

Learning to Rank を用いた キーワードや具体例を入力とする映画の検索

栗原 光祐[†] 莊司 慶行^{††} 藤田 澄男^{†††} Martin J. Dürst^{††}

[†] 青山学院大学理工学研究科理工学専攻知能情報コース 〒252-5258 神奈川県相模原市中央区淵野辺 5-10-1

^{††} 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: [†]kurihara@sw.it.aoyama.ac.jp, ^{††}{shoji,duerst}@it.aoyama.ac.jp, ^{†††}sufujita@yahoo-corp.jp

あらまし 本研究では、キーワードクエリとランキング例から、映画レビューデータを用いてそのキーワードとランキングの観点に適した映画ランキングを生成する手法を提案する。現在、Web 上には多数の映画レビューサイトや動画配信サービスが存在する。しかし、検索者が「旅行に行きたくなる映画」や「デートにぴったりな映画」など自由なクエリと各映画の評判に基づいて映画を検索または推薦する方法は確立されていない。そこで、ユーザが作成した大量の小規模な映画ランキングデータを用いてランキング学習を行い、それによって検索結果の候補となる映画リストを再ランキングさせることによって、自由なクエリから映画ランキングを生成可能にする。また、大量の小規模な映画ランキングデータを用いるかわりに、クエリに適したランキングの一例を検索クエリとして入力させることにより、その検索者の情報要求により適したランキングを生成可能にする。評価実験として、提案手法及び既存手法を用いて実際に映画ランキングを生成し、被験者アンケートによってそのランキングの有用性を評価した。また、実際に映画ランキングを入力とする映画検索システムを構築し、被験者に使用させることによってランキングを入力とする検索の是非について検証した。

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

1 研究背景

映画の評判情報は、鑑賞する映画を選定する際には非常に重要な情報源である。現在、Web 上には多数の映画レビューサイトや動画配信サービスが存在するが、それらの中で検索者の入力したクエリと各映画の評判に基づいて映画を検索または推薦する方法は確立されていない。多くの映画レビューサイトでは、映画のカテゴリごとの平均評点ランキングといった簡単な映画ランキングは提供されている。しかし、「旅行に行きたくなる映画」や「恋人と一緒に観たい」など、自由なキーワードを使って検索したい場合、膨大な映画の中からそのキーワードに適合する順に映画をランキングとして結果を得ることはできない。現状、そのような検索者の自由な情報要求をもとに映画を探そうとする場合、各映画に投稿されたレビューを1件ずつ読み、どの映画が自分の情報要求に適しているか判断する必要がある。これは非常に長い時間を要する作業である上、ストーリーの根幹にかかわる重要な情報（ネタバレ）を見てしまうリスクをとらなう。

そこで本研究は、検索者の自由な情報要求と各映画の評判情報に基づく新たな映画検索手法を提案する。これを実現するためには、大きく分けて、

- 多様な表現をもつ映画ごとの評判の集約、
- 観点ごとの映画の順位付けに関する意見の集約、
- 情報要求のより正確なクエリ表現

という3つの課題がある。

1 つめの課題は、先行研究 [1] において取り組んだものである。したがって本研究では、残る2つの課題について取り組む。

2 つめの課題は、ある特定の観点についての映画の順位付けに関する多人数の意見の集約である。先行研究で提案した Target-Topic Aware Doc2Vec (TTA-D2V) を用いて、自由なキーワードクエリからそのクエリの意に適した評判を得ている映画を探しだすことができる。しかしこの時点で得られるランキングは、映画のレビューとクエリとの言語的な関連性（分散表現間の \cos 類似度）に基づくものである。そのため、クエリとして入力された観点との関連度合が高い順（すなわち、「泣ける順」や、「旅行に行きたくなる順」）の検索結果が得られるとは限らず、映画のランキングとして最適であるとはいえない。

そこで本研究では、Learning to Rank [2] と呼ばれるアイテムの順序を最適化するための教師あり機械学習手法を用いて、各ユーザが自由な観点に基づいて作成した多数の小規模な映画ランキングを学習する。その結果得られたランキングモデルは、映画の順位付けに関する多人数の意見を集約したランキングモデルであると考えられる。このランキングモデルを用いて TTA-D2V による検索結果の順序を並び替えることにより、クエリとして入力された観点との関連度合が高い順の映画ランキングを得ることができる。

最後に3つめの課題として、検索者の情報要求をより正確に表現する方法について取り組んだ。検索者が映画を検索する際のクエリとして、単語または短文による自由なキーワードクエリは、もっとも単純で入力しやすいものである。しかし一方で、

単一のキーワードクエリでは、検索者の意図が表現しきれない場合もある。例えば、「懐かしい映画」というクエリの文面だけからでは、それが「古い年代に制作、公開された映画」を指すのか、「その検索者の幼少期や青年時代の情景や体験に近い映画」を指すのかを判別できない。

そこで本研究では、検索者が自らの情報要求を表現する手段として、キーワードではなく自らの情報要求に適した映画ランキングの例を入力させる例示検索手法も提案する。例えば、「懐かしい」というキーワードの代わりに、自分が懐かしいと感じた映画のトップ何件かを、例として入力可能にする。このアプローチでは、検索者が入力した1つのランキング例のみを正解として扱い、その場で Learning to Rank で学習する。得られたランキングモデルを用いて、TTA-D2Vの結果を再ランキングさせることで、キーワードクエリでは表現できない検索者に固有の観点によって映画を検索することを可能とする。

これら2つの提案手法により、自由なキーワードクエリまたは映画ランキングの例を用いて検索者の情報要求に適した映画をランキング形式で検索可能になると考えられる。検証のため、Yahoo!映画の実データを用いて評価実験を実施した。使用するデータはユーザーレビューデータおよびユーザーによる多数の映画ランキングデータである。評価実験は次の2つの方法によっておこなった。1つはあらかじめ用意したクエリによる検索結果を用いたクラウドソーシングによる大規模アンケート評価である。これにより、提案手法のうちキーワード検索手法が生成する映画ランキングの正確さを評価した。もう1つは実際に検索者にクエリまたはランキングの例を入力させて検索させ、その結果を評価させる実験である。これにより、提案手法のうち例示検索手法において、被験者がランキング例を入力することで自身の情報要求をより正確に表現可能かどうか、および被験者の情報要求に適した検索結果を生成可能かを評価した。

2 関連研究

本研究は、ユーザーレビューという Web 上の評判情報と Learning to Rank を用いた情報検索に関する研究である。そのため、Web 上の評判情報を用いる手段として近年主流である分散表現に基づくアイテム検索・推薦、Learning to Rank, および例示検索についてそれぞれ紹介し、論じる。

2.1 分散表現を用いたアイテム検索・推薦

本研究では、ある映画に対して投稿されたレビュー文を用いてその映画を分散表現化している。こうすることでアイテム同士の類似度を簡単に計算可能である。このことに着目した研究として以下のようなものがある。

Barkan ら [3] は、Item2Vec という、Word2Vec で用いられる Skip-Gram モデルをアイテムベースの協調フィルタリングに応用する手法を提案している。あるアイテムを表現する文脈情報としてそのアイテムと同時に購入されたアイテム集合を用い、Skip-Gram モデルを適用することでアイテムの分散表現を得る。

Phi ら [4] は、E コマースサイトにおける協調フィルタリングによるアイテム検索にアイテムの分散表現を用いている。ア

アイテムを単語、あるユーザの購入履歴を単語列すなわち文とみなし、Doc2Vec を適用することでアイテムとユーザの両方を分散表現化している。評価実験では、楽天市場のログデータからこの手法を用いた推薦システムを構築し、従来のアプローチに比べてヒット率が上昇したとしている。

映画においては同時購入といった記録が得られない代わりに、ユーザーレビューを得ることができる。そこで本研究では、先行研究 [1] において提案した Target-Topic Aware Doc2Vec (TTA-D2V) を用いて、各映画に投稿されたレビュー文から各映画の分散表現を得る。

2.2 Learning to Rank を用いたアイテム検索

Learning to Rank (LtR) は、ランキング問題を教師あり学習手法によって解く手法であり、ランキング学習とも呼ばれる。LtR の学習データセット S の構造は、

$$S = \{(q, i, r) \mid (q, i) \in P, r = \text{relevance}(q, i)\} \quad (1)$$

と表すことができる。ここで、 (q, i) はクエリ q とアイテム i のペア、 P はクエリとアイテムの全ペアの集合、 r はクエリ q とアイテム i の適合度ラベルである。クエリとアイテムのペア (q, i) をもとにその適合度 r を予測するようなモデルを学習することで、順序が未知のアイテム集合に対してその順序を推論可能となるため、特に Web 検索システムの最適化手法として盛んに用いられている。

Karmaker ら [5] は、LtR が E コマースサイトにおける商品検索において有用であるかどうかをいくつかの仮説を検証することによって検討している。それによれば LtR は商品検索手法としてロジスティック回帰や SVM に比べて有用であり、特に商品の購入率やクリック率を予測するように学習した場合において顕著であるとしている。

Ludewig ら [6] は、LtR を用いたホテルのランキング手法を提案している。ユーザの長期的な嗜好プロファイルを利用することなく、短期的なインタラクションからユーザの好みと検索の意図を導き出し、高精度なホテルランキングの提供に成功している。

LtR を用いた情報検索分野の既存の研究では、その多くがクリックログや購入履歴などを教師データとして利用する [7, 8]。対して、本研究ではユーザが自由に作成した多数のランキングを教師データとして利用する。これにより、クリックログや購入履歴など収集の難しいデータを使わずにランキングを最適化できると考えられる。

2.3 例示検索

例示検索 (Query by Example) とは、検索者が求める結果の例を検索クエリとして提示する検索方式である [9]。

例示検索の考え方は、音楽検索の分野においてとくに盛ん [10, 11, 12] で、1995 年には既に Ghias ら [13] によって鼻歌 (ハミング) によって音楽を検索する手法 (Query by Humming) が提案されている。

音楽・画像・映像などを含むマルチメディアコンテンツにおいては、アイテムをキーワードのみで表現することは難しく、

例示検索の考え方が有用であると考えられる。本研究では、映画検索時の情報要求のより正確なクエリ表現のために例示検索の考え方をを用いる。

3 提案手法

本研究では、検索クエリとしてキーワード（短い単語列）を用いるキーワード検索手法と、検索者の情報要求を表現する映画ランキングの例を用いる例示検索手法の2つを提案する。

提案する2つの手法に共通するアイデアをアルゴリズム1に示す。検索クエリ q が入力されると、アルゴリズムは、TTA-D2Vによるベクトルの \cos 類似度に基づいてそのクエリにマッチした映画のリスト $list$ を作成する。この $list$ を、学習済みのLtRモデルに入力として与えることで、 $list$ を適切な順序に並べ替えたリスト R が得られる。このリストを検索結果のランキングとする。

1つめの手法であるキーワード検索手法について第3.1節で、もう1つの手法である例示検索について第3.2節で、それぞれ詳細に説明する。

アルゴリズム1 映画ランキングを生成する

Input: q : 検索クエリ

Output: R : 映画ランキング (リスト形式)

```
function GENERATE_RANK( $q$ )
     $list$  := GET_MATCHED_MOVIES( $q$ )
     $R$  := RERANK( $list$ )
    return  $R$ 
end function
```

3.1 キーワード検索手法

キーワード検索手法では、観点ごとの映画の順位付けに関する多数の意見の集約を目的とする。この手法では、検索クエリとして1つ以上の単語を入力し、それをもとに映画のランキングを生成する。生成されたランキングは、入力されたクエリが表現する映画に対する観点に適した映画を、その観点に対する映画の順位付けに関する多数の意見を集約したものになると考えられる。この手法では、「旅行に行きたくなる映画」や「デートにぴったりな映画」など、自由なキーワードを用いて簡単に映画を検索することが可能となる。

手法の概要をフローチャート形式で図1に示す。手法は、準備フェーズと検索フェーズの2段階に分けられる。準備フェーズは最初に1回だけ実施する。検索フェーズは準備フェーズが完了した後におこなわれ、検索の度に繰り返しおこなわれる。

準備フェーズでは、主に3つのステップがある。多数の断片的なランキングから統合的な1つのモデルを学習するために、まず評判情報を用いて各映画を分散表現化する。Web上の映画レビューを用いてTTA-D2V [1]を学習し、それにより各映画の分散表現を得る。具体的には、1本の映画に対して投稿されたすべてのレビューに対して、個別のレビューIDではなく映画IDをラベルとして付与してDoc2Vecを学習する。

つぎに、映画ランキングデータセットからLtRの学習のための学習データを生成する。LtRの学習データは、クエリとアイテムのペアに対して関連度がラベル付けされたものが必要となる。本手法では、人々が自由に作成した映画ランキングを学習

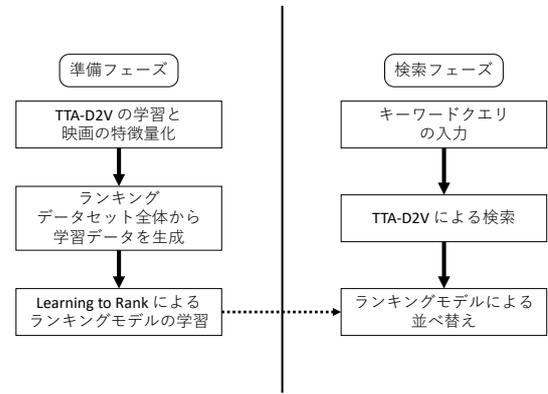


図1 キーワード検索手法のフローチャート

に用いることによって、様々な観点による映画の順位付けに関する多数の意見を集約する。ここでは、各映画ランキングに対して付けられたタイトルを、その中の映画ランキングがどのような観点で作成されたものであるかを表す「クエリ」とみるとみなす。

このような手法が実現可能になる条件として、さまざまな観点に沿ったランキングが学習に十分な量存在している必要がある点に注意が必要である。本研究における評価実験では、Yahoo!映画の「まとめ」機能からデータを取得し利用した。「まとめ」機能は、Yahoo!映画のユーザが各自自由にランキングタイトルを決め、その中で1位から最大10位まで映画をランキングすることが可能な機能である。

データの特徴量化は、各クエリと映画のペアごとにおこなわれる。使用した特徴量を表1に示す。映画のメタデータは、映画レビューサイトにおけるその映画の平均評点や、その映画を観たいと登録しているユーザの数、その映画のジャンルタグなどからなる23次元のベクトルである。映画の分散表現とクエリの分散表現はTTA-D2Vによって得られる、各300次元のベクトルである。その映画とクエリのそれぞれの分散表現の \cos 類似度も特徴量として用いる。クエリを分散表現化して扱うことにより、学習データに含まれないクエリが検索時に入力された場合においても対応できるようになると考えられる。

表1 クエリと映画のペアの特徴量

特徴量	次元数
映画のメタデータ	23
映画の分散表現	300
クエリの分散表現	300
映画とクエリの \cos 類似度	1
合計	624

そして、用意した学習データをLtR手法の1つであるLambdaMART [14]により学習し、ランキングモデルを得る。学習時には、学習データの各映画ランキングごとに5件の負例を与えて学習させる。負例の選び方は以下の通りである。ランキングのタイトルをクエリとみなし、適切な形態素解析および前処理を施したうえで前ステップで学習したTTA-D2Vモデルによって分散表現化する。そして、そのクエリとの \cos 類似度が最も低い映画を5件取得し、これをそのランキングに対する負例とする。ここまでで準備フェーズが完了し、複数の観点に

沿ったランキングを集約したモデルが完成する。

検索フェーズでは、検索者から任意のクエリを受け付け、そのクエリが表す観点の度合いの深さ順の映画ランキングを生成する。そのための手順は次のとおりである。まず、クエリの入力を受け付ける。本手法では、クエリの形式は1つ以上の単語からなるキーワードクエリである。入力されたキーワードクエリを、適切な形態素解析および前処理を施したうえで準備フェーズにて学習した TTA-D2V モデルによって分散表現化する。また、前処理によるクエリ整形として、クエリ中の「な映画」や「な作品」などの部分は除去する。

つぎに、準備フェーズで学習した TTA-D2V モデルを用いて、そのクエリが表す観点到した評判を得ている映画を検索する。学習したすべての映画は分散表現化されており、クエリもそれと同一のモデルで分散表現化されているため、クエリと映画の分散表現同士の \cos 類似度を計算し、上位 300 件程度を取得する。

最後に、準備フェーズにて学習したランキングモデルで、それらを並べ替えさせる。これを検索結果として提示する。この検索結果は、入力されたクエリが表す観点对する映画の順位付けに関する多数の意見を集約したランキングであると考えられる。

3.2 例示検索手法

例示検索手法では、情報要求のより正確なクエリ表現の実現を目的とする。この手法では、検索クエリとして検索者の情報要求を表現する映画ランキングの例を入力し、それをもとに映画のランキングを生成する。この手法では、キーワードでは表現が難しい検索者の情報要求をランキング例という形で表現させることで、多数の意見の集約としての映画ランキングではなく、より検索者個人の情報要求に適した映画ランキングシステムを実現可能になると考えられる。

手法の概要をフローチャート形式として図 2 に示す。例示検索手法も、準備フェーズと検索フェーズの 2 段階に分けられる。

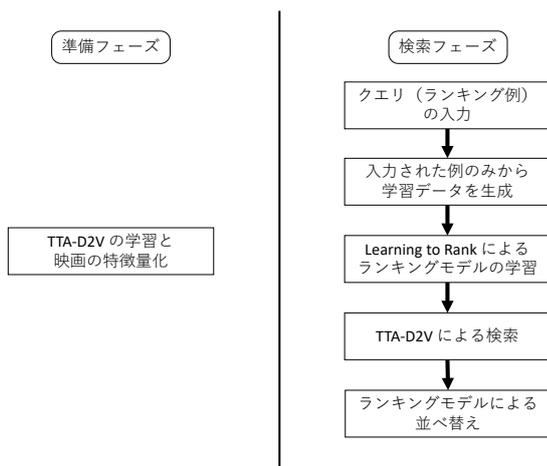


図 2 例示検索手法のフローチャート

準備フェーズは最初に 1 回だけ実施する。検索フェーズは準備フェーズが完了した後におこなわれ、検索の度に繰り返しおこなわれる。ただし、キーワード検索手法では準備フェーズで

おこなっている作業の一部が、本手法では検索フェーズに移動する。これは、本手法が目指す映画ランキングが多数の意見を集約ではなく検索者個人の情報要求により合わせることでることによる。本手法では検索者が入力するランキング例ただ 1 つのみを LtR 手法で学習し、映画をランキングする。そのため、検索要求を受け付けるたびに LtR の学習が必要となる。

準備フェーズはただ 1 つのステップからなる。検索者が入力するランキング例を学習するために、まず評判情報を用いて各映画を分散表現化する。TTA-D2V の学習と各映画の分散表現化の方法は、第 3.1 で述べたキーワード検索手法における手順と同様である。

準備フェーズが完了すれば、検索要求を受け付けることができる。検索要求を受け付けるたびに、次の 5 つのステップを実行し検索結果のランキングを生成する。

まず、クエリを入力を受け付ける。本手法では、クエリとして検索者の情報要求を表現する映画ランキングの例を受け付ける。そのランキングは「自分が懐かしいと感じる映画ランキング」や「見ていてお腹が空いた映画ランキング」など、ある程度統一された観点においてランキングされたものである必要があり、10 件程度の映画で構成されることが望ましい。

つぎに、LtR の学習のための学習データを生成する。キーワード検索手法と異なり、本手法では検索者から入力された映画ランキング例のみを学習データとする。これにより、学習結果として得られるランキングモデルが、検索者の入力したランキングに依存するモデルとなる。これを用いることにより、検索者の情報要求により適したランキングを生成する。

データの特徴量化は、映画ごとにおこなわれる。使用した特徴量を表 2 に示す。映画のメタデータと分散表現はキーワード検索手法と同一のものを用いる。本手法ではこの入力されたランキング自体が 1 つのクエリであり、また 2 つ以上のクエリに対するランキングを学習しないため、クエリの特徴量は学習に用いない。

生成した学習データを LambdaMART により学習し、ランキングモデルを得る。学習時には、学習データの各映画ランキングごとに 5 件の負例を与えて学習させる。負例の選び方は以下の通りである。入力されたランキング例の全映画の TTA-D2V による分散表現の重心を計算しベクトルを得る。そして、準備フェーズにて学習した TTA-D2V モデルを用いて、そのベクトルとの \cos 類似度が最も低い映画を 5 件取得し、これらをそのランキングに対する負例とする。

特徴量	次元数
映画のメタデータ	23
映画の分散表現	300
合計	323

つぎに、準備フェーズにて学習した TTA-D2V モデルを用いて、クエリが表す観点到した映画を検索する。まず、入力されたランキング例のそれぞれの映画について、分散表現を用いて \cos 類似度の高い映画を同じ件数ずつ取得する。このとき取

得する件数は、ランキングモデルで並び替えたい件数を基準に求める。例えば、ランキングモデルで 300 件を並べ替えてランキングを作成するとき、クエリが映画 10 件からなるランキングであった場合、それぞれの映画と \cos 類似度の高い映画を 30 件ずつ取得する。ただし、このとき取得した映画には重複がある場合がある。その時は適宜各映画との \cos 類似度上位 31 位以降から補充する。

最後に、入力されたランキング例から学習したランキングモデルに、取得した映画を入力してそれらを並べ替えさせる。それによって得られた映画ランキングは、入力されたランキング例のもつ特徴を強く反映したランキングであると考えられ、検索者個人の情報要求により適合していると考えられる。こうして得られた映画ランキングのうち上位を検索結果として提示する。

4 キーワード検索手法の評価実験

提案手法のうちキーワード検索手法のランキングアルゴリズムとしての精度を定量的に検証するため、実データを用いたクラウドソーシングによる大規模評価実験を実施した。

キーワード検索手法の有用性を評価するために、次の 2 つの手法による検索結果を比較する。

LtR の手法は、節 3.1 で述べた提案手法である。本実験では TTA-D2V で 300 件の検索結果候補を取得し、LtR モデルで並び替える。

TTA-D2V の手法は、TTA-D2V のベクトル同士の \cos 類似度を用いる手法である。クエリのベクトルとの \cos 類似度が高い順に映画を 300 件ランキングする。これは、提案手法から LtR モデルによる並び替えステップを除いたものである。

この 2 手法を用いて、用意した 10 個のクエリで映画を検索し、クエリごとに 2 種類の映画ランキングを得る。得られたランキングに登場する各映画について、アンケートによりクエリとの適合度を評価する。アンケート結果を用いて提案手法および比較手法のランキングアルゴリズムとしての精度を検証する。

アンケートは、大量のタスクを多数のワーカに実施させることが可能なクラウドソーシングプラットフォームである、Yahoo!クラウドソーシング上で実施した。

4.1 映画ランキングデータの学習

Yahoo!映画には、ユーザが自由なテーマで映画ランキングを作成可能である「まとめ」機能がある。Yahoo!映画のユーザは自由なタイトルを設定した上で最大で映画 10 件からなる映画ランキングを任意の個数作成できる。

本研究ではこのまとめデータを、映画の順位付けに関する多人数の意見が蓄積されたものであると捉え、LtR の学習データとして用いた。学習データとして用いる上で、まとめのタイトルをクエリ、まとめの中身をそのタイトル（クエリ）に対する映画の順位付けに関するユーザの意見であると考えた。

Yahoo!映画より研究用に収集したまとめデータのうち、まとめのタイトルに「ランキング」や「TOP10」など、そのまとめの中身がランキングであることが分かる表現を含むものを LtR の学習データとして用いた。そのようなまとめは 5,375 件

あった。

学習のために、使用するまとめに含まれる各映画から節 3.1 で述べた通り計 624 次元の特徴量を得た。

まとめ内の各映画は先述の通りに特徴量化される。それらに対してランキング ID とラベルが付与される。ランキング ID は 1 つのまとめに対して一意に割り当てられる。ラベルはまとめ内の順位が高い順に 10, 9, 8, ..., 1 と割り当てられる。まとめ内の映画の本数が 10 本に満たない場合でも 10 から降順に割り当てられる。負例にはラベルとして 0 を割り当てる。

このように特徴量化された映画ランキングデータを、Learning to Rank 手法で学習した。用いた手法は LambdaMART [14] で、実装は XGBoost [15] によるものである。

4.2 アンケート条件設定

クラウドソーシングによるアンケートは次のように実施した。

1 つの設問において、1 つのクエリとそのクエリで検索した結果得られた映画のうち 1 本のタイトルを被験者に提示し、そのクエリと映画の適合度を 4 段階で回答させた。被験者によっては、提示された映画を知らない場合が考えられるため、被験者は Web 上でその映画の情報（特に実際に観た人が書いたレビュー）を調べた上で回答することとした。

回答基準は、クエリが映画に適合しない場合は 0 点、少し適合している場合は 1 点、おおよそ適合している場合は 2 点、全体的かつ完璧に適合していると思う場合は 3 点である。

設問は、10 個のクエリそれぞれにつき比較手法である TTA-D2V によって 320 本の映画を取得し、合計 3,200 問用意した。取得した 320 本の内訳は、クエリのベクトルとの \cos 類似度上位 300 件全てと、301 位から 500 位までの 200 件からランダムに取得した 20 件である。提案手法は TTA-D2V による上位 300 件の映画を LtR モデルにより並び替えて提示するため、これで 3 手法の比較が可能である。

アンケートタスクでは、それぞれの設問に対して異なる 6 人の回答を収集した。設問は用意した 3,200 問からランダムに出題され、1 タスクあたり 5 問（いたずらなどを検出するためのチェック設問を加えて 6 問）、実施タスクの上限は 1 人の回答者につき 5 タスク（25 問）とした。

5 キーワード検索手法の評価実験結果

クラウドソーシングによる実験では、用意した設問 3,200 問に対して各 6 回答、計 19,200 件の回答が得られた。総回答者数は 1,024 人であった。全ての回答の自信度の平均は 1.93 であった。この回答データをもとに、各手法における 10 クエリ平均の適合率と nDCG を計算したものを表 3 に示す。本評価における適合率は、各手法によって得られた検索結果の映画のうち、クエリに適合した映画であると被験者が判定したものの割合である。今回の実験では、不適合な映画は「違う (0 点)」を選択するように被験者に指示している。そのため、本評価では各映画に対する 6 人の回答の平均が 1 点以上であった映画を適合映画とする。

表 3 より、提案手法の適合率は比較手法である TTA-D2V を平均的に下回ることが分かった。また、nDCG においても提案

表 3 各手法の適合率と nDCG (10 クエリ平均)

	LtR	TTA-D2V
1 位~30 位	0.718	0.780
順位区間別 31 位~100 位	0.679	0.737
適合率 101 位~300 位	0.667	0.637
301 位~500 位	-	0.654
nDCG @30	0.177	0.195
@100	0.386	0.425
@300	0.844	0.860

手法は比較手法を下回った。ただし、D2V 手法でも nDCG@30 で 0.195 と、完璧に適合度の高い順に提示できた場合の値である 1.0 と比べ非常に低い nDCG を示した。これは、提案手法のみならず、比較手法においても、ランキングの精度は低いことがわかる。

クエリ別に nDCG を計算した結果を表 4 に示す。「懐かしい」と「音楽が良い」の 2 つのクエリでは提案手法が比較手法を上回った。

表 4 各手法のクエリごとの nDCG@k

クエリ	nDCG@30		nDCG@100		nDCG@300	
	LtR	TTA-D2V	LtR	TTA-D2V	LtR	TTA-D2V
懐かしい	0.212	0.177	0.441	0.398	0.870	0.844
スカッと爽快感アクション	0.189	0.190	0.409	0.431	0.862	0.870
旅行に行きたくなる	0.181	0.220	0.390	0.456	0.825	0.867
泣けて前向きになれる	0.174	0.175	0.386	0.391	0.878	0.883
お腹がすく	0.099	0.230	0.274	0.455	0.653	0.744
音楽が良い	0.227	0.167	0.453	0.377	0.900	0.852
恋人と一緒に観たい	0.190	0.191	0.411	0.415	0.880	0.880
家族愛のあたたかさを感じる	0.172	0.183	0.381	0.416	0.868	0.882
血の気が引くような恐怖でゾクゾクする	0.155	0.191	0.362	0.437	0.866	0.894
一緒に推理するのが楽しい	0.168	0.230	0.354	0.470	0.841	0.881
10 クエリ平均	0.177	0.195	0.386	0.425	0.844	0.860

6 例示検索手法の評価実験

提案手法のうち例示検索手法の有用性を、検索者の具体的な情報要求およびユーザ体験に基づいて定性的に検証するため、被験者実験を実施した。

本実験では、実際に情報要求を映画ランキングの形式で入力して映画を検索可能な実験用 Web サイトを構築した (図 3)。検索者は、サイト上でクエリを自由なランキング形式で入力することができる。検索結果表示部分では、検索者が入力したクエリを用いて例示検索手法により映画を検索した結果のランキング上位 30 件を表示する。クエリにも含まれていた映画は灰色の網掛けをして表示した。

被験者実験は次のように実施した。被験者には、実験の意図および実験用 Web サイトの使い方について説明し、1 時間程度を目安にサイト上で自由に映画を検索させた。検索結果に被験者の知らない映画が登場した場合、その映画について Web で調べさせた。そして、被験者には実験の最後にアンケートに回答させた。アンケートでは、検索結果の精度や意外性、自らの要求をランキング形式で表現することの意義、および通常の Web 検索と例示検索手法とのユーザ体験の差異などについて、被験者の意見・感想を求めた。



図 3 実験用サイト

7 例示検索手法の評価実験結果

本節では、実験用 Web サイトを用いた例示検索手法の被験者実験の結果について説明する。本実験の被験者は 5 人で、年代は 20 代および 30 代、全員が男性であった。

7.1 検索結果の精度や意外性

まず、例示検索手法の精度についてどう感じるかを被験者に 4 段階で回答させた結果を、表 5 に示す。

表 5 “例示検索手法の精度についてどう感じますか?” への回答

選択肢	回答人数
良い	2
少し良い	1
少し悪い	1
悪い	1

2 人が「良い」と回答し、1 人が「少し良い」、1 人が「少し悪い」、1 人が「悪い」と回答した。「良い」と回答した被験者 A は、その理由を次のように回答した:「選んだ映画のジャンルだけでなく内容や監督、俳優にも関連のある結果が出たから。」

また、「悪い」と回答した被験者 E は、その理由を次のように回答した:「希望のタイトルがあまり出てこなかったから。」

次に、例示検索による検索結果の中で、自力では発見できない映画 (すなわち、意外だが良い映画) や、提示されてはじめて興味を持った映画がどのくらいあったかを被験者に 3 段階で回答させた結果を、表 6 に示す。

表 6 “例示検索の結果の中で、自力では発見できない映画や、提示されてはじめて興味を持った映画はありましたか?” への回答

選択肢	回答人数
たくさんあった	2
少しあった	3
全くなかった	0

2 人が「たくさんあった」と回答し、3 人が「少しあった」と回答し、「全くなかった」と回答した被験者はいなかった。また、

そのような検索結果の具体例について、被験者 A は次のように挙げた: 「スポーツ物で駅伝がテーマの「風が強く吹いている」を入力したところ「奈緒子¹」が出てきた。存在は知っていたが忘れていた。」

これらの回答から、例示検索手法は精度は高くなくとも、入力されたランキングの意図にある程度沿った検索結果を生成し、また被験者に新たな発見を与えることができる手法であることがわかった。

7.2 ランキングによる情報要求の表現

まず、ランキング形式で情報要求を表現するという検索手法について、キーワードによる情報要求の表現と比べてどう感じるか、被験者に 2 択で回答させた結果を表 7 に示す。

表 7 “例示検索手法の「自分のニーズの表現のしやすさ」をどう思いますか?” への回答

選択肢	回答人数
キーワード検索でも十分だ	0
例を挙げることでしか表現できないニーズがある	5

被験者 5 名全員が「例を挙げることでしか表現できないニーズがある」と回答した。

次に、例示検索手法のクエリ作成の手間や難しさについてどう思うか、複数回答可として被験者に回答させた結果を、表 8 に示す。

表 8 “例示検索手法の「検索の手間・難しさ」についてどう思いますか?” への回答

選択肢	回答人数
特に問題ない	1
入力に手間がかかる	2
入力する映画自体が思いつかない	3

1 人が「特に問題ない」と回答したが、2 人が「入力に手間がかかる」と回答し、3 人が「入力する映画自体が思いつかない」と回答した。うち 1 人は「手間がかかる」と「思いつかない」の 2 つの選択肢を両方選択した。

このことから、映画検索に例示検索を導入することによって、検索者は自らの情報要求をより正確に表現できるといえる。ただし、クエリとしてランキングを作成するという作業は、検索者にとって負担となることがわかった。

7.3 例示検索手法のユーザ体験

まず、例示検索手法が通常の Web 検索と比べて、興味をそそられる自分の知らない映画を探しやすいと思うか、被験者に 4 択で回答させた結果を表 9 に示す。

表 9 “例示検索は、通常の Web 検索と比べて「興味をそそられる、自分の知らない映画」を探しやすいと思いますか?” への回答

選択肢	回答人数
思う	2
少し思う	2
あまり思わない	1
思わない	0

2 人が「思う」、2 人が「少し思う」、1 人が「あまり思わない」と回答した。「思わない」と回答した被験者はいなかった。この質問に「思う」と回答した被験者 C は、その理由を次のように回答した: 「ランキング形式によって、普通のキーワード検索よりも自身に適合したオーダーメイドのようなセレクトができていていると感じるから。」

また、この質問に「あまり思わない」と回答した被験者 E は、その理由を次のように回答した: 「ランキングに含んだ作品が悪かったこともあるが、既知の作品が多かったり、意図とずれていることが多かった。」

次に、映画を検索するにあたって、結果がランキング形式で得られる必要があると思うかについて、被験者に 2 択で回答させた結果を表 10 に示す。

表 10 “検索結果がランキング形式である必要はあると思いますか?” への回答

選択肢	回答人数
思う	2
順不同でも問題ない	3

2 人が「思う」、3 人が「順不同でも問題ない」と回答した。

これらの回答から、例示検索手法はによる検索結果は、検索者にとって自己の嗜好を反映していると感じられるものであることがわかった。しかしながら、入力したランキングによっては検索者の意図を掴むことができず、検索者の意図とずれた結果を生成する場面があることもわかった。また、結果が順不同なりリストであっても検索者のユーザ体験としては問題がない可能性があることがわかった。

8 考 察

キーワード検索手法の評価の結果、提案手法であるキーワード検索手法は、比較手法である TTA-D2V によるキーワード検索に比べて平均的に低い nDCG を示した。また、上位 30 件の nDCG は 0.177 と非常に低い値であり、クエリとの適合度順に並んでいるとはいえない難しい結果となった。

このことから、第 1 章で挙げた本研究の 2 つめの課題である、各映画の順位付けに関する多人数の意見を集約して映画ランキングを生成するという目的は、提案手法では達成できなかった。

本研究で用いたデータセットは Yahoo!映画のユーザが自由に作成可能な個人的なまとめである。データセットを改めて確認したところ、タイトルに「ランキング」などの文言が含まれていても、中身がタイトルの観点の度合い順に映画が並んでいないように見受けられるものが存在した。さらに、映画への個人の評判や印象というのは個人的思い入れなどに特に強く左右されるものと考えられる。もしユーザが自身の判断においてタイトルの観点の度合い順に映画を並べていたとしても、それは非常に個人的要素の強い順序であると考えられる。そのため、その個人的順序付けを大量に集め学習したとき、ランキングモデルは学習データの中からクエリごとの普遍的特徴をよく発見できなかった可能性が考えられる。

第 7 章では、例示検索手法の被験者実験の結果を説明した。アンケートの回答では、精度については賛否両論あったものの、

1: 著者注: こちらも駅伝をテーマとする青春映画である。

全ての被験者が自らの入力したランキングに基づいて少しまたはたくさんの新たな映画を発見できており、また、映画検索に例示検索を取り入れることに意義があるとも回答した。さらに、「オーダーメイド感覚がある」という表現で例示検索手法を評価した被験者がいた。これは例示検索手法によってより正確なクエリ表現が入力可能で、しかもクエリに込められた意図を汲んだ検索結果が得られると、検索者が感じられるものであることを示している。

しかし、表 8 のとおり、例示検索手法ではランキングを作成する作業に手間や難しさを感じる被験者が多かった。ある被験者からは、「映画は 1 本観るのに 2 時間ほどかかるため、失敗が避けられるならクエリ作成に手間がかかっても許容できる。」という意見もあった。ただし、そもそもランキングを作成可能なほど映画を視聴した経験のない検索者にとっては、例示検索手法は使用が困難である可能性がある。このことは例示検索手法の大きな欠点であると考えられる。

また、第 7.3 節から、検索結果が順不同のリスト形式でもシステムとしては問題がない可能性が指摘された。その理由は、検索者が映画検索にシステムに「完璧な適合度順」を求めておらず、「新たな発見ができるかどうか」を重視しているからではないかと考えられる。そしてこのことから、例示検索は有用な映画検索システム実現のために重要な要素であると考えられる。

これらのことから、第 1 章で挙げた本研究の 3 つめの課題である、情報要求のより正確なクエリ表現を可能にするという目的は、提案手法によって達成された。提案手法はランキングをクエリとすることにより検索者がより正確に情報要求を表現可能であり、それに基づいて結果を提示することで、検索者の意図にあった新たな映画の発見を提供することができる手法であると考えられる。

9 おわりに

本研究は、検索者の自由な情報要求と各映画の評判情報に基づく新たな映画検索手法を実現するために、観点ごとの映画の順位付けに関する意見の集約、および情報要求のより正確なクエリ表現の実現に取り組み、2 つの手法を提案した。キーワード検索手法では、観点ごと映画の順序付けに関する多人数の意見を学習したランキングモデルによって検索結果の候補を並べ替える。これによって入力したキーワードの観点との関連度合いが高い順に検索結果が得られることを期待した。例示検索手法では、検索者が入力したただ 1 つのランキングを使って学習したランキングモデルによって検索結果の候補を並べ替える。これによって入力したランキングの特徴を強く反映し、検索者の要求に合致した検索結果が得られることを期待した。

評価実験の結果、キーワード検索手法は比較手法に比べて低い nDCG を示し、有用性を確認することができなかった。一方、例示検索手法はおおむね高評価であり、例示検索手法はランキングクエリに基づいて結果を提供することで、検索者の意図にあった新たな映画の発見を提供することができる手法であることがわかった。また、例示検索は有用な映画検索システム実現のために重要な要素である可能性が示唆された。

今後の課題としては、例示検索のユーザ体験の更なる向上が挙げられる。本研究では、検索者の情報要求を把握するために映画ランキングを入力させたが、これは検索者にとって負担が大きいことが分かった。そこで、適合性フィードバックなどを用いてインタラクティブなクエリ作成を導入するなどして、検索者の負担を軽減することがより良い映画検索システムを実現するために必要である。

謝 辞

本研究は JSPS 科研費 18K18161 (代表: 莊司慶行), 18H03243 (代表: 田中克己) 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] K. Kurihara, Y. Shoji, S. Fujita, and M. J. Dürst. Target-topic aware doc2vec for short sentence retrieval from user generated content. In *Proc. of iiWAS 2019*, pp. 463–467, 2019.
- [2] H. Li. A short introduction to learning to rank. *IEICE Trans. Inf. Syst.*, Vol. 94-D, pp. 1854–1862, 2011.
- [3] O. Barkan and N. Koenigstein. Item2vec: neural item embedding for collaborative filtering. In *MLSP 2016*, pp. 1–6. IEEE, 2016.
- [4] V. Phi, L. Chen, and Y. Hirate. Distributed representation based recommender systems in e-commerce. In *Proc. of DEIM 2016*, pp. C8–1, 2016.
- [5] S. K. Karmaker Santu, P. Sondhi, and C. Zhai. On application of learning to rank for e-commerce search. In *Proc. of SIGIR 2017*, pp. 475–484, 2017.
- [6] M. Ludewig and D. Jannach. Learning to rank hotels for search and recommendation from session-based interaction logs and meta data. In *Proc. of the RecSys Challenge 2019*, pp. 1–5, 2019.
- [7] J. Xu, C. Chen, G. Xu, H. Li, and E. R. T. Abib. Improving quality of training data for learning to rank using click-through data. In *Proc. of WSDM 2010*, pp. 171–180, 2010.
- [8] L. Wu, D. Hu, L. Hong, and H. Liu. Turning clicks into purchases: Revenue optimization for product search in e-commerce. In *Proc. of SIGIR 2018*, pp. 365–374, 2018.
- [9] 前島一弥, 横川智浩, 吉田稔, 山田剛一, 絹川博之, 中川裕志. Web上の表情報の例示検索における機械学習手法の検討. 人工知能学会全国大会論文集 第 22 回全国大会 (2008), pp. 28–28. 一般社団法人人工知能学会, 2008.
- [10] N. Kosugi, Y. Nishihara, T. Sakata, M. Yamamuro, and K. Kushima. A practical query-by-humming system for a large music database. In *Proc. of ACM MM 2000*, pp. 333–342, 2000.
- [11] J. Salamon, J. Serra, and E. Gómez. Tonal representations for music retrieval: from version identification to query-by-humming. *IJMIR*, Vol. 2, No. 1, pp. 45–58, 2013.
- [12] M. Rocamora, P. Cancela, and A. Pardo. Query by humming: Automatically building the database from music recordings. *Pattern Recognition Letters*, Vol. 36, pp. 272–280, 2014.
- [13] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith. Query by humming: Musical information retrieval in an audio database. In *Proc. of ACM MM 1995*, pp. 231–236, 1995.
- [14] C. J. C. Burges. From ranknet to lambdarank to lambdamart: An overview. Microsoft Research Technical Report MSR-TR-2010-82, June 2010.
- [15] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proc. of SIGKDD 2016*, KDD '16, pp. 785–794, New York, NY, USA, 2016. ACM.