

ランキングの部分的な改善を目的とした能動学習手法

柴田 幸輝[†] 加藤 誠^{††}

[†] 筑波大学情報学群知識情報・図書館学類 〒 305-8550 茨城県つくば市春日 1-2

^{††} 筑波大学 図書館情報メディア系 / JST さきがけ 〒 305-8550 茨城県つくば市春日 1-2

E-mail: [†]s1811496@s.tsukuba.ac.jp, ^{††}mpkato@acm.org

あらまし 本論文では検索エンジン管理者からのフィードバックに基づいてランキングアルゴリズムの部分的な改善を可能にする手法を提案する。ここでいう部分的改善とは「特定のクエリで特定の文書の順位を理想の順位に変える行為」と定義される。提案手法では検索エンジン管理者の意図を推定するモデルを用意し、以下の1から3のステップを繰り返して検索エンジン管理者の意図を推定するモデルを学習させる。(1) 検索エンジン管理者の意図を推定するモデル側からフィードバックして欲しいクエリを提示, (2) 検索エンジン管理者は提示されたクエリに対してフィードバックを与える, (3) 検索エンジン管理者の意図を推定するモデルは与えられたフィードバックを学習する。一連のフィードバックの後、全てのクエリ、文書において検索エンジン管理者の意図を推定するモデルを適用し、部分的な改善を行う。提案手法の有効性を検証するために、日本語の質問検索用データセットである OpenLive-Q を利用してデータセットを作成した。実験ではこのデータセットを用いて、ベースライン手法と提案手法を比較し、提案手法の有効性を確認した。

キーワード 情報検索, ランキング学習, 能動学習, クエリ推薦

1 はじめに

近年、ランキング学習などの機械学習アルゴリズムやニューラルネットワークを用いた言語モデルが検索アルゴリズムにも用いられるようになり、情報検索における文書のランク付けアルゴリズムは非常に複雑になってきている。例えば、Google などはニューラル言語モデルである BERT を検索に用いており、これを含む 200 種類以上の特徴を組み合わせることによってランキングを生成していると言われている。

他方、このような高度な検索アルゴリズムであっても、部分的に適切でない順位づけを行っている場合があり、検索エンジンを提供する側としては、ランキングの一部を修正したい場合がある。例として検索エンジンで同姓同名の犯罪者が検索結果に表示されてしまったため、全く関係ない人が風評被害を受けてしまい、風評被害を受けた本人に対する検索エンジンの名誉毀損が認められた事件があった [3]。この場合に、検索アルゴリズムを修正する必要があるが、機械学習に基づいた、複雑なアルゴリズムを人手で修正するのは容易ではない。また、修正した結果が別の検索結果に影響を及ぼす場合もあるため、他のクエリに対する結果に問題がないかを確認する必要も生じる。

そこで、本研究では検索エンジン管理者からのフィードバックに基づいて、ランキングアルゴリズムの部分的な改善を可能にする手法を提案する。ここでいう部分的改善とは「特定のクエリで特定の文書の順位を理想の順位に変える行為」と定義される。例えば人物名の検索結果で犯罪者に関係するサイトの順位を下げるといった変更がこの部分的改善に当てはまる。

提案手法において、検索エンジン管理者は、改善が必要なクエリとそのランキングの改善箇所をフィードバックとして与え

る。システム内では検索エンジン管理者の意図を推定するモデルを用意する。以下の1から3のステップを繰り返して検索エンジン管理者の意図を推定するモデルを学習させる。(1) 検索エンジン管理者の意図を推定するモデル側からフィードバックして欲しいクエリを提示, (2) 検索エンジン管理者は提示されたクエリに対してフィードバックを与える, (3) 検索エンジン管理者の意図を推定するモデルは与えられたフィードバックを学習する。一連のフィードバックの後、全てのクエリ、文書においてユーザの意図を推定するモデルを適用し、部分的な改善を行う。今回の問題設定では変更の対象となるクエリ以外のランキングは変わらないため、特にユーザランキング関数においては、変更すべきクエリのランキングを学習させることが非常に重要である。変更の対象となるクエリをより多く選ぶために、検索エンジン管理者の意図を推定するモデルがクエリ変更の対象であると予測するクエリを選んだ。

提案手法の有効性を検証するためには、特定のクエリで特定の文書の順位のみが変わったデータセットが必要であるが、それに適したデータセットが入手できなかったため、新たにデータセットを作成した。データセットの作成には日本語の質問検索用データセットである OpenLive-Q を利用し、OpenLive-Q 内のクエリを特定の基準に沿ってグルーピングし、グルーピングされたクエリの結果のみ特定の文書の順位を下げることで特定のクエリで特定の文書の順位が変わった理想の順位を再現した。実験ではこのデータセットを用いて、ベースライン手法と提案手法を比較し、提案手法の有効性を検証した。実験を行った結果、提案するクエリの選び方が多く文書の順位を改善した結果となった。

この論文における我々の貢献を以下に示す：(1) ランキングの部分的改善という新たな問題設定に取り組んだ。(2) ユーザ

の意図を推定するモデルと能動学習を用いたランキングの部分的な改善を行う手法を提案した。(3) 部分的な改善を再現するデータセットを作成した(4) 実験を行い提案手法の有効性を示した。

本論文の構成は以下の通りである。2 節ではユーザのフィードバックに基づいたランキングの改善手法に関する関連研究、および、ランキング学習における能動学習の利用について述べる。3 節では問題設定を説明し、提案手法の詳細について述べる。4 節では実験結果を示す。最後に、5 節では今後の課題と共に本論文の結論を述べる。

2 関連研究

本節では、ユーザのフィードバックに基づいたランキングの改善手法に関する関連研究、および、ランキング学習における能動学習の利用について述べる。

2.1 ユーザのフィードバックに基づいたランキング改善手法

本節ではユーザのフィードバックに基づいたランキングの改善手法に関する既存研究について述べる。これまでもユーザのフィードバックに基づいたランキングの改善手法に関する研究が行われているが大きく分けて暗黙的なフィードバックを用いた手法と明示的なフィードバックを用いた手法の2つに分けられる。

暗黙的なフィードバックを用いた手法[1],[8],[10],[15],[16]とはユーザの行動から文書の適合性を推測し、推測した文書の適合性をランキングアルゴリズムに学習させる手法である。暗黙的なフィードバックの例としてユーザのクリックログなどがある。検索結果をユーザに表示した際に、ユーザがクリックした文書を適合文書だとみなして学習を行う。これらの手法は基本的には、不特定多数のユーザからのフィードバックに基づいてランキングの改善を行う。複数のユーザがフィードバックを与える際、ユーザによってランキングの良し悪しが変わる可能性がある。例えば人物名の検索結果でランキングを変えようとした時、森というクエリは人物の名字とともることができるが森林に関する名詞の森とともれる。このようにユーザによって改善の基準が異なる可能性があるため今回の問題に暗黙的なフィードバックを用いた手法を用いるのは困難であると考えられる。今回の問題では1人のユーザによるフィードバックによってランキングの改善を行う。

明示的なフィードバックを用いた手法[19],[20],[18]とはユーザから直接得た文書の適合性をランキングアルゴリズムに学習させる手法である。明示的なフィードバックの代表例として適合性フィードバックがある。適合性フィードバックとは検索結果が与えられた時に、表示される文書に対して適合 / 不適合のどちらかを与えるフィードバック形式である。zhai と Lafferty は適合性フィードバックをクエリの言語モデルの推定に活用する手法[19]を提案した。この手法ではクエリの言語モデルと文書の言語モデルとの KL ダイバージェンスに基づいて文書のスコア付けを行っている。Xu と Akella はユーザの適合

性フィードバックを用いてロジスティック回帰モデルを更新する手法[20]を提案した。この手法では適合性フィードバックを学習する際のモデルのオーバーフィッティングを防ぐために、特徴量を、フィードバックを与える前の検索スコア、関連文書への距離、非関連文書への距離の3つに減らして学習を行う。Lv と Zhai は適合性フィードバックを取り入れる程度を予測する手法[18]を提案した。この手法ではクエリの識別性、文書の識別性、クエリと文書の乖離の3つの観点から特徴量を抽出し適合性フィードバックをどれほど取り入れるかの指標であるバランス係数を予測した。これらの手法では一人のユーザのフィードバックを用いてランキングを改善するという点で共通しているが、適合性フィードバックを用いた手法ではユーザがシステムに与えられる情報は文書が適合/不適合の2択であるため変更したい文書の順位を指定することはできない。またこれらの手法はフィードバックされたクエリの検索結果の改善のみに着目し、フィードバックされていない他の検索結果に対する影響を考慮していない。今回の問題設定では変更の対象となるクエリ、文書以外は順位を変えないことを条件としている。そのため今回の問題設定ではこれらの手法を適用するのは困難である。

2.2 能動学習

本節ではランキング学習における能動学習に関する既存研究[2],[4],[5],[9]に関して述べる。Seung らは QBC (Query By Committee) と呼ばれる能動学習手法[9]を提案した。この手法では複数のモデルを用意しそれぞれのモデルでラベル付けを行い、最終的に各々のモデルの予測が異なっていたクエリを選出する事で不確実なクエリを探す手法である。本研究の問題設定上、QBC のモデルに事前に学習させるデータが無く、フィードバックで得られるデータも極めて少ないため本研究への適用は困難であると考えられる。Tian と Lease は効率よく適合性フィードバックを学習するための文書に対する能動学習手法[2]を提案した。この手法あくまで文書のみが対象であり本研究の問題設定のように学習するクエリから選ぶタスクでは適用が困難である。Bo らは期待される損失に基づいたランキング関数の能動学習手法[4],[5]を提案している。この手法ではクエリレベルで損失を計算した後、文書レベルでも期待損失を計算し期待損失が大きいクエリと文書を選択する。これらの手法ではランキングの変化に着目するという点で本研究と類似しているが、今回は使用するモデルがクエリ識別モデルとユーザランキング関数の2種類存在し、かつクエリ識別モデルはランキング関数でない。そのためこの手法も本研究への適用は困難であると考えられる。

3 提案手法

本節では、まず問題設定について説明を行う。それから、提案手法の詳細について述べる。

3.1 問題設定

本節では本研究が取り組む問題設定に関して説明し、本研究の問題に対してのアプローチを説明する。本論文ではクエリ

Algorithm 1 提案手法の詳細**Input:** F_m, F_r

```

 $m_1 \leftarrow \text{LearnQuery}(F_m, m_0)$ 
 $r_1 \leftarrow \text{LearnRank}(F_r, r_0)$ 
for  $i \leftarrow 2 \dots n$  do
   $Q_i \leftarrow \text{GetQuery}(Q, m_{i-1}, r_{i-1})$ 
   $F_m, F_r \leftarrow \text{GetFeedback}(Q_i)$ 
   $m_i \leftarrow \text{LearnQuery}(F_m, m_{i-1})$ 
   $r_i \leftarrow \text{LearnRank}(F_r, r_{i-1})$ 
end for
for  $q \in Q$  do
  if  $\text{Predict}(q, m_n) = 1$  then
     $R_q \leftarrow \text{Rank}(q, r_n)$ 
  else
     $R_q \leftarrow \text{Rank}(q, r_0)$ 
  end if
end for
return  $\{R_q \mid q \in Q\}$ 

```

集合 Q , 文書集合 D が与えられた際に実数を返すランキング関数 r_0 を $r_0 : Q \times D \rightarrow \mathbb{R}$ と定義する. クエリ $q \in Q$, 文書 $d_i \in D$, 文書 $d_j \in D$ が与えられた時, ランキング関数 r_0 は以下で示すように文書のスコアに応じてランキングを生成する.

$$r_0(q, d_i) < r_0(q, d_j) \Rightarrow d_i \prec d_j$$

$r_0(q, d_i)$ はランキング関数 r_0 が予測するクエリ q に対する文書 d_i のスコアである. $r_0(q, d_j)$ はランキング関数 r_0 が予測するクエリ q に対する文書 d_j のスコアである. \prec は全順序を示している. ランキングの部分的改善を行うこととは, クエリ集合 Q に含まれるクエリに既存のランキング関数 r_0 を適用することによって出力されたランキングに対して, 検索エンジン管理者の意図するクエリの部分集合 S に含まれるクエリに検索エンジン管理者の意図するランキング関数 r^* を適用し, クエリ集合 Q において, 検索エンジン管理者の意図するクエリの部分集合 S に含まれないクエリには既存のランキング関数 r_0 を適用してランキングを出力する事である. 例えば人物名に関するクエリというクエリ集合においてのみ犯罪者に関する文書を低い順位にランク付けするランキング関数を適用することが挙げられる.

本研究では検索エンジン管理者の意図するクエリの部分集合 S と検索エンジン管理者の意図するランキング関数 r^* を推定する問題に取り組む. 推定したクエリの部分集合を S' , 推定したランキング関数を r' とすると, 推定したクエリの部分集合 S' に含まれるクエリには推定したランキング関数 r' を適用し, S' に含まれないクエリにはランキング関数 r_0 を適用することでランキングを出力し, 部分的な改善がなされているランキングと比較して部分的な改善がなされたか, 他のランキングに影響を与えていないかを評価する.

3.2 フレームワーク

本研究では検索エンジン管理者の意図するクエリ集合 S を推

定するクエリ識別モデル m と検索エンジン管理者の意図するランキング関数 r^* を推定するユーザランキング関数 r の2つのモデルを用いる.

提案手法の詳細をアルゴリズム 1 に示す. 検索エンジン管理者は, 初めに改善が必要なクエリを複数個選ぶ. ここで選んだクエリの集合を $Q_1 \in S$ とする. そしてクエリ集合 Q_1 に含まれるクエリが検索エンジン管理者の意図するクエリの部分集合 S に含まれるか否かを示すフィードバック F_m とそのランキングの改善箇所 F_r を入力として与える. クエリ集合 Q_1 に含まれるクエリが検索エンジン管理者の意図するクエリの部分集合 S に含まれるか否かを示すフィードバック F_m をクエリ識別モデル m_0 に, ランキングの改善箇所 F_r をユーザランキング関数 r_0 にそれぞれ学習させる ($\text{LearnQuery}(F_m, m_0), \text{LearnRank}(F_r, r_0)$). その後以下の 1 から 3 ステップを $n - 1$ 回繰り返して, 先ほど検索エンジン管理者が与えたフィードバックを学習したクエリ識別モデル m_1 と先ほど検索エンジン管理者が与えたフィードバックを学習したユーザランキング関数 r_1 を追加で学習させる. 注意すべきは以下に記した 1 から 3 のステップは i 回目のフィードバックを学習するときの流れである. (1) $i - 1$ 回分のフィードバックを学習したクエリ識別モデル m_{i-1} と $i - 1$ 回分のフィードバックを学習したユーザランキング関数 r_{i-1} 側から i 回目にフィードバックして欲しいクエリ Q_i を提示する ($\text{GetQuery}(Q, m_{i-1}, r_{i-1})$), (2) 検索エンジン管理者は提示されたクエリ集合 Q_i に対して Q_i に含まれるクエリが検索エンジン管理者の意図するクエリの部分集合 S に含まれるか否かを示すフィードバック F_m と含まれる場合, そのクエリに対するランキングの改善箇所 F_r を与える ($\text{GetFeedback}(Q_i)$), (3) $i - 1$ 回分のフィードバックを学習したクエリ識別モデル m_{i-1} は Q_i に含まれるクエリが改善の対象であることを示すフィードバック F_m を学習し, $i - 1$ 回分のフィードバックを学習したユーザランキング関数 r_{i-1} はランキングの改善箇所 F_r を学習する ($\text{LearnQuery}(F_m, m_{i-1}), \text{LearnRank}(F_r, r_{i-1})$). クエリ $q \in Q_i$ に対して検索エンジン管理者が与える q がクエリが検索エンジン管理者の意図するクエリの部分集合 S に含まれるか否かを示すフィードバック F_m と q のランキングの改善箇所 F_r はそれぞれ以下のように表される.

$$F_r = \{(d_i, d_j) \mid r^*(q, d_i) < r^*(q, d_j)\}$$

$$F_a = \begin{cases} 1 & (q \in S) \\ 0 & (q \notin S) \end{cases}$$

F_r はクエリ q では検索エンジン管理者の意図するランキング関数 r^* においては文書 d_i よりも文書 d_j の順位が上であることを示すペアワイズプリファレンスである. F_m は $F_m \in (0, 1)$ でありクエリ q が検索エンジン管理者の意図するクエリ集合 S に含まれる場合は 1 を返し, クエリ q が検索エンジン管理者の意図するクエリ集合 S に含まれない場合は 0 を返す.

一連のフィードバックの後, n 回のフィードバックを学習したクエリ識別モデル m_n がクエリ集合 Q の全てのクエリ $q \in Q$ に対して検索エンジン管理者の意図するクエリの部分集合 S に含まれるか否かを予測し ($\text{Predict}(q, m_n)$), n 回のフィー

ドバックを学習したクエリ識別モデル m_n が検索エンジン管理者の意図するクエリの部分集合 S に含まれると予測したクエリ ($\text{Predict}(q, m_n) = 1$) に対してはユーザランキング関数 r' を適用し、クエリ q に対するランキング R_q を出力し、 n 回のフィードバックを学習したクエリ識別モデル m が検索エンジン管理者の意図するクエリの部分集合 S に含まれない ($\text{Predict}(q, m_n) = 0$) と予測したクエリに対してはランキング関数 r_0 を適用しクエリ q に対するランキング R_q を出力することでランキングの部分的な改善を行う。

3.3 クエリ識別モデル

本節ではユーザの意図するクエリの部分集合 S を推定する方法に関して述べる。クエリ $q \in Q$ が与えられた時、クエリ q が検索エンジン管理者の意図するクエリ集合 S に含まれているか否かを2値分類するクエリ推定モデルを用意し、このモデルをクエリ集合 Q の全てのクエリに対して適用することでユーザの意図するクエリの部分集合 S を推定する。クエリ識別モデルはクエリ $q \in Q$ が与えられた時、そのクエリがユーザの意図するクエリの部分集合 S に含まれているか否かの予測 y_q を出力する。クエリがユーザの意図するクエリの部分集合 S に属しているか否かの予測 y_q は以下のように定義される。

$$y_q = \begin{cases} 1 & (P(y = 1|q) \geq 0.5) \\ 0 & (P(y = 1|q) < 0.5) \end{cases}$$

$P(y = 1|q)$ はクエリ識別モデル m がクエリ q を検索エンジン管理者の意図するクエリ集合 S に含まれると予測する確率である。

3.4 ユーザランキング関数

本節ではユーザの意図するランキング関数 r^* を推定する方法に関して述べる。本論文ではユーザランキング関数 r を用意し、ユーザのフィードバック F_r を学習させることによりユーザの意図するランキング関数 r^* を再現する。ユーザランキング関数 r はクエリ q 、文書 d が与えられた時、その文書 d のクエリ q に対するスコアを出力する関数である。ユーザランキング関数も r ランキング関数 r_0 と同様に文書のスコアに応じてランキングを生成する。

3.5 能動学習

一人のユーザが与えられるフィードバックの量には限りがある。そのため本研究では能動学習を行い、モデル側からユーザに対してフィードバックして欲しいクエリを選択し提示する。今回の問題設定では検索エンジン管理者の意図するクエリの部分集合 S に含まれるクエリのランキングのみ変化する。そのため特にユーザランキング関数においては、検索エンジン管理者の意図するクエリの部分集合 S に含まれるクエリのランキングを学習させることが非常に重要である。そのためクエリ選択の基準としてはスコア関数 score 関数を定義しその値が大きいものをフィードバックして欲しいクエリとしてユーザに提示する。

スコア関数は大きく分けて2つの要素から成り、1つがクエリ識別モデル m が検索エンジン管理者の意図するクエリ集合

S に含まれると予測する確率、もう一つがランキング関数 r_0 が出力するランキングとその時点でのユーザランキング関数 r が出力するランキングのスパイマンの順位相関係数 [14] の差である。

クエリ識別モデル m が検索エンジン管理者の意図するクエリ集合 S に含まれると予測する確率を採用した理由は、クエリ識別モデル m の観点からユーザの意図するクエリの部分集合 S に含まれるクエリを推定するためである。

ランキング関数 r_0 が出力するランキングとその時点でのユーザランキング関数 r が出力するランキングとのスパイマンの順位相関係数の差を採用した理由は、ランキングの変化の大きいクエリはユーザランキング関数の学習においてより多くのフィードバックが期待できるためである。そのためランキングの変化の度合いを図ることができるスパイマンの順位相関係数の差を採用した。

クエリ識別モデルが改善の対象であると予測する確率はクエリ識別モデル m の観点から、スパイマン順位相関係数の差はユーザランキング関数 r の観点からユーザの意図するクエリの部分集合 S に含まれる可能性の高いクエリを選択できるようにしている。クエリ $q \in Q$ のスコア関数 score は以下のように定義される。

$$\text{score}(q) = \alpha P(y = 1|q) + (1 - \alpha)(1 - \text{spr}(R_0, R_{t-1}))$$

$P(y = 1|q)$ はクエリ識別モデルが改善の対象であると予測する確率、 $\text{spr}(R_0, R_{t-1})$ はランキング関数 r_0 が出力するクエリ q のランキング R_0 と $t-1$ 回分のフィードバックを学習した後のユーザランキング関数 r_{t-1} が出力するクエリ q のランキング R_{t-1} とのスパイマンの順位相関係数であり、 α はパラメータである。クエリ識別モデルが改善の対象であると予測する確率 $P(y = 1|q)$ は以下のように定義される。

$$P(y = 1|q) = \frac{1}{1 + \exp^{-q_{\text{dist}}}}$$

q_{dist} とはクエリ識別モデルの境界面とクエリベクトルとの符号付き距離である。ランキング関数 r_0 が出力するクエリ q のランキング R_0 、 $t-1$ 回分のフィードバックを学習した後のユーザランキング関数 r_{t-1} が出力するクエリ q のランキング R_{t-1} 、 R_0 と R_{t-1} に順位付けされている文書の集合 D_q が与えられた時、スパイマンの順位相関係数は以下のように求められる。

$$\text{spr}(R_0, R_{t-1}) = 1 - \frac{6 \sum_{d \in D_q} (d_0 - d_{t-1})^2}{N^3 - N}$$

d_0 はランキング関数 r_0 が出力するクエリ q のランキング R_0 における文書 $d \in D_q$ の順位、 d_{t-1} は $t-1$ 回目のフィードバックを学習したユーザランキング関数 r_{t-1} が出力するクエリ q のランキング R_{t-1} の文書 $d \in D_q$ の順位、 N は R_0 と R_{t-1} に順位付けされている文書の集合 D_q に含まれる文書の数である。

4 実 験

我々のタスクには公開されたデータセットが存在しないため、

まずデータセットの作成の概略について説明する．それからベースライン手法を含む実験設定について紹介し最後に実験結果を示す．

4.1 データセット

提案手法の有効性を検証するためには，特定のクエリで特定の文書の順位のみが変わったデータセットが必要であるが，それに適したデータセットが入手できなかったため，新たに日本語の質問検索用データセットである OpenLive-Q [12] を使ってデータセットを作成した．OpenLive-Q はクエリと質問から成り，クエリと質問のペアに対して適合度が与えられているデータセットである．主にクエリに対する質問を適合度順に並び替えるタスクに使われており本研究の問題設定に適している．

データセットの作成手順を以下に示す．特定のクエリで特定の文書の順位のみが変わったデータセットを作成するにはそのベースとなるランキングが必要である．OpenLive-Q ではクエリと質問の適合度しか与えられておらず，ベースとなるランキングが存在しなかったため，ベースとなるランキングを作成するために OpenLive-Q 内のクエリをモデルの学習に使う教師データ (896 個)，モデルの評価に使うテストデータ (497 個)，ハイパーパラメータの調整に使う検証データ (100 個) の 3 つに分け，ランキング関数に教師データのクエリと OpenLive-Q 内で適合度が与えられている文書との適合度を学習させた．このランキング関数を r_0 とする．その後教師データ，テストデータ，検証データ内の全てのクエリに対して学習させたランキング関数 r_0 を適用してランキングを出力した．

その後，検索エンジン管理者の意図するクエリの部分集合 S を再現するために DBpedia [11] を用いてクエリのグルーピングを行った．DBpedia とは Wikipedia から情報を抽出してその抽出した情報を公開しているコミュニティプロジェクトである．DBpedia を用いることでクエリが属しているカテゴリの情報を得ることができる．DBpedia では基本的に正式名称でクエリを検索しなければ正確に情報を得ることができない．例えば東京大学と DBpedia 内で検索すれば正しく東京大学が属しているカテゴリを得ることができる．しかし東大という略称で DBpedia 内で検索すると正しいカテゴリを得ることができない．多くの略称には正式名称に対するリダイレクト先が存在している．先ほどの東京大学の例であれば東大という東京大学の略称で DBpedia 内で検索するとリダイレクト先に東京大学が登録されている．OpenLive-Q 内のクエリの一部にも略称となってしまうクエリが存在した．これらのクエリに対しては正式名称での正確な情報を得るためにリダイレクト先から正式名称を入手し，そこから得られたカテゴリをクエリのグルーピングに使用した．カテゴリの例としては人物名や職種，大学名などがある．クエリのグルーピングはクエリが共通して持つカテゴリごとに行い，最終的には 25 種類のクエリのグループを作成した．

クエリのグルーピングの後，検索エンジン管理者の意図するランキング関数 r^* が出力するランキングを再現するために次の操作を行った．初めにランダムなシード値を固定し，25 種類

のクエリグループ別に教師データ内でグループに属するクエリをランダムで 1 つ選び，選んだクエリに対してランキング関数 r_0 が出力するランキングの上位 10 件の文書の中からランダムで 1 つ文書を選び，その文書の文書ベクトルを基準のベクトルとした．教師データ，テストデータ，検証データの内，グルーピングされているクエリに対して基準のベクトルとコサイン類似度が近い文書ベクトルを上位 10 件取得し，その順位を下位 10 位に変更した．この作業によって特定の文書の順位のみ変更することを再現した．文書の順位を変更したランキングに関しては理想のランキングと呼ぶこととする．

今回のデータセット作成作業ではランダムで選ぶクエリグループ内のクエリと，基準のベクトルによってはデータセット自体が大きく変化する．ランダムの影響を考慮してランダムのシード値は 0 から 9 の計 10 個用意した．そのため，最終的には (クエリのグループの数の 25 グループ) \times (シードの数 10 個) の計 250 件の理想のランキングを作成した文書ベクトルにはユーザランキング関数の学習に使用される特徴量ベクトルを使用した．

4.2 実験設定

本研究ではモデルの性能を評価するには検索エンジン管理者の意図するクエリの部分集合 S と検索エンジン管理者の意図するランキング関数 r ないし検索エンジン管理者の意図するランキング関数 r が出力したランキングが必要不可欠である．しかし実際のユーザがクエリ集合 Q の全てのクエリに対して，検索エンジン管理者の意図するクエリの部分集合 S に含まれているか否かの情報やランキングの改善箇所の情報を与えることはクエリ集合 Q のクエリ数が膨大であるため難しい．そのため本研究では実際にユーザからフィードバックを得ることは行わず，作成したデータセット内の理想のランキングからフィードバックを得て評価を行うシミュレーション形式を採用した．

我々は実験におけるベースライン手法として，ランダムでクエリを選ぶクエリ推薦方法を定めて提示するクエリの選び方を比較し，部分的な改善がなされたか，他のランキングに影響を与えていないかを評価した．また本研究の提案手法であるスコア関数は，1 つがクエリ識別モデル m が検索エンジン管理者の意図するクエリ集合 S に含まれると予測する確率，もう一つがランキング関数 r_0 が出力するランキングとその時点でのユーザランキング関数 r が出力するランキングとのスピアマンの順位相関係数の差の 2 つの要素から成る．そのためそれぞれの要素のみで選ばれるクエリの選び方とも比較を行う．またクエリ識別モデルを使用することも本研究の提案に含まれるためクエリ識別モデルを使わない手法とも比較を行った．クエリ識別モデルなしの手法はクエリ識別モデルを使わずに全てのクエリに対してユーザランキング関数を適用した手法である．構築したデータセットを用いた実験の手順を以下で述べる．

初めに変更の対象となるクエリ 4 つと対象でないクエリ 1 つをランダムで選びそれらのクエリが変更の対象であるかどうかをクエリ識別モデル m に学習させ，変更の対象となるクエリ 4 つに関しては理想のランキングから正しいランキングを得てラ

ンキング関数 r に学習させた。その後はそれぞれのクエリ選定基準でクエリを 5 つ選び、それらのクエリがそれぞれクエリが検索エンジン管理者の意図するクエリの部分集合 S に含まれるか否かをクエリ識別モデル m に学習させて、もしその中にクエリが検索エンジン管理者の意図するクエリの部分集合 S に含まれるクエリが存在した場合は理想的なランキングからランキングを取得し、ランキング関数 r に学習させた。このクエリを選んでモデルを学習させる流れを 15 回繰り返した ($n = 15$)。

本研究の問題設定では文書の順位を変更する必要がある。そのため適合度を正確に予測する事は重要ではなく、あくまで文書間のスコアの大小関係を正確に予測することが非常に重要である。そのためユーザランキング関数は文書の適合度の大小関係を学習するペアワイズランキング手法である RankNet [6], [17] を使用した。また、クエリ識別モデル m には線形 SVM (サポートベクトル・マシン) を使用した [7]。

クエリ識別モデルの学習に用いるクエリの特徴ベクトル \mathbf{v}_q は、データセット内のクエリ・文書特徴量が存在する文書の単語頻度ベクトルの和を採用した。具体的にはクエリ q が与えられた際に、データセット内でクエリ q に対して関連度が高い文書の集合を D_q とし、クエリの特徴量は以下のように定義される。

$$\mathbf{v}_q = \sum_{d \in D_q} \mathbf{v}_d$$

\mathbf{v}_d は文書 d の tf ベクトルであり全語彙の集合 V のうち 5 件以上の文書に出現した単語の集合である V' を用いて作られる V' 次元のベクトルで、単語 $st \in V'$ のインデックスに対応する次元がその単語の出現頻度であるベクトルである。

ユーザランキング関数の学習に使用する文書の特徴量は OpenLiveq ライブラリ¹で得られる特徴量を採用した。この特徴量はランキング学習モデルのベンチマークセットである LETOR [13] で利用されている 30 個の特徴量に加えて、与えられた回答の数、ページビューの数などを加えた特徴量である。

クエリ識別モデルとユーザランキング関数のハイパーパラメータは検証データを用いて決定された。

4.3 実験結果

4.3.1 Pairwise Difference

本研究では作成したランキングがどれほど部分的改善がなされたかを評価するために Pairwise Difference という評価指標を作成した。Pairwise Difference は任意の文書のペアのうち順位を入れ替えるべき文書のペアを適合文書とみなして実際に入れ替わったかどうかを評価する評価指標である。評価の手順としては、初めに理想のランキングと既存のランキングとを比較を行なって順位が入れ替わった文書ペアの集合を取得する。次に評価対象のランキングと既存のランキングとを比較を行なって順位が入れ替わった文書ペアの集合を取得する。Pairwise Difference ではこれらの文書ペアの集合のうち、文書ペア一つを適合文書とみなす。最終的には理想のランキングと既存

表 1 ユーザに提示したクエリのうち変更の対象であったクエリの割合。

クエリ識別モデルあり	
提案手法	14.208
確率のみ	4.916
順位相関係数のみ	5.504
ランダムクエリ	5.9

のランキングとを比較を行なって得た適合文書集合と、評価対象のランキングと既存のランキングとを比較を行なって得た適合文書集合を比較して適合率、再現率、F 値を求める。

4.3.2 手法ごとの Pairwise Difference F 値の推移

図 1 に各クエリ選定基準によるフィードバック毎の理想のランキング全体との Pairwise Difference の F 値の平均値を示す。縦軸が理想のランキング全体との Pairwise Difference の F 値の平均値であり横軸が与えたフィードバックの回数である。実験の結果、全体的には低めの数値であったが提案手法が高い数値を示した。また、提案手法のクエリの選び方のうちランキング関数 r_0 が出力するランキングとその時点でのユーザランキング関数 r が出力するランキングとのスピアマンの順位相関係数の差のみを用いてクエリを選ぶ手法、クエリ識別モデル m が検索エンジン管理者の意図するクエリ集合 S に含まれると予測する確率のみでクエリを選ぶ手法とランダムでクエリを選ぶ手法と比べて低い値を示した。以上のことからあくまでクエリ識別モデル m が検索エンジン管理者の意図するクエリ集合 S に含まれると予測する確率、もう一つがランキング関数 r_0 が出力するランキングとその時点でのユーザランキング関数 r が出力するランキングとのスピアマンの順位相関係数の差の 2 つの要素を組み合わせることが非常に重要であるということが考えられる。また、クエリの選び方で大きくモデルの性能が変わっているため、本研究で取り組んでいるランキングの部分的改善というタスクにおいては学習するクエリの選び方は非常に重要であるということが考えられる。

4.3.3 クエリ識別モデルの有無

図 2 に提案手法でのクエリ識別モデル使用時と不使用時の Pairwise Difference F 値の推移を示す。実験の結果、クエリ識別モデルを使用した場合が Pairwise Difference F 値が高い結果となった。この結果からクエリ識別モデルを使用し、変更を加えるランキングを制限することで、より正確にランキングの部分的な改善ができてることが考えられる。

4.3.4 能動学習の評価

表 1 に各クエリの選定基準ごとのユーザに提示したクエリのうち検索エンジン管理者の意図するクエリの部分集合 S に含まれていたクエリの割合の平均値を示す。確率のみ、順位相関係数のみ、ランダムクエリのクエリ選択基準では 1 データ、15 フィードバックの間で検索エンジン管理者に提示される 75 個のクエリのうち平均 5 から 6 個のクエリが検索エンジン管理者の意図するクエリの部分集合 S に含まれるクエリであった。対して提案手法では 75 個の検索エンジン管理者に提示されるク

1: <https://github.com/mpkato/openliveq>

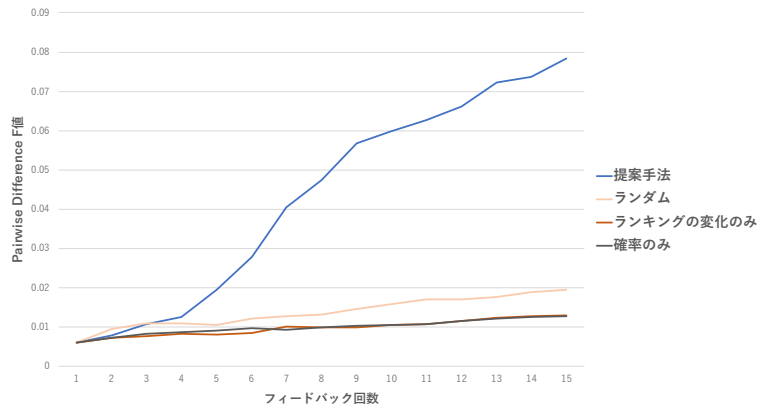


図 1 手法ごとの Pairwise Difference F 値の推移.

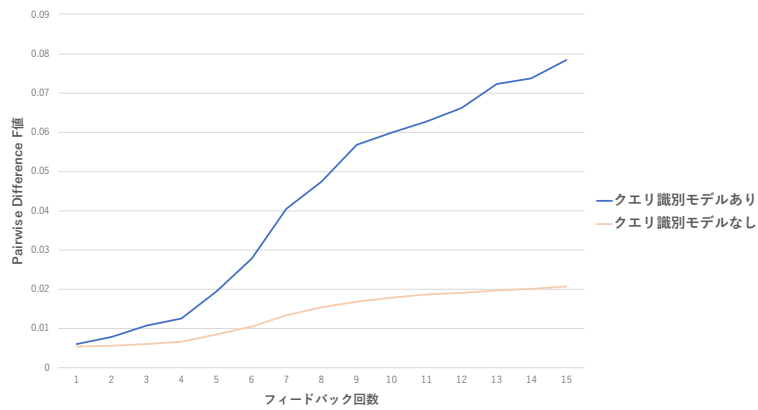


図 2 クエリ識別モデル使用時と不使用時の Pairwise Difference F 値の推移.

表 2 クエリの選び方によるクエリ識別モデルの性能の差.

	適合率	再現率	F 値
提案手法	0.00153	0.00066	0.00092
確率のみ	0.00030	0.00218	0.00053
順位相関係数のみ	0.00074	0.00082	0.00078
ランダムクエリ	0.00111	0.00081	0.00094

表 3 改善された文書の割合が高かったクエリグループ.

クエリグループ	F 値	改善の対象となるクエリを選んだ回数
大学名	0.19182	20.5
教育関係	0.17766	22.4
組織名	0.1505	21.0

表 4 改善された文書の割合が低かったクエリグループ.

クエリグループ	F 値	改善の対象となるクエリを選んだ回数
和製英語	0.0	4.4
ゲーム	0.00036	10.0
製品名	0.00103	5.0

エリのうちおよそ 3 倍の 14 個のクエリが検索エンジン管理者の意図するクエリの部分集合 S に含まれるクエリであった。このことから本研究で提案しているスコア関数を最大化するクエリの選び方は検索エンジン管理者の意図するクエリの部分集合 S に含まれるクエリを選ぶという点では有効であると考えられる。また表??の改善した文書の数に着目してみると、クエリ識別モデルを使用しない学習方法において、検索エンジン管理者の意図するクエリの部分集合 S に含まれるクエリを多く提示したスコア関数を最大化するクエリの選び方がより多くの文書の順位を変更した結果となった。このことから変更の対象に当てはまるクエリを多く学習することがランキング関数の性能の向上に貢献している可能性が高い。

4.3.5 クエリ識別モデルの評価

表 2 にクエリの選び方によるクエリ識別モデルの性能の差を 3 つの評価指標である適合率、再現率、F 値を用いて示す。適合

率に関しては提案手法、再現率に関しては確率のみのクエリの選び方、F 値に関してはランダムで選ぶクエリの選び方がそれぞれ高い数値を示した。この結果から提案手法のクエリの選び方はクエリ識別モデルの性能には寄与しないことが考えられる。

4.3.6 クエリグループ

表 3、表 4 に提案手法において 25 個のクエリグループのうち、ユーザのフィードバックを学習することで改善された文書の割合が高かったクエリグループ 3 件と改善された文書の割合が低かったクエリグループ 3 件、また能動学習の際に改善の対象となるクエリを選んだ回数の平均値を示す。「大学名」、「教

育関係」,「組織名」などといったクエリグループが最も改善できたグループとなっていた。能動学習の際に改善の対象となるクエリを選んだ回数の平均値に着目してみると,改善された文書の割合が高かったクエリグループでは計 75 個のクエリのうち,およそ 20 個の改善の対象となるクエリを選んでいった。一方,「和製英語」,「ゲーム」,「製品名」といったクエリグループが改善率が 0 に近い結果となった。能動学習の際に改善の対象となるクエリを選んだ回数の平均値に着目するとおよそ 5 から 10 個の改善の対象となるクエリを選んでいった。これらのことから,改善の対象となるクエリを選ぶことが部分的改善においては重要であることが考えられる。

5 ま と め

本論文では検索エンジン管理者からのフィードバックに基づいてランキングアルゴリズムの部分的な改善を可能にする手法を提案した。ここでいう部分的改善とは「特定のクエリで特定の文書の順位を理想の順位に変える行為」と定義される。提案手法では検索エンジン管理者の意図を推定するモデルを用意し,以下の 1 から 3 のステップを繰り返して検索エンジン管理者の意図を推定するモデルを学習させた。(1) 検索エンジン管理者の意図を推定するモデル側からフィードバックして欲しいクエリを提示,(2) 検索エンジン管理者は提示されたクエリに対してフィードバックを与える,(3) 検索エンジン管理者の意図を推定するモデルは与えられたフィードバックを学習する。モデル側からフィードバックして欲しいクエリを提示する際のクエリの選び方としては,変更の対象となるクエリをより多く選ぶために,検索エンジン管理者の意図を推定するモデルがクエリ変更の対象であると予測するクエリを選んだ。一連のフィードバックの後,全てのクエリ,文書において検索エンジン管理者の意図を推定するモデルを適用し,部分的な改善を行った。提案手法の有効性を検証するために,日本語の質問検索用データセットである OpenLive-Q を利用してデータセットを作成した。実験ではこのデータセットを用いて,ベースライン手法と提案手法を比較し,提案手法が変更の対象である文書の順位を多く改善した結果となった。今後の課題としてはスコア関数にクエリ識別モデルの性能の向上に着目した指標を追加することが考えられる。

謝辞 本研究は JSPS 科研費 21H03775 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] Agichtein, Eugene; Brill, Eric; Dumais, Susan. "Improving web search ranking by incorporating user behavior information". *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 2006, p. 19-26.
- [2] Aibo, Tian; Matthew, Lease. "Active learning to maximize accuracy vs. effort in interactive information retrieval". *SIGIR '11*. 2011, p. 24-28.
- [3] BBCNews. "Google loses australia 'gangland' defamation lawsuit". <https://www.bbc.com/news/technology-20153309>, (参照 2021-01-10).

- [4] Bilgic, Mustafa; Bennett, Paul N. "Active query selection for learning rankers". *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. 2012, p. 1033-1034.
- [5] Bo, Long; Chapelle, Olivier; Ya, Zhang; Yi, Chang; Zheng, Zhaohui; Tseng, Belle. "Active learning for ranking through expected loss optimization". *SIGIR '10*. 2010, p. 19-23.
- [6] Burges, Chris; Shaked, Tal; Renshaw, Erin; Lazier, Ari; Deeds, Matt; Hamilton, Nicole; Hullender, Greg. "Learning to rank using gradient descent". *Proceedings of the 22nd international conference on Machine learning*. 2005, p. 89-96.
- [7] Fan, Rong-En; Chang, Kai-Wei; Hsieh, Cho-Jui; Wang, Xiang-Rui; Lin, Chih-Jen. Liblinear: A library for large linear classification. *the Journal of machine Learning research*. 2008, vol. 9, p. 1871-1874.
- [8] Filip, Radlinski; Thorsten, Joachims. "Query chains: learning to rank from implicit feedback". *KDD '05*. 2005, p. 239-248.
- [9] H.S.Seung; M.Opper; H.Sompolsky. "Query by committee". *COLT*. 1992, p. 287-294.
- [10] Joachims, Thorsten; Swaminathan, Adith; Schnabel, Tobias. "Unbiased learning-to-rank with biased feedback". *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 2017, p. 781-789.
- [11] Kato, Fumihiko; Takeda, Hideaki; Koide, Seiji; Ohmukai, Ikki. "Building dbpedia japanese and linked data cloud in japanese". *2013 Linked Data in Practice Workshop (LDPW2013)*. 2014, p. 1-11.
- [12] Kato, Makoto P; Takehiro, Yamamoto; Tomohiro, Manabe; Akiomi, Nishida; Sumio, Fujita. "Overview of the ntcir-13 openliveq task". *Proceedings of the NTCIR-13 Conference*. 2017.
- [13] Qin, Tao; Liu, Tie-Yan; Xu, Jun; Li, Hang. Letor: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*. 2010, vol. 13, no. 4, p. 346-374.
- [14] Spearman, Charles. The proof and measurement of association between two things. 1961.
- [15] Thorsten, Joachims. "Optimizing search engines using click-through data". *KDD '02*. 2002, p. 133-142.
- [16] Xinyi, Dai; Jiawei, Hou; Qing, Liu; Yunjia, Xi; Ruiming, Tang; Weinan, Zhang; Xiuqiang, He; Jun, Wang; Yong, Yu. "U-rank: Utility-oriented learning to rank with implicit feedback". *CIKM '20*. 2020, p. 2373-2380.
- [17] Yu, Hai-Tao. "Pt-ranking: A benchmarking platform for neural learning-to-rank". 2020.
- [18] Yuanhua, Lv; Zhai, ChengXiang. "Adaptive relevance feedback in information retrieval". *CIKM '09*. 2009, p. 255-264.
- [19] Zhai, Chengxiang; Lafferty, John. "Model-based feedback in the language modeling approach to information retrieval". *Proceedings of the tenth international conference on Information and knowledge management*. 2001, p. 403-410.
- [20] Zuobing, Xu; Ram, Akella. "A bayesian logistic regression model for activerelevance feedback". *SIGIR '08*. 2008, p. 20-24.