DEIM Forum 2022 C41-2

個人の投稿したランキングを用いたBERTによる映画のキーワード検索

宮下　天祥†　　荘司　慶行†　　藤田　澄男††　　Martin J. Dürst†

† 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 5–10–1
†† ヤフー株式会社　〒102–8282 東京都千代田区紀尾井町 1-3 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,duerst}@it.aoyama.ac.jp, †††sufujita@yahoo-corp.jp

あらまし　本論文では,「泣ける映画ベスト 10」や「見ると旅に出たくなる映画トップ 10」などの，任意のキーワードで表される観点に基づく映画ランキングを自動生成する手法を提案する．一部のオンライン映画レビューサイトには，映画のメタデータや映画に対するレビュー文に加えて,「私的な○○の映画トップ 10」などの私的なランキングを「まとめ」と題して投稿できる機能がある．そこで本研究では，BERT を用いて映画をベクトル化し,「まとめ」のランキングを正解として，ランキング名とそれに含まれる映画の関係をニューラルネットワークで学習した．キーワードと映画の関連性をラベル付けする大規模なクラウド評価を通じて，提案手法の精度と有用性を明らかにする．

キーワード　情報検索，ユーザレビュー，BERT, Learning to Rank

1　はじめに

近年，オンデマンドやサブスクリプション（定額配信）による映画配信サービスの増加に伴い，映画の選び方が変化してきている．例えば，旧来の映画館での映画視聴であれば，その時点で上映されているものの中から，見たい映画を選ぶ．このような選び方であれば，選択肢はたかだか数本なので，それぞれの映画が見るべき映画かどうかを，しらみつぶしに調べることが可能であった．また，近年一般的であった DVD レンタルショップを利用する場合でも，店頭にある限られた候補から映画を探すか，あるいは，見たい映画をあらかじめ決めておいて，それが店頭に在庫されているかを探すことが一般的であった．一方で，映画配信サイトでは，利用者は過去に公開されたほとんどすべての映画から，次に見るべき 1 本を，その場で選ぶ必要がある．このような場合に，1 本 1 本の映画について，見るべきかどうかを順次調べていくことは難しい．

加えて，映画の視聴されかたも変化してきている．劇場のスクリーンやテレビの前で身構えて視聴するというよりは，パソコンやスマートフォンで気軽に視聴する場合が増加している．何かの作業中に「ながら見」をしたり，映画の再生速度を上げて視聴するなどの，より日常に溶け込んだ形式での映画視聴が，一般化しつつある [1]．

このような映画視聴形態の変化を受けて，より気軽に映画を検索できる技術の需要が高まっている．例として,「作業中に流しておけるような，映像のきれいな映画が見たい」,「とにかくアクションが派手な映画を見たい」などの，任意のキーワードで映画をランキングして探したいという検索ニーズがある．しかしながら，従来の映画検索では，このような「映像がきれい」や「アクションが派手」などのキーワードで検索した場合，それに該当する映画を発見することはできない．これは，従来の映画情報サイトにおける映画検索が，映画のあらすじやメタデータを対象としているためである．映画情報サイトに登録されたメタデータとしては，映画の原題，制作国，監督，音楽，出演俳優などの情報のほか，解説，映画のあらすじなどが一般的である．しかし，映画の解説やあらすじは，あくまで第三者の視点からその映画がどういうものかを説明する文章に過ぎない．そのため，その映画を見た人がその映画にどのような感想を持つかといった情報は含まれない．

このような状況下で，その映画が「映像がきれいかどうか」など，視聴者がどう感じたかを判断するうえで，レビュー情報は重要な情報源である．しかし，現代の動画配信サイト上の膨大な映画から,「映像がきれいな映画」を探そうとした場合を考えると，それぞれの映画に対して投稿されたレビューを 1 つずつ読んでいき，自分の見たい映画であるかを判断することは，現実的ではない．

そこで，本論文では，BERT を用いて，ユーザの作成したランキングと映画のレビューから映画とクエリの関係の深さを学習し，自由なクエリにもとづいて映画をランキングする手法を提案する．そのために，ユーザが独自に，自分なりの映画ランキングを投稿可能なサービスに注目した．一部の映画レビューサイトには，個人が好きな映画をリスト化して公開する機能がある．たとえば，Yahoo! Japan の運営する映画情報サイトである「Yahoo!映画」では,「まとめ」という名前で，好きな観点から映画を最大 10 本登録できる [2]．この「まとめ」には,「私的 泣ける映画 ベスト 10」などのタイトルがつけられており，最大 10 本の映画が登録されている．そこで，ある映画に対して,「この映画は，タイトルに『泣ける』を含むまとめに登場しそうか」など,「まとめ」のタイトルと，それぞれの映画の関係を，データから学習する．こうすることで，任意のキーワードに対して，そのキーワードをタイトルに含む「まとめ」に登場しそうかを推定し，映画を任意のキーワードに対してランキン

1：朝日新聞出版刊 AERA 1/18 号（Kindle 版）:「『2 時間も一つの画面に集中できない』現象が拡大中」

2：Yahoo! 映画 新着まとめ：https://movies.yahoo.co.jp/matome/

グ可能にする．

これを可能にするための手法として，

- BERT を用いたクエリとレビューの分散表現の生成，
- 機械学習によるクエリとレビューの関係の深さの学習，
- ユーザの入力したクエリからのランキングの作成

の 3 つのステップからなるアルゴリズムを実際に作成した．

1 つ目のステップは，BERT を用いたクエリとレビューの分散表現の生成である．ここでは BERT と呼ばれる自然言語処理モデルを用いて，クエリやレビューといった自然言語からなる文を，言葉の意味などを数値化したベクトルの分散表現に変換する．これにより，文の意味をデータとして扱えるようにする．

2 つ目のステップは，機械学習によるクエリとレビューの関係の深さの学習である．ここでは，レビューのベクトルから生成した映画のベクトルとクエリのベクトルを学習データとした深層学習を行う．この学習には，Yahoo 映画サイトの「まとめ」という機能を用いる．「まとめ」とは，ユーザが任意のタイトルをつけて，そこに映画を 10 件まで登録できる機能である．例えば，「泣ける映画 10 選」といったタイトルの「まとめ」に，そのユーザが泣けると思った映画を 10 件登録できる．この「まとめ」のタイトルをクエリとみなし，映画がそのまとめに入っているか，入っていないかを正解とした二値分類タスクを行う．これにより，入力されたクエリが特定の映画と一致するか，しないかの二値分類を行うことのできる学習済みモデルを生成する．

3 つ目のステップでは，実際にユーザが入力したクエリを BERT を用いてベクトル化して，あらかじめ用意しておいた映画を表すベクトルと結合したものを，2 つ目のステップで作成した学習済みモデルに通すことによって，入力クエリと映画が一致する確率を出力する．そして，それらの結果からランキングを生成する．

こうして作成した手法により，実際に映画を任意のキーワードでランキングできるようになる．ランキングの精度と有効性を確認するため，被験者実験を行った．提案手法と複数の比較手法で生成したランキングに登場する映画が，クエリとどのくらい関係性が深いかを，被験者に評価させた．

本文は本章を含め全 6 章から構成される．第 2 章では本研究に関連した研究について整理する．第 3 章では本研究で提案する，レビューから生成した映画を表すベクトルとクエリを深層学習によってマッチングする手法について述べる．第 4 章では提案手法の評価を被験者実験により行う．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，口コミ（eWOM）を用いて映画を検索可能にする研究である．これを可能にする技術として，BERT と Learning to Rank を用いる．そのため，本節では，これらについて説明する．

2.1 口コミ（eWOM）を用いたアイテム検索

本研究は，映画サイトに投稿されたレビューからユーザの求めるものに近い映画を抽出し，ランキング化するという研究である．このような，口コミに着目したアイテム検索の例はいくつか存在する．

例として，Ramanand ら [1] は，レビューや購入者アンケートなどの文書から，商品・サービスに関する提案や購入意向を示す「wishes」を抽出する手法を提案している．レビュー等から抽出した「wishes」は, 商品やサービスの改善, 質の向上だけでなく，顧客の求める商品の推薦, 提示などにも応用可能である．

また，杉木ら [2] は，レビュー文の係り受け解析を行い，意見情報を抽出することで，自由なクエリに適応する商品を検索する手法を提案している．実際の宿泊施設検索システムを用いた実験から，従来の手法では対応が困難であったクエリからの，商品の検索が可能であることを明らかにしている．そこで本研究では，自由なクエリに適応する映画を検索するために，ユーザのレビュー情報を用いている．

2.2 BERT を用いた情報検索

BERT は，Devlin ら [3] が提案した，文脈を読むことを可能にした自然言語処理モデルのことである．BERT は，文中の単語の意味を，高い精度でベクトルの分散表現に変換できるため，様々な分野に応用されている．

BERT はその用途によって学習済みの言語モデルをファインチューニングすることで，様々なタスクに特化させたうえで用いられている．文の類似度に注目したファインチューニングの例として，Nils ら [4] は，BERT の改良版である Sentence Bert を提案している．SentEval を用いた文埋め込みの性能評価実験において，この手法は，より文の意味を効果的にベクトル化可能であることを明らかにしている．

他に，特定の分野に特化させた例として，Zhuang ら [5] は，金融に関係する単語に特化した BERT のファインチューニングモデルである，FinBERT を提案している．金融分野の質問応答文書からなるデータセットである FiQA を用いた性能評価実験から，FinBERT が，金融に関係する単語の意味を，効果的にベクトルに反映させたことを明らかにしている．

別の例として，Shibata ら [6] は，BERT を用いて，クエリと関連性の高い FAQ を検索する手法を提案している．localgovFAQ と StackExchange データセットを用いた実験から，この手法がより正確な FAQ 検索を可能にしたことを明らかにしている．

本研究でも，映画レビューという特殊な領域に BERT を用いているが，今回の実験では，ファインチューニングを行っていないシンプルな BERT で行った．

また，BERT を情報検索に応用した例も存在する．Yang ら [7] は，文書のアドホック検索に BERT を適応させる手法を提案している．TREC Microblog Tracks を用いた実験から，マイクロブログに存在するような短い文書の検索には，BERT を用いた手法が効果的であることを明らかにしている．

Yunqiu ら [8] は，一般的なクエリに比べてはるかに長いクエ

リからの検索である，リーガルケース検索のための BERT モデル，BERT-PLI を提案している．COLIEE 2019 の関連判例検索タスクから，この手法が長い文書の意味をより正確に理解できることを明らかにした．

Zhuolin ら [9] は，英語のクエリと多言語の文書との関連性の学習に BERT を用いることで，英語のクエリから多言語の文書を検索する手法を提案している．IARPA の MATERIAL プログラムによって提供されたクエリと検索コーパスを用いた，リトアニア語のテキストと音声の文書を検索するタスクから，この手法が他の手法を凌駕する性能を発揮したことを明らかにしている．

本研究は，映画という，映像を扱った研究であるため，映画の内容を文章で扱うことが困難である．そこで，ユーザの投稿したレビューデータを用いた．また，レビュー文と短いクエリでは，文の性質が異なるため，単純なベクトル同士の類似度比較が困難である．そこで，クエリとレビュー文のマッチング手法として，Learning to Rank を用いた．

2.3 Learning to Rank

Learning to Rank は，ランキング問題を機械学習によって解決する手法であり，ランキング学習とも呼ばれる．ランキング学習には主に，ポイントワイズ法，ペアワイズ法，リストワイズ法の 3 つのアプローチ [10] が存在しており，本研究ではポイントワイズ法を用いる．ランキング学習は，クエリと関連度の高い文書のランキングを生成することが目的であり，主に情報検索に用いられている．

特定のトピックに属する文書を検索する例として，Amir ら [11] は，主張を裏付けるための根拠の検索に，BERT とランキング学習を用いた手法を提案している．FEVER dataset を用いた根拠を示す文の検索タスクにおいて，この手法が，他の手法と比較して，最も優れた性能を発揮したことを明らかにしている．

また，Yu ら [12] は，与えられた質問に対する回答を含んだ文書の検索に，ランキング学習を用いた手法を提案している．TREC の標準的なベンチマークデータセットを用いた実験から，この手法が回答文検索において最先端の手法に匹敵する性能であったことを明らかにしている．

Fabio ら [13] は，コンテンツベースの画像検索にランキング学習を用いた手法を提案している．Corel GALLERY Magic Stock Photo Library 2 と Caltech の 2 つのデータセットを用いた評価実験から，この手法が従来のランキング手法を上回る性能であったことを明らかにしている．

本研究は，映画というアイテムを検索する研究であるため，ランキング学習をアイテム検索に応用した例についてもいくつか挙げる．Shubhra ら [14] は，E-コマースサイトにおける商品検索にランキング学習を適用した手法を提案している．Web 上からランダムに抽出されたクエリと商品情報を用いた検索タスクにおいて，ランキング学習手法の一つである LambdaMART を用いた手法が最も高いスコアを出したことを明らかにしている．

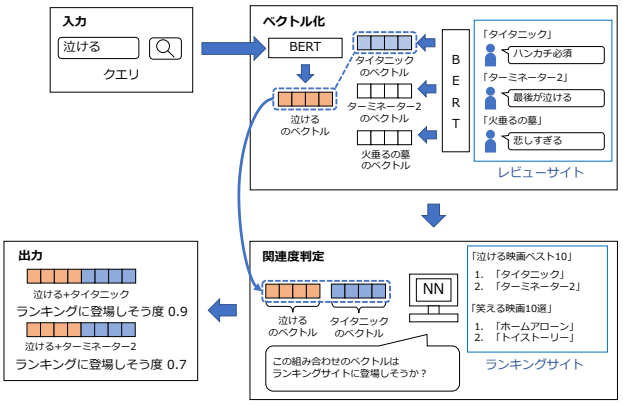

図 1　手法の概要図．ユーザ投稿ランキングを正解として，クエリとアイテムの関連度を推定する

Blake ら [15] は，精度の低い GPS による，位置情報検索の精度を上げるために，ランキング学習を用いた手法を提案している．マンハッタンにおける実際の施設へのチェックイン情報を用いた実験で，提案手法が最も位置特定の精度が高いことを明らかにしている．Jing ら [16] は，エンティティ検索におけるランキング学習の有用性の検証を行っている．DBpedia のデータを用いたエンティティ検索の実験から，ランキング学習が文書検索だけでなくエンティティ検索にも有用であったことを明らかにしている．したがって，本研究でも，クエリから映画をランキング化するための手法として，ランキング学習を用いる．

本研究に先駆けて，Kurihara ら [17] は，レビュー情報を用いた映画検索に，ランキング学習を用いた手法を提案している．この研究では，LambdaMART という決定木に基づく手法を用いて，映画のランキング順位を推定するモデルを提案している．一方で，本研究では，クエリとアイテムの関連度そのものを，ニューラルネットワークで学習するという，新しい方法で映画をランキング可能にすることを目指している．

3 提 案 手 法

本研究では，入力された任意のキーワードクエリに対して，映画をランキング化して返す，検索アルゴリズムを提案する．手法の概要を図 1 に示す．この手法では，はじめに，個人作成の映画ランキングを収集し，ランキングのタイトルとそのランキング中に含まれる映画を抽出する．次に，ランキングタイトルに含まれる語と，映画を，それぞれ BERT でベクトル化する．そして，2 つのベクトルをつなぎ合わせたものを Deep Learning によって，「この映画は，このキーワードをタイトルに含むランキングに登場するか否か」という 2 値分類の問題として学習する．

こうして学習したモデルに，任意のクエリと映画を入力することで，その映画がそのクエリをタイトルに含むランキングに登場する可能性を推定できるようになる．この登場する可能性を，すべての映画に対して計算することで，任意のキーワードクエリに対する映画のランキングが作成できる．

3.1 映画を表すベクトルの生成

映画サイトには，映画に対して，様々なユーザの投稿したレビューが含まれている．本研究では，ユーザの投稿したレビューに，映画の性質を表す情報が含まれていると仮定している．そのために，映画をベクトルとして表現する際に，映画のメタデータや映像情報でなく，その映画に対するレビュー文を映画を表すベクトルとして用いる．

はじめに，レビュー文をベクトル化するための前処理を行う．まずは，BERT での処理を可能にするために，レビューを文単位で分割した．BERT によってベクトル化できる文字数には制限があるため，長いレビュー文をそのままベクトル化することはできない．したがって，レビュー文を句読点や「!」や「?」などの記号で分割した．また，不必要な文字や記号の除去も行った．「」や空白などの文字が入っていると，ベクトル化の際にノイズとなりかねないので，あらかじめ除去した．

次に，前処理後の各レビュー文について，実際に BERT で 768 次元の分散表現形式のベクトルを算出する．BERT では，計算量の上限から，長い文章を 1 度に分散表現化することができない．そこで，ある映画 m に対するすべてのレビューに含まれるそれぞれの文 $r \in R(m)$ をベクトル化し，それをプーリングすることで映画の特徴ベクトルとして扱う．任意のレビュー文 r を表すベクトルを $BERT_r(r)$ と表した際に，映画 m のベクトル $\mathbf{v}(m)$ は，

$$\mathbf{v}(m) = \frac{\mathrm{BERT}_r(r)}{|R(m)|} \qquad (1)$$

と定義できる．これは，その映画に対するすべてのレビューに含まれる各文のベクトルを平均したものである．

3.2 データセットのクレンジングと正解作成

特定のクエリからの映画検索を可能にするために，クエリと映画を結びつける正解データが必要になる．そこで，本研究では，クエリと映画を結びつける正解データとして，インターネット上で個人が自分独自の映画ランキングを投稿可能なサイトのデータを用いる．本論文では，実験の際に，Yahoo!映画の「まとめ」に登録された個人のランキングを学習用データとして用いたので，ユーザ投稿ランキングを「まとめ」と呼んで説明する．

近年では，インターネット上で個人の作成した，任意の映画ランキングを共有することが一般的である．例えば，個人 blog などに注目しても,「私的 泣ける映画トップ 10」などのランキングを見つけることができる．このようなランキングを登録できるウェブサービスも一般的になってきている．例として，世界最大手の映画レビューサイトである IMDB では「Watchlist[3]」という機能が提供されている．この機能を使うと，ユーザは好きなタイトルのリストを作成でき，そこに好きな映画を登録できる．日本国内でも，Yahoo!映画には「まとめ」と呼ばれる機能があり，好きなタイトルのリストを作り，最大 10 件の映画を登録できる．このような個人作成のランキングには,「泣ける映画 10 選」,「好きな任侠映画ベスト 10」のようなタイトルのものが多く含まれる．本研究では，このような，誰かが何らかの観点に基づいて作成したランキングを収集し，それを正解データとして用いることで，クエリと映画の関係を学習する．

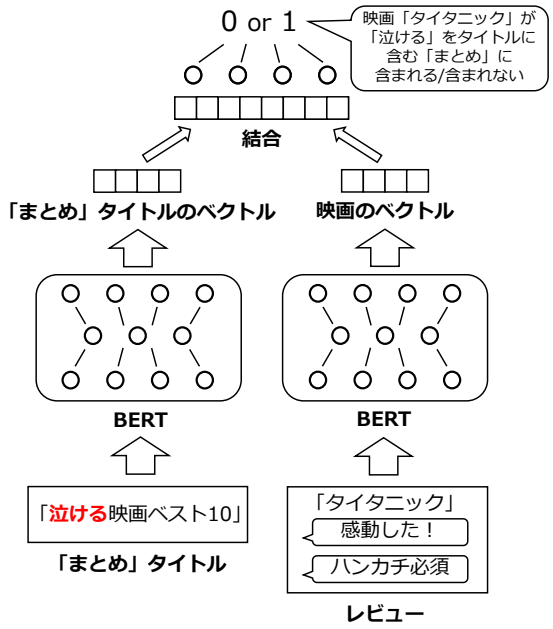

図 2 映画が，クエリをタイトルに含んだ「まとめ」に含まれているか，含まれていないかを予測するモデルの学習手順

オンラインの「まとめ」のデータは，多くの個人が投稿したデータであるので，学習に用いるためには様々な前処理が必要になる．はじめに，特定の観点に基づかないランキングを除外した．ユーザの投稿した「まとめ」は様々であり，単にそのユーザが良いと思った映画をランキングにしたものも存在する．たとえば,「わたしの人生のマイベスト」や,「私的 2002 年ベスト 10」などの映画は，特定の観点に基づかない．そのような「まとめ」には，クエリと映画の関係性を表す情報がないため，取り除いた．また，ほとんどの「まとめ」タイトルに用いられている「映画」という単語や，数字なども取り除いた．

次に，学習用データの作成について述べる．学習データは，入力データと正解からなる．入力データは，前処理を行った「まとめ」のタイトルと，映画のタイトルである．これらをベクトル化し，結合して，実際のニューラルネットワークへの入力として用いる．そして，映画がその「まとめ」に含まれているかいないかを，0 か 1 の 2 値でラベル付けしものが正解である．まずは,「まとめ」に含まれている映画を用いて正例を作成した．次に,「まとめ」に含まれていない映画から負例を作成した．これを深層学習に用いる学習用データとした．

3.3 映画とクエリの関係性の学習

実際に，キーワードと映画の関連の深さを，ニューラルネットワークで学習する．それぞれの映画と，キーワードが分散表現ベクトルとして表された際に，提案手法は，図 2 に表されるように，2 値分類を行うニューラルネットワークとして学習される．この際，映画とキーワードのそれぞれのベクトルを結合

3 : IMDB Watchlist: https://imdb.com/list/watchlist

表 1　学習に使用したネットワーク層の詳細

層	レイヤ数	活性化関数
全結合層	1,536	Relu
全結合層	64	Relu
全結合層	64	Relu
全結合層	2	Sigmoid

したうえで,「その映画が，キーワードをタイトルに含むまとめに登場しそうか」を推定するタスクで学習を行う．本節では，作成した学習用データを用いた，深層学習の手順を述べる．学習の際には，学習用データを訓練用データとテスト用データに分割して学習を行う．

まず，映画を表すベクトルと「まとめ」タイトルのベクトルを結合した 1536 次元のベクトルをニューラルネットワークへ入力する．このニューラルネットワークは，表 1 で表されるように，4 層の全結合層からなる．出力層は，その映画がそのキーワードを含む「まとめ」に登場するかの 2 値であるので，2 次元である．この時，損失関数 $\mathrm{loss}(p,q)$ は p を正解の値，q を予測値としたとき，

$$\mathrm{loss}(p,q) = -\sum_{x} p(x) log(q(x)) \qquad (2)$$

で表される．この損失関数によって得られた値から，ネットワークモデルの最適化を行う．

ここまでが，1 つの入力に対する手順である．これを，訓練用データすべてについて行った．

次に，この学習済みモデルの精度の評価を行った．精度評価は，テスト用データを入力として，先ほどの学習で用いた損失関数から損失を算出した．

3.4　ランキング生成

このように学習した言語モデルを用いて，実際に任意のキーワードクエリから，映画をランキングする．現在の手法では，クエリとすべての映画に対して総当たりで関連度を計算し，関連度順に映画をランキングする．

実際の手順として，まず，入力されたクエリを BERT でベクトル化する．次に，このクエリのベクトルに対して，すべての映画 1 本 1 本のベクトルを結合する．このようなクエリとそれぞれの映画ベクトルを結合したベクトルを，1 つずつ順に第 3.3 節で作成した学習済みモデルに通す．これにより，クエリを「まとめ」タイトルとしたとき，その映画を含んでいる確率を算出する．こうして，すべての映画に対して，クエリとの関連度で映画を順位付けできる．

4　評 価 実 験

提案手法から生成された映画のランキングが，よりユーザの感覚と一致しているか検証するために，被験者実験を行った．特定のクエリについて，各手法から生成されたランキングに登場する映画を，リスト化して，被験者にどれだけクエリと一致しているかを評価させた．

4.1　比 較 手 法

本研究の技術的貢献に結び付く仮説として，

H1　：レビュー情報を使うことで，映画のあらすじやメタデータなどに表れないようなキーワードで映画を検索可能である，

H2　：映画に関する文書と，クエリのそれぞれがベクトルとして表現された場合，コサイン類似度などの単純な類似度比較では不十分である

の 2 つが挙げられる．これらの仮説について検証するために，

- **提案手法**：レビュー文から生成した，映画を表すベクトルと，クエリを深層学習によりマッチングする手法，
- **映画単位で類似度比較**：レビュー文から生成した映画を表すベクトルと，クエリのベクトルを類似度比較する手法，
- **文単位で類似度比較**：レビュー文のベクトルと，クエリのベクトルを類似度比較する手法，
- **メタデータ**：映画サイトにあるメタデータのみを用いた手法，

という，提案手法を含む 4 つの比較手法を用意した．

提案手法は，第 3 節で説明した，映画のレビュー文から生成したベクトルとクエリのベクトルの関係の深さを，深層学習によって推定する手法である．この手法は，2 つの仮説 H1 および H2 の両方を反映する．

映画単位で類似度比較は，映画のレビュー文から生成したベクトルとクエリのベクトルを，単純にコサイン類似度により類似度比較する手法である．入力としてクエリを受け取ると，BERT を用いてクエリのベクトル化を行う．そして，クエリのベクトルと映画を表すベクトルのコサイン類似度を算出する．これの結果から，類似度の高い映画をランキングとして出力する．この手法は，仮説 H1 のみを反映する．

文単位で類似度比較は，映画を表すベクトルの生成を行う際に用いた，レビュー文単位のベクトルと，クエリのベクトルを類似度比較する手法である．入力されたクエリのベクトルと，Yahoo!映画に投稿されたレビュー文のベクトルのコサイン類似度を，総当たりで算出する．そして，類似度の高かったレビューの投稿された映画について，レビュー文とクエリの類似度に基づいてランキングする．同じ映画に投稿された複数のレビューがランキングに登場する場合，最も類似度の高いレビューから算出された類似度をその映画の類似度と扱った．この手法は，仮説 H1 を反映している．一方で仮説 H2 については，そもそも映画とクエリを比較する際に単純な類似度では計算が不十分であるので，映画とクエリを比較せず，レビューの 1 文とクエリを比較している．

メタデータのみに基づく手法は，Yahoo!映画のウェブサイトに登録されている，映画に関するメタデータを用いた手法である．映画に関する情報として，原題，製作年度，上映時間，製作国，ジャンル，監督，製作総指揮，脚本，音楽といったメタデータと，解説，あらすじなどの文章が存在する．まず，入力されたクエリと上記の情報について，テキストマッチを行う．

表 2　実験に用いたクエリとその特徴

クエリ	特徴
泣ける 笑える ショッキング	映画を観た人の感情を表す
デートに最適 子供向け	映画を観る状況や場面を表す
サスペンス アニメ	映画のジャンルを表す
ジブリ どんでん返し 北野武	映画の内容を表す

マッチした映画が存在する場合，それらの映画についている解説文，あらすじ文のベクトルと，クエリのベクトルのコサイン類似度を算出する．そして，類似度の高かった映画をランキングとして出力する．マッチした映画が存在しない場合，全ての映画について，クエリのベクトルと，解説文，あらすじ文のベクトルのコサイン類似度を算出する．そして，同様にランキング化を行う．本手法では，仮説 H1 も H2 も支持しない．

4.2　データセット

ランキングに使用する映画情報として，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 40,000 件程度のデータを収集した．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．

4.3　実験タスク

クエリと映画のペアについて，オンラインで適合度を回答させる被験者実験を行った．ラベル付けは 10 人の被験者によって行われた．初めに，被験者には，事前に作成した Google スプレッドシートのリンクが与えられた．1 枚のシートには，1 つのクエリが書かれており，各行に 1 つずつ，最大 40 件の映画のタイトルが記載されている．これは，1 つのクエリに対する，4 つの比較手法による映画のランキングの上位 10 件ずつを混ぜたものである．この際，複数の手法の結果に含まれた映画は，1 項目にまとめた．被験者は，それぞれの映画が，クエリとの関係性がどれだけ深いかを，1 から 5 の 5 段階で評価した．この際，知らない映画に対しては，インターネットでの検索を許した．

使用したクエリは 10 種類である．実際のクエリを表 2 に表す．それぞれの被験者は，クエリ 2 つ分，合計 80 件の映画についてラベル付けした．それぞれのクエリと映画のペアについて，2 名の被験者がラベル付けした．

4.4　実　　装

実験に使うデータの作成を行うためのシステムは，Python によって作成した．このシステムは，入力としてクエリを渡すと，そのクエリから各手法が出力したランキングの上位 10 件を csv 出力するものである．クエリをベクトル化する際に用いる BERT のモデルとして京都大学黒橋研究所の BERT 日本語 pretrained モデルを使用した．BERT でベクトル化を行う際の torkenizer として，MeCab を用いた．また，形態素解析を行う際の辞書として，新語や固有表現に対応している ipadic-neologd を用いた．学習済みモデル作成のための深層学習には，Python のライブラリである Keras を用いた．

表 3　各手法の適合率，nDCG

	p@1	p@5	p@10	nDCG
提案手法	0.50	0.56	0.58	0.64
メタデータのみ	0.50	0.46	0.40	0.60
映画単位で類似度比較	0.20	0.24	0.27	0.47
文単位で類似度比較	**0.80**	**0.74**	**0.72**	**0.75**

4.5　実 験 結 果

本節では，被験者実験の結果について述べる．本実験では，各手法の適合率，ランキング精度の指標である nDCG，実験に用いたクエリごとのスコアの平均を算出した．

はじめに，適合率によって各手法のランキング上位を評価した．適合率とは，正と予測したものがどのくらい正しかったかを示した指標である．本実験では，ユーザのつけた 5 段階の評価の中で，3 以上の評価を正解と見なしたときの適合率を算出した．p@k（precision at k）は，ランキングの k 位までの映画の適合率を表す．今回は，p@1，p@5，p@10 の 3 つの適合率を算出した．

各手法における適合率を表 3 に記す．p@1，p@5，p@10 全てにおいて，文単位での類似度比較手法が最も適合率が高かった．提案手法は，p@1，p@5，p@10 全てにおいて，文単位の類似度比較手法に次いで 2 番目であった．また，映画単位の類似度比較手法は，p@1，p@5，p@10 全てにおいて，適合度が最も低かった．

次に，ランキングの精度を確認するために，nDCG（normalized Discounted Cumulative Gain）による評価を行った．nDCG は，DCG (Discounted Cumulative Gain) を正規化したものである．ランキング中の i 番目の要素の適合度を rel_i，評価に用いる要素数を k としたとき，DCG は

$$\mathrm{DCG} = \mathrm{rel}_1 + \sum_{i=2}^{k} \frac{\mathrm{rel}_i}{\log_2 i} \tag{3}$$

と表される．理想のランキングに対する DCG を $DCG_{perfect}$ としたとき，nDCG は，

$$\mathrm{nDCG} = \frac{\mathrm{DCG}}{\mathrm{DCG}_{perfect}} \tag{4}$$

と表される．

各手法における nDCG を表 3 に記す．最も nDCG の値が高くなったのは，0.75 で文単位の類似度比較手法であった．最も nDCG の値が低くなったのは，0.47 で映画単位の類似度比較手法であった．

表 4　クエリごとの各手法の検索結果上位 10 位内の映画の被験者のつけた評点の平均（5 段階評価，最低 1，最大 5）

	泣ける	アニメ	サスペンス	ジブリ	デートに最適	子供向け	ショッキング	どんでん返し	北野武	笑える
提案手法	**3.7**	2.2	**3.6**	1.6	**3.5**	3.8	2.4	3.9	1.5	**3.0**
メタデータのみ	2.1	**3.8**	3.1	1.9	1.9	1.7	**3.7**	2.9	**4.0**	1.7
映画単位で類似度比較	2.1	2.2	1.7	1.7	2.9	2.7	2.4	2.9	1.8	1.7
文単位で類似度比較	2.9	2.6	**3.6**	**4.3**	**3.5**	**4.2**	3.4	**4.5**	3.6	1.7

表 5　深層学習によるクエリマッチングが特に有効に働いた「泣ける」というクエリに対する結果例

提案手法		映画単位で類似度比較	
映画名	評点	映画名	評点
奇蹟の輝き	3.0	初恋白書	2.0
花いちもんめ.	3.5	イエローヘア	1.5
グローリー・デイズ〜旅立ちの日〜	3.0	刑事コロンボ'90 ／超魔術への招待	2.0
ミリィ／少年は空を飛んだ	4.0	ライターをつけろ	1.5
クレヨンしんちゃん オトナ帝国の逆襲	4.5	パンと裏通り	2.0
ペイ・フォワード　可能の王国	3.5	新・刑事（デカ）まつり〜一発大逆転〜	1.5
素晴らしき哉，人生！	4.0	ヒューマン・トラフィック	3.0
ジャック	3.5	ジャングル・ジュース	2.0
ライフ・イズ・ビューティフル	4.0	極道黒社会　RAINY DOG	3.0
きみに読む物語	4.0	悪名市場	2.0

表 6　メタデータとのテキストマッチが特によく働いた「北野武」というクエリに対する結果例

提案手法		メタデータのみ	
映画名	評点	映画名	評点
映画クレヨンしんちゃん　雲黒斎の野望	1.0	菊次郎の夏	4.5
名探偵コナン　瞳の中の暗殺者	1.0	ソナチネ	5.0
ゴジラ・モスラ・キングギドラ　大怪獣総攻撃	1.5	3―4X10 月	3.5
劇場版 AIR	2.0	座頭市	4.0
ルパン三世　ヘミングウェイ・ペーパーの謎	1.5	アウトレイジ ビヨンド	4.0
生地獄	2.0	アウトレイジ 最終章	4.0
モスキート	1.0	BROTHER	3.5
スター・ウォーズ／帝国の逆襲　特別篇	1.0	TAKESHIS '	4.5
劇場版ラーゼフォン　多元変奏曲	2.0	HANA － BI	3.0
ルパン三世　カリオストロの城	1.5	アウトレイジ	4.0

4.6　クエリごとの被験者評点

本実験では，ユーザに提示された映画が，与えられたクエリとどれだけ関係性が深いかを 5 段階で評価させた．各手法におけるクエリごとの被験者のつけた評点の平均値を表 4 に記す．「泣ける」,「サスペンス」,「デートに最適」,「笑える」といったクエリでは，提案手法が最も正しく評点の高いアイテムを発見できた.「アニメ」,「ショッキング」,「北野武」といったクエリではメタデータのみの手法が最も高精度であった．その他のクエリについては，文単位の類似度比較手法が最も高精度であった．

また，提案手法による検索結果が高い評点を獲得した「泣ける」というクエリに対する結果例と，メタデータのみの手法による検索結果が高い評点を獲得した「北野武」というクエリに対する結果例を，それぞれ表 5，表 6 に示す．表 5 から，提案手法の発見したアイテムはすべて，被験者が「泣ける」度合いを 3 以上だと答えた．一方で，メタデータのみの手法では，ほとんどが 3 を下回った．表 6 から，メタデータのみの手法が発見したアイテムはすべて，被験者が「北野武」度合いを 3 以上だと答えた．一方で，提案手法では，すべての映画が 2 を下回った．

5　考　　察

実験により算出された適合率から，提案手法は，メタデータのみを用いた手法よりは精度が高いことを示した．しかし，全体的に，レビュー文単位での類似度比較手法には劣るという結果となった．これは，映画を表すベクトルというものが曖昧なものであり，映画の性質を正確に分散表現として表すことが困難であったためと考えられる．対して，レビュー文そのものからは，粒度の高い分散表現を得られるため，精度が高くなったと考えられる．nDCG は，文単位の類似度比較手法が最も高くなったことから，レビューを用いることの有用性は確認できた．

クエリごとのスコアからは，クエリの種類によって有用な手法が分かれることが示された．メタデータに情報が存在しそうな「アニメ」,「ショッキング」,「北野武」といったクエリについては，メタデータを用いた手法のスコアが高くなった.「泣ける」,「笑える」,「デートに最適」,「どんでん返し」といった，映画の性質を表すようなクエリについては，レビューを用いた検索手法のスコアが高くなった．このことから，レビューを用いることで，メタデータには存在しないような情報からの，映画の検索が可能になることを明らかにした．

6　まとめと今後の課題

本論文では，ユーザの作成したランキングと映画のレビューからクエリと映画を結びつけ，特定の観点からの映画の検索を可能にする手法を提案した．Yahoo!映画のデータを用いた深層学習によってクエリと映画の関係性学習し，クエリを入力すると映画のランキングを出力するシステムを作成した．いくつかのクエリを入力としたときの各手法の出力から，クエリとの一致度合いをスコア付けする被験者実験で，各手法の検索精度を評価した．結果として，レビュー文単位の類似度比較手法の p@k の値が，他の手法と比較して最も高くなった．一方，映画単位での類似度比較手法の p@k の値は，ほかの手法と比較して最も低くなった．また,「アニメ」,「ショッキング」,「北野武」といったクエリではメタデータのみを用いた手法のスコアが高かった.「泣ける」,「笑える」,「デートに最適」,「どんでん返し」といったクエリでは，レビューを用いた検索手法のスコアが高くなった．以上のことから，レビューを用いることで，より幅広いクエリからの検索が可能になることを明らかにした．一方で，クエリと映画のマッチングに深層学習を用いることについては，クエリと近い内容のランキングが学習データに含まれるかによって，精度が大きく変動する傾向が分かった．

今後の課題として，映画を表すベクトルの生成方法の見直し，および関係性の学習方法の改善が挙げられる．本研究で用いた，映画を表すベクトルは，多様なレビュー文の平均によって算出したため，ベクトルの表す特徴が平坦になってしまうことが考えられる．これにより．クエリとの正確なマッチングが困難となる可能性があるため，改善が必要である．また，クエリと映画との関係性の学習に用いたネットワーク層は単純なものであるため，今後さらなる実験を通して，よりタスクに有効なネットワークモデルを構築したいと考えている．

謝　　辞

本研究はJSPS科研費18K18161（代表:荘司慶行），21H03775（代表：大島裕明）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] J. Ramanand, Krishna Bhavsar, and Niranjan Pedanekar. Wishful thinking - finding suggestions and 'buy' wishes from product reviews. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pp. 54–61, Los Angeles, CA, June 2010. Association for Computational Linguistics.

[2] 杉木健二, 松原茂樹ほか. 消費者の意消費者の意見に基づく商品検索見に基づく商品検索. 情報処理学会論文誌, Vol. 49, No. 7, pp. 2598–2603, 2008.

[3] 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.

[4] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.

[5] 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.

[6] Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. Faq retrieval using query-question similarity and bert-based query-answer relevance. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, p. 1113–1116, New York, NY, USA, 2019. Association for Computing Machinery.

[7] Wei Yang, Haotian Zhang, and Jimmy Lin. Simple applications of bert for ad hoc document retrieval. *arXiv preprint arXiv:1903.10972*, 2019.

[8] Yunqiu Shao, Jiaxin Mao, Yiqun Liu, Weizhi Ma, Ken Satoh, Min Zhang, and Shaoping Ma. Bert-pli: Modeling paragraph-level interactions for legal case retrieval. In *IJCAI*, pp. 3501–3507, 2020.

[9] Zhuolin Jiang, Amro El-Jaroudi, William Hartmann, Damianos Karakos, and Lingjun Zhao. Cross-lingual information retrieval with BERT. In *Proceedings of the workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS2020)*, pp. 26–31, Marseille, France, May 2020. European Language Resources Association.

[10] Tie-Yan Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, Vol. 3, No. 3, p. 225–331, mar 2009.

[11] Amir Soleimani, Christof Monz, and Marcel Worring. BERT for evidence retrieval and claim verification. In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part II*, Vol. 12036 of *Lecture Notes in Computer Science*, pp. 359–366. Springer, 2020.

[12] Lei Yu, Karl Hermann, Phil Blunsom, and Stephen Pulman. Deep learning for answer sentence selection. *Proceedings of the Deep Learning and Representation Learning Workshop: NIPS-2014*, 12 2014.

[13] Fabio F. Faria, Adriano Veloso, Humberto M. Almeida, Eduardo Valle, Ricardo da S. Torres, Marcos A. Gonçalves, and Wagner Meira. Learning to rank for content-based image retrieval. In *Proceedings of the International Conference on Multimedia Information Retrieval*, MIR '10, p. 285–294, New York, NY, USA, 2010. Association for Computing Machinery.

[14] Shubhra Kanti Karmaker Santu, Parikshit Sondhi, and ChengXiang Zhai. On application of learning to rank for e-commerce search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, p. 475–484, New York, NY, USA, 2017. Association for Computing Machinery.

[15] Blake Shaw, Jon Shea, Siddhartha Sinha, and Andrew Hogue. Learning to rank for spatiotemporal search. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, p. 717–726, New York, NY, USA, 2013. Association for Computing Machinery.

[16] Jing Chen, Chenyan Xiong, and Jamie Callan. An empirical study of learning to rank for entity search. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, p. 737–740, New York, NY, USA, 2016. Association for Computing Machinery.

[17] Kosuke Kurihara, Yoshiyuki Shoji, Sumio Fujita, and Martin J. Dürst. Learning to rank-based approach for movie search by keyword query and example query. New York, NY, USA, 2021. Association for Computing Machinery.