

書籍自動整列を用いた整列順序に基づく書籍列推薦手法

宮本 達矢[†] 北山 大輔^{††}

[†] 工学院大学大学院工学研究科 〒163-8677 東京都新宿区西新宿 1-24-2

^{††} 工学院大学情報学部 〒163-8677 東京都新宿区西新宿 1-24-2

E-mail: [†]em18014@ns.kogakuin.ac.jp, ^{††}kitayama@cc.kogakuin.ac.jp

あらまし 近年、推薦システムは単純な精度のみによる評価ではなく、novelty や diversity, explainability, coverage, serendipity といった様々な観点からの評価が行われている。これは、精度だけでは多様性や新規性、意外性などの要素が失われてしまい、ユーザに対し十二分な満足感を与えられないと判明したためである。我々は、先行研究において explainability と serendipity を考慮した書籍列推薦手法の提案を行った。本論文では、本システムのユーザビリティをより高めるため、ユーザの入力を書籍列ではなく書籍集合とするための、自動書籍整列手法を提案する。これによりユーザ自身は整列意図を考慮することなく、より自由に本システムでの推薦が可能になると考えられる。

キーワード 推薦システム, 書籍推薦, 分散表現, 回帰

1 はじめに

近年、推薦システムは単純な精度のみによる評価ではなく、novelty や diversity, explainability, coverage, serendipity といった様々な観点からの評価が行われている [1-3]。これは、精度だけでは多様性や新規性、意外性などの要素が失われてしまい、ユーザに対し十二分な満足感を与えられないと判明したためである。

この問題を解決するため、我々は先行研究 [4] において、ユーザの書籍整列順序に基づく書籍予測モデルの提案と評価を行った。本システムでは、ユーザが何らかの意図に基づき整列された書籍列を入力とした際、システムは書籍列の整列意図を予測する。その後、入力とした書籍列の持つ傾向に沿って並んだ書籍列にふさわしい書籍を予測し、予測した推薦書籍を挿入した書籍列をユーザに推薦する。ここで挙げる整列意図とは、「コメディ要素の強い順」や「ミステリの難しい順」といった基準や、「書籍の読みやすい順」に「雰囲気の良い順」といったユーザの感覚に基づく順序などである。書籍列はユーザの整列意図に沿っているため、ユーザは推薦書籍の位置により、なぜその書籍が推薦されたのか推薦理由を理解しやすいと考えられる。また、推薦システムがユーザの意図していなかった整列順序の法則を発見することで、少し意外な書籍を推薦することができる可能性が存在する。以上のことから、本推薦システムが高い explainability と serendipity を有していることを期待した。

本論文では、本システムのユーザビリティをより高めるため、ユーザの入力を書籍列ではなく書籍集合とするための、自動書籍整列手法を提案する。ユーザが書籍集合を入力し、システムが自動的に何らかの意味に沿った順序に書籍を並べ替え、これを推薦元となる書籍列として書籍の予測を行う。

本論文の概要を図 1 に示す。はじめに、レビューから書籍特徴ベクトルの作成を行う。次に、入力された書籍集合から書籍列の作成を行い、書籍列での位置ごとに特徴ベクトルの予測を

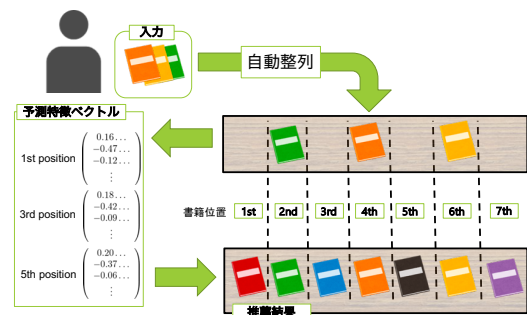


図 1 書籍自動整列を用いた書籍推薦の概念図

行う。最後に、書籍列の各位置において最も類似度の高い推薦書籍を書籍列に挿入し、作成した書籍列を推薦する。

本論文の構成は以下の通りである。2 節では関連研究について述べる。3 節では自動整列を用いた整列順序に基づく書籍の予測について述べる。4 節では自動整列の精度評価を行う。5 節では自動整列のユーザビリティ評価を行う。6 節では本論文のまとめと今後の課題について述べる。

2 関連研究

書籍列中の書籍の予測に類似する研究として、プレイリスト推薦が挙げられる。池田ら [5] は、楽曲の音響特徴に注目し、滑らかに楽曲が遷移するプレイリストの推薦を行った。楽曲の音響特徴から作成した特徴ベクトルを、2次元空間へと写像することで、プレイリストに含まれる楽曲の探索を行なっている。我々は、作成した特徴ベクトルを異なる特徴空間へ写像せず、書籍を予測する点で異なっている。

ある観点から見たアイテム同士の順序関係に注目した研究として、旭ら [6] の Web 上に存在する比較文を用いた補完エンティティの発見が挙げられる。ユーザが入力した 2 つのエンティティから、エンティティを含む比較文を収集することでエンティティ同士の関連性を明らかにする。その後、入力された

2つのエンティティを特定の観点から補完するエンティティを発見する。我々は、入力としたアイテム同士を補完するだけでなく、より特定のアイテムの持つ特徴を強めたアイテムを発見することも目指している点で異なっている。

explainability に関連する研究として、Chang ら [7] のクラウドソーシングを用いた自然言語の推薦理由文の生成が挙げられる。Chang らは、あるトピックに関連する語を含むレビューをワーカーに補助として提示することで、専門家以外のワーカーが機械的に自然言語の推薦理由を作成できるようにした。我々は、書籍列中での位置によって推薦理由を説明することを目指しているが、よりユーザが理解しやすい推薦理由を提示するためには、自然言語による説明が必要であると考えられる。

serendipity に関する研究として、Chen ら [8] の serendipity が推薦において与える影響の調査が挙げられる。Chen らは serendipity と novelty や diversity といった指標との関係、またユーザ満足度や購入意欲に与える影響について調査した。実際のオンラインショッピングサイトでのアンケート調査を通じ、serendipity が実際にユーザ満足度や購入意欲に良い影響を与えていることを示している。

3 自動整列を用いた整列順序に基づく書籍の予測

ユーザが複数冊の書籍によって構成された書籍集合を与えたとき、書籍集合からシステムが何らかの意味を持つ順序に並んだ書籍列を作成する。その後、作成した書籍列に、書籍列の意味を持った順序関係に沿った推薦書籍を挿入し、作成した書籍列の推薦を行う。意味を持った順序として、「ミステリ要素の強くなる順」や「雰囲気は軽くなっていく順」などが考えられる。

本論文では、書籍列の自動整列を行うため、回帰による書籍列の評価または次元削減を行う。最も評価の高い書籍列または、低次元空間での整列順序を書籍列として採用し、書籍列に沿った推薦書籍を列中に挿入するため、書籍特徴ベクトルの予測を行う。

3.1 書籍特徴ベクトルの作成

書籍の特徴を予測するために、本研究ではレビューに注目した。ユーザが自由に投稿することのできるレビューには、書籍自体の内容に触れる文章や、書籍を読んで得た感想などが述べられている。よって、書籍レビューを用いることで、書籍の特徴を表すことができると考えた。レビューから書籍特徴ベクトルを作成するため、本研究では Paragraph vector [9] を利用する。

はじめに、レビューに対し形態素解析を行う。この際、単語を原型に戻し、レビュー中に含まれる人名を同一の記号に置き換えた。形態素解析には、辞書として mecab-ipadic-NEologd¹ を設定した MeCab [10] を利用した。

次に、書籍特徴ベクトルを作成する。ある書籍に対し投稿されたレビュー全てを合わせて1文書とし、推薦候補である全ての書籍から作られた文書の集合を全文書とする。この全

文書に対し、Doc2Vec を用いて書籍特徴ベクトルを作成する。Doc2Vec のパラメータはそれぞれ、feature vector size は 300、window size は 15、min-count は 5 とした。

3.2 書籍の自動整列

順序自動指定手法として、書籍特徴ベクトルの特徴空間をそのまま用いて推定する手法と、低次元空間への圧縮を行い位置を推定する2手法が考えられる。特徴空間をそのまま用いる手法として線形回帰を、低次元空間への圧縮を用いる手法として主成分分析を利用する。

3.2.1 線形回帰を用いた順序自動推定

書籍が何らかの意味を持つ順序に並ぶとき、書籍特徴ベクトルが特徴空間上で直線上に整列すると考えた。よって、与えられた書籍集合から作成できる各順列に並んだ書籍特徴ベクトルに対し線形回帰を行い、その決定係数のスコアによって書籍整列順序の評価を行う。説明変数として、書籍の列中での位置を与える。次に目的変数として書籍特徴ベクトルが当てはまるが、線形回帰では複数の要素を目的変数とすることは不可能である。このため、目的変数は書籍特徴ベクトルの各要素とし、要素ごとに線形回帰を行う。その後、全ての要素の決定係数を算出し、この平均のある順列のスコアとする。最後に、最もスコアの高い順列を推薦元の書籍列として採用する。

3.2.2 主成分分析を用いた順序自動推定

書籍特徴ベクトルを1次元の特徴空間に落とし込むことで、書籍が何らかの意味を持つ順序に並ぶと考えた。よって、与えられた書籍集合に含まれる書籍特徴ベクトルに対し主成分分析を行うことで、第1主成分によってのみ表される1次元ベクトルに圧縮する。その後、第1主成分の小さい順に入力書籍を並べ替え、この順を推薦元の書籍列として採用する。

3.3 特徴ベクトルの予測

推薦書籍を挿入した書籍列は式1のように表すことができる。このとき $book_i^{in}$ は i 冊目に出現する入力書籍を表し、 $book_i^{rec}$ は i 冊目に出現する推薦書籍である。

$$BookLine = [book_1^{rec}, \dots, book_n^{in}, book_{n+1}^{rec}, \dots, book_{n+m}^{in}, book_{n+m+1}^{rec}, \dots] \quad (1)$$

3.1節で作成した書籍特徴ベクトルから、回帰を用いて書籍列の i 番目の要素である推薦書籍 $book_i^{rec}$ の予測特徴ベクトルを作成する。

我々は先行研究 [4] より、書籍列中での位置 i および入力書籍 $book_i^{in}$ 同士の類似度によって、最適な予測手法が異なることを確認している。よって、位置 i の予測手法 $BookPredFunc_i$ は式2のように変化する。このとき $Book^{in}$ は、書籍列中に存在する入力書籍集合を示し、 α は入力書籍同士の類似度の閾値を示している。本論文では、先行研究にてカーネルリッジ回帰が最も高い精度を示した入力書籍類似度を基準とするため、 $\alpha = 0.233$ とした。 $BookSim$ は与えられた書籍集合の類似度を示し、書籍集合に含まれる書籍2冊の書籍特徴ベクトルの \cos 類似度の全組み合わせの平均を示す。

¹ : <https://github.com/neologd/mecab-ipadic-neologd>

$BookPredFunc_i =$

$$\begin{cases} \text{KernelRidge} & \text{if } BookSim(Book^{in}) < \alpha \\ & \text{if } BookSim(Book^{in}) \geq \alpha \text{ and} \\ \text{Linear} & \text{exist}(book_l) \text{ and } exist(book_r) \\ & (l < i < r) \\ \text{CCA} & \text{otherwise} \end{cases} \quad (2)$$

$$BookSim(Book) = \frac{\sum_{\{book_p, book_q \in Book\}} CosSim(book_p, book_q)}{|Book|C_2} \quad (3)$$

はじめに入力書籍同士の類似度を計算し、入力書籍類似度が閾値 α を下回っていた場合、カーネルリッジ回帰 (*KernelRidge*) による予測を行う。次に、位置 i の左右方向に入力書籍が存在する場合、線形回帰 (*Linear*) による予測を行う。最後に、位置 i の左右どちらか位置方向にのみ入力書籍が存在する場合、正準相関分析による予測を行う。各回帰手法の実装には、全て `scikit-learn`² を用いた。

3.3.1 線形・カーネルリッジ回帰による予測

(*Linear, KernelRidge*)

最小二乗法を用いた線形回帰およびガウスカーネルを用いたカーネルリッジ回帰によって、予測特徴ベクトルの作成を行う。説明変数として入力書籍の書籍列中での要素番号を表した整数 i を、説明変数として入力書籍の書籍特徴ベクトルを与える。しかし、線形およびカーネルリッジ回帰では複数の目的変数を対象に回帰を行うことはできないため、書籍特徴ベクトルの各要素ごとに回帰を行う。今回書籍特徴ベクトルは 300 次元であるため、300 個の回帰モデルを作成する。その後、作成した回帰モデル群を用いて、 i 番目の推薦書籍の予測特徴ベクトルを作成する。

3.3.2 正準相関分析による予測 (CCA)

正準相関分析 [11] は、複数の要素を持ったグループ同士の正準相関係数を求める分析である。正準相関分析のモデルは平均 0、分散 1 に標準化されたグループをそれぞれ X, Y とすると式 4 のように定式化 [12] され、重み行列 A, B と誤差行列 E を求め式変形をすることで、回帰式を求めることができる。

$$YB = XA + E \quad (4)$$

予測特徴ベクトルを求めるにあたり、書籍列に含まれる全ての入力書籍において、それぞれの位置を表す要素番号 i を X 。書籍特徴ベクトルの各要素からなる変数群を Y とし、正準相関分析を行う。回帰式を求めたのち、書籍列 i 番目の予測特徴ベクトルを回帰によって作成する。

3.4 推薦書籍の決定

書籍列要素のうち、入力書籍 $book^{in}$ が収まっていない i 番目の予測特徴ベクトルと、推薦候補書籍の書籍特徴ベクトルの \cos 類似度を算出する。その後、書籍列中で最も類似度の高い書籍を持つ i 番目の要素から順に推薦書籍を決定する。予測特

徴ベクトルと最も類似度の高い書籍を $book_i^{pred}$ の挿入書籍とする。

ただし、このときすでに書籍列中に同一著者の作品が存在している書籍はフィルタリングを行い、挿入を行わないようにする。これは、同一著者の作品は既にユーザが知っている、あるいは簡単に見つけることができ、推薦する価値が薄いと考えられるためである。

4 書籍の自動整列の精度評価

書籍の自動整列の精度評価を行うため、実験を行った。

4.1 実験データ

小説を対象として、実験を行った。小説およびレビューは読書メーター³から収集した。書籍特徴ベクトルを作成するために利用した書籍は 238,135 冊であり、これらの書籍に対し投稿されたレビューの総数は 10,102,455 件である。

4.2 実験方法

書籍の自動整列結果を評価するため、クラウドソーシングサービスであるクラウドワークス⁴を用いて正解データの作成を行った。はじめに、我々が 4.1 節にて収集した書籍のうち、投稿されたレビュー数が 2,000 件以上の書籍 443 冊から、ランダムに 100 件の書籍を選択した。次に、ワーカーがこの 100 冊から 5 冊を選択し、何らかの意図に基づく順序に並び替えた書籍列を作成する。このとき、「発売日順」や「ページ数順」といった書誌情報に基づく整列を禁止した。禁止した理由として、今回特徴ベクトルを作成するにあたり利用したレビュー中には書誌情報がほとんど含まれておらず、予測がほぼ不可能なためである。最後に、ワーカーが作成した書籍列から、5 冊中 3 冊の順序関係が他のワーカーと同じ順序関係、あるいは逆順の順序関係になった書籍列を正解データとした。これは、他のワーカーと一部が同一の順序関係を持つ書籍列の場合、ある程度一般的な観点から整列された書籍列であると考えられるためである。タスクには合計 104 人 115 件⁵の回答があり、最終的に 36 人 36 件の正解データを作成した。

3.2 節の 2 手法にて作成した書籍列と、正解データの書籍列のスピアマンの順位相関係数によって評価を行う。ただし、正解データと自動整列結果が完全な逆順である場合、書籍列としては同一であると見なすことができるため、スピアマンの順位相関係数の絶対値で評価を行う。また、平均実行時間についても比較を行う。

比較手法として、ランダム整列手法を用意した。ランダム整列手法では、入力書籍を完全にランダムな順序で並べ替えを行う。

4.3 結果

結果は表 1, 2 のようになった。各手法で正解データそれぞ

2 : <https://scikit-learn.org/stable/>

3 : <https://bookmeter.com/>

4 : <https://crowdworks.jp/>

5 : 書籍列にすべて異なる書籍を選択した場合に 2 回まで回答可能とした。

表 1 自動整列の手法ごとの結果

	スピアマンの順位相関係数	p 値
ランダム手法	0.40	0.53
線形回帰	0.58	0.35
主成分分析	0.57	0.35

表 2 自動整列の手法ごとの平均実行時間

	平均実行時間
線形回帰	21.7450 秒
主成分分析	0.0004 秒

れの書籍列を予測し、全ての相関係数の平均を取っている。また、ランダム整列手法では、各データで 10 回実行した平均値を示している。

相関係数が高かったのは線形回帰を用いた順序自動推定手法であり、相関係数は 0.58 と正解データに対し相関を示している。しかし、主成分分析を用いた順序自動推定手法においても 0.57 であり、相関係数には 0.01 の差しか生まれなかった。また、書籍列一つにつき 5 冊程度しか存在しないため、p 値は 0.35 と高くなってしまった。より正確に検証を行うためにはより長い書籍列を用いて表を行う必要があるが、書籍数が増加するにつれ正解データを作成することは難しくなると考えられる。

実行時間では、2 種類の順序自動推定手法の間で大きな差が生まれた。主成分分析を用いた順序自動推定手法が平均 0.1 秒以下で終わるのに対し、線形回帰を用いた順序自動推定手法では平均 21 秒も時間が掛かってしまう。このような差が生まれる原因として、主成分分析は冊数にかかわらずベクトル演算によって計算が終わる事に対し、線形回帰では組み合わせの回数分だけ回帰を繰り返すことが考えられる。実装上の問題ではあるが、ベクトル演算に比べ繰り返し計算は非常に時間がかかる。冊数が増加するにつれ、線形回帰と主成分分析の差はさらに広まる事が考えられる。

実験の結果、精度の差はあまり見られず、実行時間には大きな差が見られた。このことから、より実行時間の早い主成分分析を用いた順序自動推定手法が優れていると考えられる。

5 自動整列のユーザビリティ評価

自動整列を用いたことによるユーザビリティの変化を検証するため、実験を行った。書籍特徴ベクトルの作成には 4.1 節にて説明したものと同様のデータを用いた。しかし、推薦候補書籍はこのうちレビュー投稿数が 10 件以上である 94,552 冊とした。

5.1 推薦アプリケーション

被験者実験を行うにあたり、提案手法を実装した推薦アプリケーション「並べて PON！」および「オート PON！」を構築した。「並べて PON！」は入力書籍の自動整列を用いずに書籍列を推薦し、「オート PON！」は入力書籍の自動整列を用いて書籍列を推薦する。推薦アプリケーションのインターフェースは一部機能を除き同一であり、図 2 に示す。



図 2 推薦システムのインターフェース

入力書籍を検索する方法として、書籍タイトルまたは著者名による検索や、書籍データベース内の書籍のランダム検索、レビュー件数 1000 件以上の書籍に限定したランダム検索を用意した。検索した書籍を選択し、入力書籍セットボタンを押下することで、選択した書籍を入力書籍として設定する。「並べて PON！」では、入力書籍表示ウィンドウ上に表示されている書影をドラッグ&ドロップすることによって、入力書籍の順序を入れ替えることができる。「オート PON！」では、入力書籍は自動整列手法を用いて推薦時に内部的に並び替えが行われる。

3 冊または 4 冊の書籍を入力書籍に設定後、推薦ボタンを押下することで書籍列が推薦される。書籍列は入力書籍と推薦書籍によって構成され、入力書籍同士の間と列の両端にそれぞれ 3 冊ずつ推薦書籍が配置される。よって入力書籍が 3 冊の場合は 15 冊、入力書籍が 4 冊の場合は 19 冊の書籍によって書籍列は構成されている。

また、これらの推薦アプリケーションの説明として、チュートリアル動画^{6,7}を作成した。

5.2 実験方法

実験は「並べて PON！」と「オート PON！」で個別のタスクとして行った。被験者は 4 節と同様にクラウドワークスを用いて集め、「並べて PON！」では 18 人、「オート PON！」では 20 人の被験者を募集した。それぞれの被験者のおおよその読んだことのある小説の冊数は表 3 のようになった。

実験は、以下のような手順で実施した。

- (1) 被験者に「次の休みに読みたい小説の候補として 2 冊の小説を探す」タスクを提示。
- (2) 被験者がシステムのチュートリアル動画を視聴。
- (3) 被験者がシステムを用いて次の休みに読む書籍の候補を 2 冊探す。
- (4) 被験者がシステムに関する質問(表 4)に回答。

システムの評価項目として、各システムへの質問の他に、平均推薦回数および平均入力書籍数、平均入力書籍列類似度、平均推薦書籍列類似度について評価する。書籍列類似度は、書籍列に含まれる書籍 2 冊ずつの cos 類似度総当たりの平均を示している。

6 : <https://youtu.be/20bfM79m4Z0>(並べて PON！)

7 : <https://youtu.be/04P1tj0IpZQ>(オート PON！)

表 3 各システムの被験者のおおよその読んだことのある小説の冊数

	並べて PON!(人)	オート PON!(人)
0~50 冊	6	2
51~100 冊	3	3
101~500 冊	6	6
500 冊以上	3	3

表 4 システムに対する質問

	質問 (5 段階で回答)
Q1	推薦された書籍の並び方の基準はわかりやすかった
Q2	複数冊の小説を入力する労力に対して推薦された書籍が見合っていた

5.2.1 結果

結果は表 5 のようになった。「並び方の基準」および「労力に見合う」においては、値が 5 に近いほど基準がわかりやすく労力に見合う良い結果となっている。

並び方の基準がわかりやすかったのは自動整理を用いる「オート PON!」であり、より入力の手間と推薦結果の労力に見合っていたのは「並べて PON!」になった。自動整理手法による推薦の方がよりわかりやすかった理由として、手動整理において被験者の想定した並び順がシステムにとってわかりづらい基準だった可能性がある。これに対し自動整理ではシステム側がわかりやすい順序に並べ替えるため、被験者にとってもわかりやすい書籍列が作られた可能性がある。労力に関しては、被験者が想定した意図通りに並んだ書籍列が推薦されたことで、書籍を並べ替える苦労に見合った書籍を得ることができたと考えられる。

平均推薦回数および平均入力書籍数では、どちらも自動整理を用いる「オート PON!」が若干高い値を示した。自動整理では、入力の際に並び順を考えるという手間が省かれるため、比較的気軽に推薦を試しやすく、また入力書籍も増やしやすかったと考えられる。

書籍列類似度においても、入力および推薦共に自動整理を用いる「オート PON!」が若干高い値を示した。この理由として、自動整理では書籍の並び順を考えながら入力書籍を選ぶ必要がなく、被験者はより自由に様々な入力書籍を選択したことが考えられる。結果的に自動整理では多様性に富んだ推薦を行ったと考えられるが、これがユーザにとって好ましい推薦であるのか明らかにする必要がある。また、あまりに関連性のない入力書籍同士が与えられた場合、システムにとってもユーザにとっても理解しづらい書籍列になってしまう可能性がある。

これらの結果より、ユーザビリティという観点においては、推薦を繰り返しやすく入力書籍も自由に決めることができると考えられる自動整理の方が優れていると考えられる。また、ユーザ自身が整理意図を決めるより、システムが適切に推薦しやすい順序に並び替える方が、ユーザにとっても整理意図がわかりやすい推薦を行える可能性がある。

表 5 手動整理と自動整理の結果

	並べて PON!	オート PON!
並び方の基準 (Q1)	3.23	3.70
労力に見合う (Q2)	3.78	3.70
平均推薦回数	1.33 回	1.85 回
平均入力書籍数	3.54 冊	3.68 冊
平均入力書籍列類似度	0.30	0.26
平均推薦書籍列類似度	0.43	0.37

6 まとめ

本論文では、書籍集合から書籍列を作成するための、自動書籍整理手法を提案した。我々が先行研究にて作成した explainability と serendipity を考慮した書籍列推薦手法に提案手法を適用し、推薦システムの入力に書籍列ではなく書籍集合を用いることで、ユーザビリティが向上することを目的としている。

提案手法では、回帰による整理順序の評価による書籍列の作成、または次元削減による書籍の整理による書籍列の作成を行った。評価実験として、自動整理および手動整理手法を実装した推薦アプリケーションを作成し、ユーザビリティの比較を行った。

これらの結果、自動整理を用いた推薦アプリケーションの方がユーザにとって推薦を試しやすく、また推薦結果も理解しやすい可能性があることを確認した。しかし、ユーザが入力書籍を自由に選ぶことで、関連性の薄い書籍によって理解し難い書籍列が作られてしまう可能性がある。

今後の課題として、作成した書籍列の整理意図をタグや自然言語による説明を用いて明示的に表示することを考えている。書籍列の作成をシステム側が行うことでユーザの負担は減少すると考えられるが、同時にユーザ自身は入力した書籍集合からどのような意図で並んだ書籍列が作成されているのかわからなくなってしまう。推薦された書籍列から意図を読み取ることしかできないため、より整理意図の説明は重要であると考えられる。また、提案手法においてシステムが並べやすい書籍列の持つ意味について明らかにする必要がある。現在レビューを元に特徴ベクトルを作成していることから、書誌情報に基づく書籍列は作成することは難しいと考えられるが、これ以外にもシステムでは作成しづらい書籍列が存在する可能性がある。これを明らかにし、レビュー以外の情報も用いるようにするなどの提案手法の改善を行うことで、より適切な意味を持った順序に並ぶ書籍列を推薦できると考えられる。

謝辞 本研究の一部は、2019 年度科研費基盤研究 (B)(課題番号: 19H04118) によるものです。ここに記して謝意を表すものとします。

文 献

- [1] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, Vol. 22, No. 1, pp. 5–53, 2004.
- [2] Yehuda Koren. Tutorial on recent progress in collabora-

- tive filtering. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pp. 333–334. ACM, 2008.
- [3] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pp. 257–260. ACM, 2010.
- [4] 宮本達矢, 北山大輔. ユーザの書籍整列順序に基づく書籍予測モデルとその評価. 研究報告データベースシステム (DBS), Vol. 2019, No. 3, pp. 1–5, 2019.
- [5] 池田翔武, 奥健太, 川越恭二. 楽曲遷移の滑らかさを考慮したプレイリスト推薦. 情報処理学会論文誌データベース (TOD), Vol. 9, No. 4, pp. 1–5, 2016.
- [6] 旭直人, 山本岳洋, 中村聡史, 田中克己. 比較文集約に基づくエンティティ間の順序関係推定による補間エンティティの発見. 情報処理学会論文誌, Vol. 52, No. 12, pp. 3527–3541, 2011.
- [7] Shuo Chang, F. Maxwell Harper, and Loren Gilbert Terveen. Crowd-based personalized natural language explanations for recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pp. 175–182. ACM, 2016.
- [8] Li Chen, Yonghua Yang, Ningxia Wang, Keping Yang, and Quan Yuan. How serendipity improves user satisfaction with recommendations? a large-scale user evaluation. In *Proceedings of the The 2019 World Wide Web Conference*, WWW '19, pp. 240–250. ACM, 2019.
- [9] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 1188–1196, 2014.
- [10] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-2004)*, pp. 230–237, 2004.
- [11] 赤穂昭太郎. 正準相関分析入門. 日本神経回路学会誌, Vol. 20, No. 2, pp. 62–72, 2013.
- [12] Alan Julian Izenman. *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. Springer Publishing Company, Incorporated, 1 edition, 2008.