較項目として，何をもとにベクトルを生成したかを比較する．映画の特徴を表すデータソースとして利用可能なデータは，映像だけではない．例えば，メタデータ，あらすじ，ユーザ投稿のレビューなど扱うデータによって生成されるベクトルにどのような特徴が現れるか明らかにする必要がある．

ベクトル化手法の具体的な比較項目として，使用する言語モデル，ベクトルの選択方法，ベクトルの集約方法を比較する．

近年，自然言語処理技術の発展が急速に進んでいる．それに

1：総務省 情報通信白書 令和 3 年版：
https://www.soumu.go.jp/johotsusintokei/whitepaper/r03.html

伴い，自然言語を扱う様々な手法が提案されている．古典的には TFIDF や LDA などのトピックモデル，Word2Vec などの分散表現，BERT に代表される Transformer ベースの言語モデルなどが提案されている．手法によってベクトルの性質に違いが表れるかを明らかにする必要がある．

映画のベクトルを用いて映画の検索や推薦をする際，様々な用途や目的が考えられる．例えば，あるユーザが過去に「スターウォーズ」を見たことがあるとする．このユーザに対して「スターウォーズ」に似た映画を推薦したい場合に，制作陣やキャストが似た映画を探したい場合もあれば，映画のジャンルが似た映画，ストーリ展開が似た映画を探したい場合など，主観的であったり，客観的であったり，様々である．そこで，本研究では，使用するデータ，使用するベクトル化手法，目的とするタスクについて，どのような場合に，どういったデータどのようにベクトル化するのが良いかを網羅的に実験し，明らかにする．

本研究で用いたデータは，映画のタイトルや製作年度，監督などの情報を含む映画のメタデータ，自然言語で書かれたあらすじと解説，ユーザ投稿のレビューである．これらのデータのベクトル化には Devlin ら [1] が提案した BERT を用いる．この際，BERT を追加学習なしで用いる場合と，追加学習ありで用いる場合について比較する．追加学習の方法として，BERT を映画レビューコーパスで追加事前学習する場合について検証する．また，映画がユーザ投稿のランキングに含まれるかどうかの判定タスクと，2 つのレビューが同じ映画に対して書かれたものかの判定タスクの 2 つで BERT をファインチューニングした場合についても検証する．

次に，ベクトル選択の方法を比較する．一般的に，BERT の出力した文単位のベクトルを扱う方法として，各単語トークンのベクトルの平均プーリングが用いられる．しかし，BERT を用いて自然言語分野の下流タスクを実行する際，特殊トークンである [CLS] トークンのベクトルを用いることが効果的であることが知られている．そのため，[CLS] トークンのベクトルを用いた場合についても比較する．

最後に，ベクトルの集約方法について比較する．ベクトル化に使用するデータとして，レビューを選択した場合には，BERT により生成したベクトルを映画単位で集約する必要がある．そのため，ベクトル集約の方法として，平均プーリング，最大プーリング，LexRank による要約について比較する．

最後に，これらのデータ，手法で生成した映画の特徴ベクトルの性質を，客観的，主観的に評価するためのタスクを行った．具体的なタスク一覧を表 1 に示す．タスクは大きく，分類タスク，回帰タスク，ベクトル空間が意味を持つかの評価タスク，二値判定タスクの 4 つに分けられる．これらのタスクについてそれぞれ客観的，主観的な評価を行った．

本文は本章を含めた全 6 章から構成される．第 2 章では本研究に関連する研究を提示する．第 3 章では本研究で比較を行った，ベクトルの生成手法についてそれぞれ述べる．第 4 章ではいくつかの下流タスクで，各ベクトル化手法によるベクトルの精度比較の詳細を述べる．第 5 章では評価実験を通して得られた結果について考察し，第 6 章でまとめと今後の展望について述べる．

2 関 連 研 究

本研究は，レビューを用いて映画の検索やクラスタリングのための特徴量を生成する研究である．これを可能にする技術として，BERT と呼ばれる言語モデルを用いる．また，映画に関するレビューは多人数が投稿しているものである．このようなものは，それぞれで矛盾が生じたり，文の構造が異なったりするため，単純な手法での集約が困難である．そのため，本節では，これらについて説明する．

2.1 レビューを用いたアイテムの検索や推薦

レビューには，そのアイテムを使った人の感想や，実際に使わないとわからないことなど，メタデータからは得られない情報が含まれることがある．このような，情報を用いたアイテム検索やクラスタリングは，従来的に行なわれている．

Zheng ら [2] は，ユーザのレビューと商品のレビューの両方を用いて，そのユーザがその商品をどの程度好むかを推定する手法を提案している．Amazon のレビューデータ等を用いた実験からレビューデータを用いることの有効性を示している．

Jakob ら [3] は，自由記述のレビューから意見抽出することで，映画の推薦の精度を向上させる研究を行っている．IMDB から抽出したレビューデータを用いた映画推薦の実験から，特に推薦におけるコールドスタート問題に対して，レビューデータを用いることが有効であることを明らかにしている．

そのため，本研究では，映画の特徴を表す情報として，ユーザ投稿のレビューを用いることの効果について検証する．

2.2 言語モデルを用いた特徴量生成

自然言語を機械で扱うための言語モデルは，数多く提案されている．中でも近年，Transformer ベースの言語モデルが，自然言語処理分野において主流になっている．

本研究では，Devlin ら [1] が提案した BERT を用いる．これは，文脈を読むことを可能にした自然言語処理モデルである．この BERT を，文書の特徴量生成に応用することは一般的に行われている．

例えば，Zhuang ら [4] は金融分野の文書を正しく特徴量化するために，金融分野のコーパスでファインチューニングした BERT モデルである FinBERT を提案している．金融ニュースや金融サイトからクロールしたデータを用いてファインチューニングを行い，それぞれのタスクにおける評価実験からその有効性を実証している．

また，Emily ら [5] は臨床ドメインのテキストを特徴量化するために，実際の診断書などを用いて BERT をファインチューニングした ClinicalBERT を提案している．MedNLI データセットなどを用いた臨床ドメインの下流タスクから，ファインチューニングの効果を実証している．

Chalkidis ら [6] は，裁判の判例などから抽出したテキストを用いて BERT のファインチューニングを行い，法律関係の文

表 1　評価タスク一覧

	分類タスク	回帰タスク	二値判定タスク	ベクトル間類似度比較
客観的	ジャンル推定	年代推定		
主観的	イメージワード推定	評点推定	ユーザタグ推定	映画類似度判定

章を特徴量化することに特化した BERT モデルである Legal-BERT を提案している．EURLEX57K データセットなどを用いた法律分野の下流タスクから，提案したモデルが最も高い精度を発揮したことを明らかにしている．

Wu ら [7] は，BERT を用いて対話文の特徴量を生成するために，対話文のコーパスを独自の事前学習方法によって事前学習した TOD-BERT を提案している．MetaLWOZ データセットなどの対話文からなるコーパスを用いた 4 種類の下流タスクから，提案した事前学習の効果を明らかにしている．

しかし，本研究は，体裁の整った文書からの特徴量生成とは異なり，多人数の自由記述による意見から特徴量を生成する必要がある．

2.3　言語モデルによる意見集約

言語モデルによる意見集約の方法は様々である．例えば，単文書要約（SDS）と呼ばれる，単一の文書を言語モデルを用いて要約するタスクを，言語モデルを用いて解決する研究が行われている．

Liu ら [8] は，初めての BERT を用いた抽出型要約モデルである BERTSUM を提案している．これは，文書に含まれる各文が要約文に必要かどうかを判定するタスクによってファインチューニングを行ったモデルである．CNN/Daily Mail news highlights dataset 及び，the New York Times Annotated Corpus を用いた実験から，提案手法の有効性を示している．

Lewis ら [9] は，sequence-to-sequence モデルを，BERT に学習させるためのノイズ除去オートエンコーダである BART を提案している．このモデルを，要約用データセットを用いて学習することで，BERT による要約を可能にしている．

Zhang ら [10] は，重要度の高い文をマスクし，マスク部分を予測させるタスクによりファインチューニングを行った抽象型要約モデルである PEGASUS を提案している．XSum や CNN ／ DailyMail データセットを用いた実験から，提案手法が，従来手法を上回る性能であったことを明らかにしている．

しかし，本研究では，映画に投稿された多人数のレビューから映画の特徴量を生成する．そのため，複数文書要約（MDS）のアプローチが必要となる．しかし，複数文書要約は，単一文書の要約とは異なる．まずは，文体が安定しないという点である．複数文書要約の場合，それぞれの文書を同じ人が書いているとは限らないため，文体にばらつきが生じる可能性がある．次に，文書間で矛盾が起こりうるという点である．同じ物事について言及している文書であったとしても，著者の立場や知識量によって文書間で矛盾が生じる可能性がある．また，複数の文書を一度に言語モデルに入力しようとすると，入力長が長く，計算量が大きくなりすぎるという問題もある．これらの問題に対処するための研究は，従来的に行われている．

入力長の問題を扱っている研究として，beltagy ら [11] は，長い文書を入力可能な Transformer ベースのモデルである，Longformer を提案している．これは，トークンのアテンション行列をスパース化することで，アテンションの計算量を削減し，長い文書の入力を可能にしている．WikiHop や，TravelQA などの，長い文書を用いた分類タスクや QA タスクなどから，長い文書を切り落とさずに入力することが有効であることを示している．

また，Zaheer ら [12] は，同じく長い文書を入力可能な Transformer ベースのモデルである，Big Bird を提案している．WikiHop や，TravelQA などの，長い文書を用いた分類タスクや QA タスクなどから，長い文書において，従来の Transformer ベースのモデルより高性能であったことを明らかにした．

Grail ら [13] は，長文を分割して入力し，各入力の情報を Transformer 層の間に組み込んだ双方向の GRU で伝播することで，長文を理解可能なモデルを提案している．提案したモデルを用いた，PubMed や arXiv などの長文の要約タスクから，提案手法が長文理解に有効であることを明らかにしている．

Jin ら [14] は，単語，文，文書単位の表現を生成し，同じ粒度間の意味関係と異なる粒度間の相互作用を用いて複数文書からの要約を生成する手法を提案している．Multi-News データセットを用いた実験から手法の有効性を明らかにしている

また，xiao ら [15] は，文書間の情報の接続と集約を学習可能なタスクでファインチューニングを行ったモデルによる複数文書要約の手法を提案している．いくつかのドメインのニュース記事や，Wikipedia などのコーパスを使った要約タスクによる実験から提案手法が有効であることを示している．

しかし，今回は正解となる映画レビュー文からの要約のためのデータセットを用意することが困難なため，要約のためにモデルをファインチューニングする手法は使えない．

また，商品レビューの意見集約に取り組んだ例として，angelidis ら [16] は，各レビュー文のアスペクト抽出と極性分析によって顕著な意見を識別し，要約する手法を提案している．6 つの製品ドメインの Amazon レビューを用いた要約タスクから，提案手法の有効性を明らかにしている．

しかし，本研究で用いる映画のレビューにはアスペクトの情報がないため，アスペクト抽出器を学習できない．

3　提 案 手 法

本研究では，映画の特徴を表すベクトルの生成方法を，どのようなデータ，どのようなベクトル化手法を用いるかで比較する．そして，これらの方法で生成したベクトルを用いて，いくつかの下流タスクを実行することで，それぞれのベクトルの性質の違いを明らかにする．実際の研究の流れを図 1 に示す．

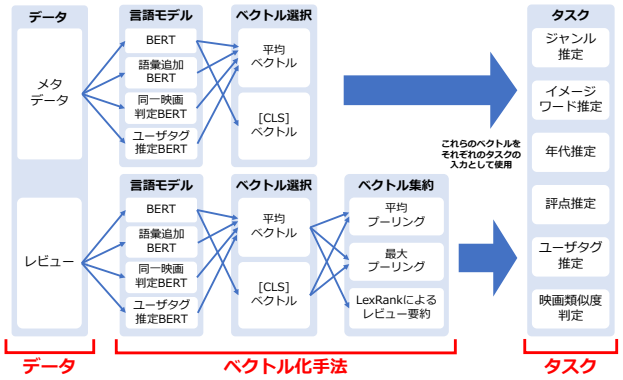

図 1　研究の概要図．データ，ベクトル化手法，タスクの 3 つの組合わせを網羅的に検証し，それぞれのタスクにおけるそれぞれのベクトルの性質の違いを明らかにする．

本研究では，ベクトル化に使用するデータとして，メタデータとレビューを用いる．今回は，これらのデータをベクトル化するために，Devlin ら [1] が提案した BERT を用いる．この際，BERT を追加学習すると，生成されるベクトルに性質の違いが表れるかを検証する．また，レビューの投稿数は映画によってばらつきがあり，映画によっては，一度にすべてのレビューを言語モデルに入力できないという課題がある．そのため，複数のレビューを集約して 1 つのベクトルにする必要がある．しかし，レビューは多人数の意見を含んでいるため，内容の矛盾などの問題が起こりうる．そこで，レビューの集約方法によって生成されるベクトルに性質の違いが表れるかについても検証する．本章ではこれらについて述べる．

3.1　言語モデル

本研究は映画という特定の分野に関する研究である．そこで，一般的なコーパスで訓練された事前学習済みモデルである BERT を映画のドメインに適応させると，生成されるベクトルにどのような性質の違いが現れるか検証する．また，BERT モデルを特定のタスクでファインチューニングした時の，ベクトルの性質の違いについても検証する．

3.1.1　映画レビュー文による追加事前学習

映画情報サイトに投稿された映画のレビュー文を用いて，BERT のマスク言語モデルによる追加の事前学習をする．

3.1.2　同一映画判定タスクによるファインチューニング

2 つのレビュー文を入力とし，この 2 つのレビューが同じ映画に対して書かれたものかを判定する二値判定タスクにより BERT をファインチューニングする．

3.1.3　ユーザタグ推定タスクによるファインチューニング

ユーザ投稿の映画ランキングのタイトルを，そのランキングに含まれる映画のタグとみなし，それぞれの映画に付けられたタグが，映画を正しく表すかどうかを判定する二値判定タスクにより BERT をファインチューニングする．

3.2　ベクトルの選択

入力文を BERT でベクトル化する際，まずは入力文をトークナイザにより単語レベルに分割してからトークン化する．そして，このトークンの集まりを BERT に入力することで，各トークンに対応したベクトルが得られる．これらのベクトルを文単位で集約する際，一般的には平均プーリングが用いられる．しかし，Nogueira ら [17] は BERT により得られた文脈情報を下流のタスクへと活用する際に，[CLS] トークンのベクトルを下流のタスクの入力として用いている．本研究は，映画のレビュー文から得られた文脈情報を用いて，様々な下流のタスクに応用するための研究であるため，[CLS] トークンのベクトルを用いる手法についても比較する．

3.3　ベクトルの集約方法

映画の特徴量を生成するために，レビュー文を BERT でベクトル化する．この際，レビュー文すべてを一度に BERT に入力することはできない．そのため，まずはレビュー単位でベクトル化を行い，その後映画単位で集約する．今回比較した集約方法は，

- **プーリング**：平均プーリング，最大プーリング，
- **要約**：LexRank によるレビュー要約，

である．本節では，これらについて説明する．

3.3.1　プーリング

今回，多数のベクトルのプーリング手法として，平均プーリングと最大プーリングの 2 つを比較する．平均プーリングは，ベクトルの各次元の平均を取り，それを集約ベクトルとして扱う手法である．最大プーリングは，ベクトルの各次元の最大値を取り，それを集約ベクトルとして扱う手法である．

3.3.2　要　　約

映画に投稿されたレビューの集約方法として，要約を用いる．今回は，LexRank を用いて事前に多人数のレビューを要約したものをベクトル化する手法を検証する．具体的には，まずそれぞれのレビュー同士のコサイン類似度を求める．次に類似度が 0.3 を超えるレビュー同士をグループに分ける．これらのグループの中で，最もレビューの数が多いグループに含まれるレビューを重要度の高いレビューとしてスコア付けする．スコアの高いレビュー上位 20 件を結合したものを要約とする．ただし，この際すでに要約に含まれるレビューとコサイン類似度が 0.15 より高いレビューは，冗長性排除のために除く．

4　評 価 実 験

本研究で提案した手法による生成ベクトルが，タスクによってどのような性質を示すか検証するために，様々なタスクによる実験を行った．

4.1　データセット

ベクトル化に用いる映画情報として，Yahoo!映画のウェブサイトから，レビューが 10 件以上ついた映画を 15,000 件程度収集した．映画に関するレビュー情報として，Yahoo!映画から 170,000 件程度のデータを収集した．二値判定タスクの正解データとして，Yahoo!映画から 10,000 件程度の「まとめ」データを収集した．その中で，特定の観点を含まない「まとめ」を除いた 7,000 件程度のデータを研究に用いた．

4.2 比較手法

使用するデータ，ベクトルの集約方法，使用する言語モデルによって，生成されるベクトルにどのような違いが現れるかを明らかにするために，表 2 で示される手法で比較を行った．

解説文の平均ベクトル ：解説文の平均ベクトルは，映画のメタデータである映画の解説文をベクトル化したものを映画のベクトルとして用いる手法である．解説文のベクトルとして，各単語のベクトルの平均を取ったものを用いる．

解説文の [CLS] ベクトル ：解説文の [CLS] ベクトルは，映画のメタデータである映画の解説文をベクトル化したものを映画のベクトルとして用いる手法である．解説文のベクトルとして，[CLS] ベクトルを用いる．

レビュー平均プーリング ：レビュー平均プーリングは，レビューを BERT でベクトル化し，各次元の平均を取ったものを映画のベクトルとして用いる手法である．レビューのベクトルとして，各単語のベクトルの平均を取ったものを用いる．

レビュー最大プーリング ：レビュー最大プーリングは，レビューを BERT でベクトル化し，各次元の最大を取ったものを映画のベクトルとして用いる手法である．レビューのベクトルとして，各単語のベクトルの平均を取ったものを用いる．

レビューの [CLS] ベクトルの平均 ：レビューの [CLS] ベクトルの平均は，各レビューの [CLS] トークンのベクトルの各次元の平均を取ったものを映画のベクトルとして用いる手法である．

レビューの [CLS] ベクトルの最大 ：レビューの [CLS] ベクトルの最大は，各レビューの [CLS] トークンのベクトルの各次元の最大を取ったものを映画のベクトルとして用いる手法である．

LexRank によるレビュー要約 ：LexRank によるレビュー要約は，1 つの映画に対して投稿された複数のレビューを，LexRank を用いて要約し，要約されたレビュー文をベクトル化したものを映画のベクトルとして用いる手法．要約されたレビュー文のベクトルとして，各単語のベクトルの平均を取ったものを用いる．

映画レビュー BERT（メタデータ） ：映画レビュー BERT（メタデータ）は，実験に用いる映画のレビュー文を用いて，マスク言語モデルによる追加事前学習を行った BERT モデルで，メタデータの解説文をベクトル化する手法である．解説文のベクトルとして，各単語のベクトルの平均を取ったものを用いる．

映画レビュー BERT（レビュー） ：映画レビュー BERT（レビュー）は，実験に用いる映画のレビュー文を用いて，マスク言語モデルによる追加事前学習を行った BERT モデルで，レビューをベクトル化する手法である．この際，ベクトルの集約方法として平均プーリングを用いる．また，レビューのベクトルとして，各単語のベクトルの平均を取ったものを用いる．

同一映画判定 BERT（メタデータ） ：同一映画判定 BERT（メタデータ）は，2 つのレビューが同じ映画に対するレビューかどうかを判定する二値判定タスクによりファインチューニングを行った BERT モデルで，解説文をベクトル化する手法である．解説文のベクトルとして，各単語のベクトルの平均を取ったものを用いる．

同一映画判定 BERT（レビュー） ：同一映画判定 BERT（レビュー）は，2 つのレビューが同じ映画に対するレビューかどうかを判定する二値判定タスクによりファインチューニングを行った BERT モデルで，レビュー文をベクトル化する手法である．この際，ベクトルの集約方法として平均プーリングを用いる．また，レビューのベクトルとして，各単語のベクトルの平均を取ったものを用いる．

ユーザタグ推定 BERT（メタデータ） ：ユーザタグ推定 BERT（メタデータ）は，ユーザ投稿の映画ランキングのタイトルを，そのランキングに含まれる映画のタグとみなし，それぞれの映画に付けられたタグが，映画を正しく表すかどうかを判定する二値判定タスクによりファインチューニングを行った BERT モデルで，解説文をベクトル化する手法である．解説文のベクトルとして，各単語のベクトルの平均を取ったものを用いる．

ユーザタグ推定 BERT（レビュー） ：ユーザタグ推定 BERT（レビュー）は，ユーザ投稿の映画ランキングのタイトルを，そのランキングに含まれる映画のタグとみなし，それぞれの映画に付けられたタグが，映画を正しく表すかどうかを判定する二値判定タスクによりファインチューニングを行った BERT モデルで，レビュー文をベクトル化する手法である．この際，ベクトルの集約方法として平均プーリングを用いる．また，レビューのベクトルとして，各単語のベクトルの平均を取ったものを用いる．

これらの言語モデルを用いて，メタデータをベクトル化する場合と，レビューをベクトル化する場合について比較した．

4.3 実験タスク

ベクトルの性質の違いを明らかにするため，大きく分けて以下の 4 種類のタスクで実験を行った．

- **分類タスク**：分類タスクは，複数のラベルを推定するタスクである．
- **回帰タスク**：回帰タスクは，数値を推定するタスクである．
- **ベクトル間類似度比較**：ベクトル間類似度比較は，ベクトル間の距離を比較するタスクである．
- **二値判定タスク**：二値判定タスクは，0 か 1 かの二値を判定するタスクである．

また，本研究で使用するメタデータは客観的なデータであるのに対し，レビューは主観的なデータである．そのため，これらのデータによる性質の違いを明らかにするため，客観的なタスクと主観的なタスクで比較した．

最終的な評価タスクは表 1 で示される．

4.3.1 ジャンル推定

ジャンル推定は，映画のメタデータである映画のジャンルを推定する分類タスクである．「ホラー」,「ファンタジー」などの 17 種類のジャンルから構成される．

表 2 比 較 手 法

手法名	データ	選択ベクトル	集約方法	言語モデル
解説文の平均ベクトル	解説文	平均ベクトル		BERT 日本語モデル
解説文の [CLS] ベクトル		[CLS] ベクトル		
映画レビュー BERT（メタデータ）				映画レビューによる追加事前学習を行った BERT
同一映画判定 BERT（メタデータ）				同一映画判定による追加学習を行った BERT
ユーザタグ推定 BERT（メタデータ）		平均ベクトル		ユーザタグ推定による追加学習を行った BERT
レビュー平均プーリング	レビュー文	[CLS] ベクトル	平均プーリング	BERT 日本語モデル
レビュー最大プーリング			最大プーリング	
レビューの [CLS] ベクトルの平均			平均プーリング	
レビューの [CLS] ベクトルの最大			最大プーリング	
LexRank によるレビュー要約			LexRank	
映画レビュー BERT（レビュー）		平均ベクトル		映画レビューによる追加事前学習を行った BERT
同一映画判定 BERT（レビュー）			平均プーリング	同一映画判定による追加学習を行った BERT
ユーザタグ推定 BERT（レビュー）				ユーザタグ推定による追加学習を行った BERT

4.3.2 イメージワード推定

イメージワード推定は，映画のレビューにつけられたイメージワードを推定する分類タスクである．これは，ユーザの映画に対するイメージを表すもので,「泣ける」,「かっこいい」など 20 種類のワードから構成される．今回は映画ごとに，最もつけられた割合の多いイメージワードを映画のイメージワードとする．

4.3.3 年 代 推 定

年代推定は，映画のメタデータである映画の製作年度を推定する回帰タスクである．

4.3.4 評 点 推 定

評点推定は，ユーザが映画に対して投稿した，映画の評点の平均値を推定する回帰タスクである．評点は，ユーザが 1 から 5 までの 5 段階で評価したものを使用する．

4.3.5 ユーザタグ推定

ユーザタグ推定は，ユーザ投稿の映画ランキングのタイトルを，そのランキングに含まれる映画のタグとみなし，このタグを推定するタスクである．具体的には，各映画に付けられたタグが，その映画を表す正しいものであるかどうかを推定する二値判定タスクである．

4.3.6 映画類似度判定

映画類似度推定は，比較元となる映画のベクトルと，比較対象となる 2 つの映画のベクトルを比較し，どちらの方がベクトル間距離が近くなるかを比較するタスクである．比較対象の 2 つのうち，どちらのほうが比較元と似ているかの正解データは，被験者実験により作成する．具体的には，比較元のベクトルと 2 つの比較対象のベクトルとのコサイン類似度をそれぞれ算出し，値が大きい方を予測ラベルとする二値判定タスクである．

4.4 実　　装

本研究では，主に Python を用いてシステムの実装をした．レビューや解説文のベクトル化には，東北大学 乾研究室の日本語 Wikipedia 事前学習モデル [2] を用いた．BERT をレビュー文で追加事前学習する際には，BERT の BertForMaskedLM クラス [3]を用いた．学習のパラメータは，学習率を 0.00002 とし，そのほかはデフォルトの値を利用した．追加事前学習の epoch 数は 3 回を予定していたが，過学習となったため 1 回とした．同一映画判定とユーザタグ推定によるファインチューニングでは，BERT の BertForSequenceClassification クラスを用いた．ファインチューニングのパラメータは，学習率を 0.00002 とし，そのほかはデフォルトの値を利用した．ファインチューニングの epoch 数は 3 回を予定していたが，過学習となったため 1 回とした．実験タスクの推論に用いたニューラルネットワークは，Python 向けの機械学習用ライブラリである Keras[4]を用いて実装した．全てのタスクで入力層と出力層を含む 4 層ネットワークを用い，隠れ層の活性化関数には Relu 関数を用いた．年代推定と評点推定で用いたネットワークの入力層は 768 次元，隠れ層は 64 次元，出力層は 1 次元である．Optimizer は Adam，損失関数は平均絶対誤差（MAE）を用いた．学習の epoch 数は 8 回に設定した．ジャンル推定で用いたネットワークの入力層は 768 次元，隠れ層は 64 次元，出力層は 17 次元であり，活性化関数は softmax 関数を用いた．Optimizer は Adam，損失関数は CategoricalCrossentropy を用いた．学習の epoch 数は 5 回に設定した．イメージワード推定で用いたネットワークの入力層は 768 次元，隠れ層は 64 次元，出力層は 20 次元であり，活性化関数は softmax 関数を用いた．Optimizer は Adam，損失関数は CategoricalCrossentropy を用いた．学習の epoch 数は 5 回に設定した．ユーザタグ推定で用いたネットワークの入力層は 1536 次元，隠れ層は 64 次元，出力層は 2 次元であり，活性化関数は sigmoid 関数を用いた．Optimizer は Adam，損失関数は BinaryCrossentropy を用いた．学習の epoch 数は 5 回に設定した．各タスクにおけるモデルの性能は 5 分割交差検証により評価した．

2：GitHub「cl-tohoku/bert-base-japanese-whole-word-masking」：
https://github.com/cl-tohoku/bert-japanese

3：Hugging Face「BERT」：
https://huggingface.co/transformers/v3.1.0/model_doc/bert.html

4：Keras「Keras Documentation」：
https://keras.io/ja/

表 3　比較手法のそれぞれのタスクにおける結果

	ジャンル推定	イメージワード推定	年代推定	評点推定	ユーザタグ推定	映画類似度推定
	適合率	適合率	MAE	MAE	適合率	適合率
解説文の平均ベクトル	0.61	0.39	**6.09**	0.40	0.76	0.65
解説文の [CLS] ベクトル	0.55	0.33	6.99	0.39	0.71	0.54
映画レビュー BERT（メタデータ）	0.60	0.38	6.40	0.38	0.80	0.65
同一映画判定 BERT（メタデータ）	0.61	0.38	6.30	0.38	0.76	0.65
ユーザタグ推定 BERT（メタデータ）	0.60	0.38	6.55	0.38	0.70	0.64
レビュー平均プーリング	**0.63**	**0.48**	7.02	**0.32**	0.80	0.70
レビュー最大プーリング	0.50	0.35	8.89	0.41	0.78	0.50
レビューの [CLS] ベクトルの平均	0.53	0.36	8.13	0.37	0.75	0.70
レビューの [CLS] ベクトルの最大	0.43	0.27	9.12	0.41	0.75	0.60
LexRank によるレビュー要約	0.44	0.29	9.80	0.42	0.76	0.55
映画レビュー BERT（レビュー）	0.61	0.46	7.26	0.33	**0.81**	**0.73**
同一映画判定 BERT（レビュー）	0.55	0.37	8.04	0.36	0.75	0.70
ユーザタグ推定 BERT（レビュー）	0.42	0.26	8.74	0.41	0.70	0.48

4.5 実験結果

本節では，実験の結果について述べる．比較手法のそれぞれのタスクにおける結果を表 3 に示す．まず，ジャンル推定，イメージワード推定においては，レビュー平均プーリングが最も適合率が高かった．しかし，ジャンル推定においては，メタデータを用いた手法がレビューを用いた手法に近い精度を出した．一方で，イメージワード推定においては，レビュー平均プーリングと映画レビュー BERT（レビュー）が，他の手法に大きな差をつけて高精度という結果となった．年代推定においては，メタデータをベクトル化した手法が最も MAE が低かった．また，レビューを用いた手法は，メタデータを用いた手法に比べて精度が低いという結果となった．評点推定においては，レビュー平均プーリングが最も MAE が低かった．また，イメージワード推定の時と同様に，レビュー平均プーリングと映画レビュー BERT（レビュー）が，他の手法に大きな差をつけて高精度という結果となった．ユーザタグ推定タスクにおいては，映画レビュー BERT（レビュー）が最も適合率が高かった．しかし，メタデータを用いた手法と，レビューを用いた手法で大きな差がでないという結果となった．ベクトル間類似度比較タスクにおいては，映画レビュー BERT（レビュー）が最も適合率が高かった．

5 考　　察

本章では実験の結果について考察する．まずは，ベクトル化に用いるデータが生成されるベクトルに与える影響について考察する．表 3 から，レビューを平均プーリングした手法が多くのタスクにおいて，高精度であることが分かる．このことから，レビューにはメタデータより多くの映画の特徴を表す情報が含まれていると考えられる．しかし，年代推定やジャンル推定といった，客観的な情報を推定するタスクにおいては，メタデータをベクトル化した手法が比較的高精度という結果となった．特に，年代推定においては，メタデータを用いた手法が一貫してレビューを用いた手法を上回る性能を発揮している．このことから，それぞれのベクトルには，ベクトル化に用いたデータの性質が強く反映されていると考えられる．

次に，ベクトル化手法の違いが生成されるベクトルに与える影響ついて考察する．表 3 から，平均を用いた手法がすべてのタスクにおいて最も高精度であることが分かる．このことから，下流タスクへ用いるベクトル及び，複数のベクトルの集約方法としては平均を用いることが最も単純で良い方法であると考えられる．また，LexRank によるレビュー要約は，要約しない場合と比較して大幅に精度が落ちた．これは，要約する段階で映画を表す情報を捨ててしまっているため，うまく映画の特徴を捉えられず精度が落ちてしまったと考えられる．

最後に，BERT を追加学習することの効果についての考察を述べる．BERT をレビュー文によって追加事前学習することでユーザタグ推定，映画類似度推定といった，人のつけたラベルを推定するタスクにおける精度が向上した．このことから，映画レビューという一般的な文章とは形式の異なるテキストの構造や映画という特定分野の語彙を BERT が学習したことで，より人間の感覚に近いベクトルが生成可能になったと考えられる．一方で，ファインチューニングをした BERT は全てのタスクで精度が落ちてしまうという結果となった．これは，BERT を特定のタスクに特化させることで，汎用性を失ってしまったためだと考えられる．そのため，ファインチューニングはあくまでも最終タスクにおける精度向上のためにするべきだということが明らかとなった．

6 まとめと今後の課題

本研究では，映画をベクトル化する際に用いるデータ，ベクトル化手法によって，生成されるベクトルにどのような性質の違いが表れるかを，6 つのタスクを通して検証した．そして，実験の結果から，ベクトル化に用いるデータの性質が生成ベクトルに反映されること，BERT による特徴量生成の際には，平均を用いることが効果的であること，映画レビューを用いた追加事前学習によって人間の感覚に近いベクトルが生成可能なこと，ファインチューニングによってモデルの汎用性が失われてしまうことを明らかにした．

今後の課題として，より多人数の意見を効果的に集約する方法を模索することが挙げられる．今回は，最終的に多人数のレビューを集約する方法として最も効果的な手法は平均プーリングという結果となった．しかし，単純な平均では，少数派の意見が集約後のベクトルに残りにくいことが考えられる．本研究の目的は，多人数の意見から映画に関する情報を集約することであるため，意見の多様性を尊重することが重要である．そのため，単純な平均では失われてしまうような情報を効果的に集約する方法を模索する必要がある．

謝　　辞

本研究の一部は JSPS 科研費 21H03775，21H03774，22H03905 による助成，ならびに 2022 年度国立情報学研究所共同研究 22S1001 の助成を受けたものです．ここに記して謝意を表します．

文　　献

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[2] Lei Zheng, Vahid Noroozi, and Philip S. Yu. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, p. 425–434, New York, NY, USA, 2017. Association for Computing Machinery.

[3] Niklas Jakob, Stefan Hagen Weber, Mark Christoph Müller, and Iryna Gurevych. Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations. In *Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion*, TSA '09, p. 57–64, New York, NY, USA, 2009. Association for Computing Machinery.

[4] Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. Finbert: A pre-trained financial language representation model for financial text mining. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 4513–4519. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Special Track on AI in FinTech.

[5] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pp. 72–78, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[6] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 2898–2904, Online, November 2020. Association for Computational Linguistics.

[7] Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 917–929, Online, November 2020. Association for Computational Linguistics.

[8] Yang Liu. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*, 2019.

[9] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, Online, July 2020. Association for Computational Linguistics.

[10] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[11] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[12] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, Vol. 33, pp. 17283–17297. Curran Associates, Inc., 2020.

[13] Quentin Grail, Julien Perez, and Eric Gaussier. Globalizing BERT-based transformer architectures for long document summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1792–1810, Online, April 2021. Association for Computational Linguistics.

[14] Hanqi Jin, Tianming Wang, and Xiaojun Wan. Multi-granularity interaction network for extractive and abstractive multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6244–6254, Online, July 2020. Association for Computational Linguistics.

[15] Wen Xiao, Iz Beltagy, Giuseppe Carenini, and Arman Cohan. PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5245–5263, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[16] Stefanos Angelidis and Mirella Lapata. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3675–3686, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[17] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.