ドメイン特徴に基づくランキング学習モデルのドメイン適応

伊藤　拓誠† 　丸田　敦貴†† 　加藤　誠†††

† 筑波大学 知識情報・図書館学類　〒 305–8550 茨城県つくば市春日 1–2
†† 筑波大学大学院 人間総合科学学術院　〒 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　〒 305–8550 茨城県つくば市春日 1–2
E-mail: †s2010485@u.tsukuba.ac.jp, ††s1711567@klis.tsukuba.ac.jp, †††mpkato@acm.org

あらまし　本論文は，ランキング学習でターゲットドメインの適合性判定データが利用できない設定において，ランキング学習モデルのドメイン適応を行うことを目的とする．我々は，ターゲットドメインからドメイン特徴を抽出し，ドメイン特徴のみからターゲットドメインにおけるランキングモデルの最適な重みを予測する重み予測モデルを提案する．また，重み予測モデルの学習には，多種多様なデータを含むデータセットをドメイン特徴が異なるように分割して複数のドメインを用意し，各ドメインでドメイン特徴の抽出とランキングモデルの重みの最適化を行うことで作成した学習データを利用する．評価実験の結果，重み予測モデルで重みを予測したモデルの性能が，ドメインの差異を考慮せずに学習した汎用的なモデルの性能を上回ることが確認された．

キーワード　情報検索，検索モデル，ランキング学習，ドメイン適応

1　はじめに

情報検索分野におけるランキングタスクは，与えられたクエリに対して関連する文書を適合性の高い順に並べ替えることを目的とする．このタスクを行うランキングモデルを構築する方法の一つにランキング学習 [11] がある．ランキング学習は機械学習を活用して性能の良いランキングモデルを構築する手法で，クエリと文書，それらの適合性判定データからなる学習データを大量に用いてモデルを学習する．ランキング学習の効果は広く認められており，Google や Bing といった文書検索を中心とするサービスで研究が行われているほか，Airbnb や eBay などのそれ以外のサービスでも活用されている [7, 20, 22, 27].

効果的なランキング学習を行うには，他の機械学習タスクと同様にタスクを行うドメインにおける学習データが大量に必要となる．学習データの中でも，人手によるアノテーションや大規模なクリックデータが必要な適合性判定データの収集が難しい．この課題に対して，学習データの豊富なソースドメインから獲得した知識を学習データの乏しいターゲットドメインに転移することで学習データの不足を補う転移学習の研究が行われている [1–3, 12]．しかしながら，これらの研究ではターゲットドメインの適合性判データを少なからず利用しており，ターゲットドメインの適合性判定データがない場合に性能の良いランキングモデルを構築する方法は依然として確立していない．

そこで本論文では，ターゲットドメインの適合性判定データを用いずに転移学習を行うトランスダクティブ転移学習 [18] という設定におけるランキング学習手法を提案する．トランスダクティブ転移学習の設定でランキング学習を行う場合，利用できるデータは一般的にソースドメインのクエリ，文書，適合性判定データに加え，ターゲットドメインのクエリ，文書となる．ただし，適合性判定データではないユーザの情報などを追加で利用することはできる．これらのデータのみを利用して，ターゲットドメインにおいて性能の良いランキングモデルを構築することが本研究の目的である．

本研究では，ドメインからドメイン特徴と呼ばれる特徴を抽出し，ドメイン特徴のみからそのドメインにおけるランキングモデルの最適な重みを予測する**重み予測モデル**を構築することで上記の提案を実現する．重み予測モデルは，複数のソースドメインを用意し，それらのドメインのドメイン特徴とランキングモデルの最適な重みとの関係性を学習することで構築する．また，複数のソースドメインを用意する際，単純にドメインの異なる複数のデータセットを集める方法では学習に十分な数を用意するのが難しい．そこで本研究では，多種多様なデータを含む 1 つの大きなデータセットをドメインが異なるように分割して複数のドメインを用意する方法を採用する．

実験では，分割元のデータセットとして多種多様な Web 文書とクエリを含む AOLIA データセット [13] を利用した．このデータセットをクエリ・文書ペアの特徴に基づき k-means 法で分割し，500 個のドメインを作成した．500 個のドメインを学習用のソースドメインと検証用のターゲットドメインおよび評価用のターゲットドメインに分割した後，各ドメインでドメイン特徴の抽出とランキングモデルの重みの最適化を行い，それらを入出力とする重み予測モデルを学習した．また，重みを予測する対象であるランキングモデルには，単純な重みを持つ線形のモデルを採用した．実験の結果，重み予測モデルで予測した重みを活用した提案モデルは，ドメインの差異を考慮せず大量のデータで学習した汎用的なモデルよりも性能が向上した．また，提案モデルは，最も性能が良くなると予想された，評価用に適合性判定データがあるターゲットドメインで直接学習したモデルよりも性能が高くなることが確認された．さらに，主にクエリと文書の関連度を測るクエリ・文書特徴をドメイン特徴として用いた場合に提案モデルの性能が最も高くなることが

明らかになった．

この論文における我々の貢献を以下に示す：(1) 転移学習の問題設定の中でもターゲットドメインのラベルを利用することができないトランスダクティブ転移学習という設定において利用することができるランキング学習タスクの手法を提案した．(2) 提案手法がドメインの差異を考慮せずに大量のデータで学習した汎用的なモデルよりも効果的であることを実験によって明らかにした．また，利用するドメイン特徴によって提案手法の性能が変化することを明らかにした．

本論文の構成は以下の通りである．2 節ではランキング学習における転移学習に関する関連研究，および， 適応的なランキング学習に関する関連研究について述べる．3 節では問題設定を説明し，重み予測モデルの学習方法とデータセットの分割方法について述べる．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，ランキング学習における転移学習に関する関連研究，および，適応的なランキング学習に関する関連研究について述べる．

2.1 ランキング学習における転移学習

学習データの豊富なソースドメインで得た知識を学習データの乏しいターゲットドメインに転移することでターゲットドメインの学習データの不足を補う転移学習（Transfer Learning）はランキング学習の分野でも研究されている．ランキング学習における代表的な転移学習の手法には，アルゴリズムレベルの手法 [1, 3]，特徴量レベルの手法 [2]，データレベルの手法 [2, 12] が存在する．しかし，上記の手法はいずれもソースドメインのラベルとターゲットドメインのラベルの両方が利用できる設定であり，本研究のようにターゲットドメインのラベルが利用できない設定には対応していない．

また，Gao らによりトランスダクティブ転移学習の設定におけるランキング学習手法が提案されている [4]．具体的には，まず，特徴空間におけるソースドメインのインスタンスの分布とターゲットドメインのインスタンスの分布を二分する超平面を算出する．次に，ソースドメインのインスタンスのうち超平面に距離が近いものがターゲットドメインの学習に重要であると見なし，ターゲットドメインの学習に与える影響が大きくなるよう重みづける．そして，重みづけたソースドメインのデータを利用してターゲットドメインのモデルを学習することでターゲットドメインのモデルを構築する．これはデータレベルの手法である．上記の手法において効果的な超平面を算出するためには，ターゲットドメインにある程度のクエリ・文書ペアが必要である．一方，本研究の提案手法はドメインからドメイン特徴を 1 つ抽出すれば，予め学習した重み予測モデルで重みの予測を行うことができる．入力としてドメイン特徴が 1 つあれば良いので，ドメイン特徴を仮想的に設定することも可能で，その場合にも一定の効果が見込める．これは，ターゲットドメインにデータが十分存在しない場面や，文書は存在するがクエリは存在しないような場面など，より広い場面で利用することが可能であることを示す．また，本研究の提案手法はアルゴリズムレベルの手法であるため，データレベルの手法とは異なりターゲットドメインで一からモデルを学習する必要がないという利点がある．

2.2 適応的なランキング学習

あらゆるクエリに対して同一のモデルを用いるのではなく，入力されるクエリの種類に対して適応的にモデルの振る舞いを変更することでランキングタスクにおける性能が向上することが報告されている [9]．

これを受けて，予め複数のランキングモデルを構築しておき，入力されるクエリに応じてモデルを選択することでランキングの性能向上を目指す研究がなされている [5, 17, 21]．具体例として，Geng らの手法 [5] を紹介する．この手法では，学習データ中のクエリをクエリ特徴に基づいてクラスタリングした後，クラスタごとにランキングモデルを構築することで複数のランキングモデルを用意する．そして，学習データにない新たなクエリが入力されると，入力されたクエリの特徴からそのクエリが属するクラスタを算出する．入力されたクエリに対して，そのクエリが属すると算出されたクラスタで構築されたモデルを選択してランキングタスクに利用することで，クエリに適応したランキングを行う事が可能となる．本論文はドメインごとにモデルを構築するため，その点で類似しているが，上記の手法は，単一のドメインにおけるクエリの違いを扱っており，異なるドメインにおける効果については検証されていない．更に，上記の手法はクエリごとにモデルを選択するための処理が必要であり，速度が重要となる場面においては有効でない．

また，Macdonald [15] らは回帰木を利用したランキングモデルにクエリ特徴を利用することでモデルの性能が向上すると報告している．この性能の向上は，あるクエリ特徴で木を分岐させた際に分岐以降のサブツリーがある種のクエリに特化して学習され，1 つのモデルを利用していてもクエリの特徴に応じて処理を変更することが可能であることに起因している．しかし，この手法も単一のドメインにおけるクエリの違いを扱っており，異なるドメインにおける効果については検証されていない．

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，提案手法の詳細と，本研究で利用するデータセットを構築する方法について述べる．

3.1 問 題 設 定

本論文では，複数のソースドメインから得た知識を活用して，あるターゲットドメインにおいて性能の良いモデルを構築することを目的とする．ソースドメインとは，ドメイン内のクエリ，文書，その他の情報およびクエリ・文書ペアに対する適合性判定データにアクセスできるドメインである．一方ターゲットドメインとは，ドメイン内のクエリ，文書，その他

の情報にはアクセスできるが，適合性判定データにはアクセスできないドメインである．ここで，複数のソースドメインを $\{m_1, m_2, m_3, ..., m_{n-1}\}$，あるターゲットドメインを m_n としたドメイン集合 $\mathcal{M} = \{m_1, m_2, m_3, ..., m_n\}$ を考える．m_i は 1 つのドメインを表し， クエリ集合 Q_i，文書集合 D_i，適合性判定済みのクエリ・文書ペアの集合 $J_i \subseteq Q_i \times D_i$，適合性判定の基準 $r_i : J_i \to \mathbb{N} \cup \{0\}$ などの要素を含む．上述の通り，ターゲットドメイン m_n には適合性判定済みのクエリ・文書ペアが存在しないため $J_n = \varnothing$ である．これらの要素の他にも，ドメインにはユーザ情報やクエリログなど様々な情報が含まれる．

あるドメイン m_i でランキングタスクを行うためには，与えられたクエリ $q \in Q_i$ に対してドメイン内の文書のランキングを決定する必要がある．特に，本論文では与えられたクエリ $q \in Q_i$ とドメイン内の各文書 $d \in D_i$ との適合性を予測し，文書を適合性の降順に並べ替えることで文書のランキングを決定する．また，検索システムの実用上，適合性判定がされていないクエリに対してランキングを決定する必要がある．そのためには，ある $(q, d) \notin J_i$ に対してその適合性を予測できればよい．適合性の予測には，ドメインごとに最適化された重み $\mathbf{w}_i$ を持つモデル $f : Q \times D \to \mathbb{R}$ を構築して利用する．ここで，Q はあらゆるクエリを含むクエリ集合，D はあらゆる文書を含む文書集合を示す．

また，最適化された重み $\mathbf{w}_i$ を得る代表的な方法として，ランキング学習がある．ランキング学習とは，機械学習を活用して性能の良いランキングモデルを構築する手法である．ランキング学習の代表的な手法にポイントワイズ，ペアワイズ，リストワイズの手法があるが，本論文ではポイントワイズの手法について取り上げる．ポイントワイズの手法では，学習データにおいて入力となるクエリ・文書ペア $(q, d) \in J_i$ からその適合性 $r_i(q, d)$ を上手く予測できるようにモデル f の重み $\mathbf{w}_i$ を学習する．ランキング学習は，関数 $\mathbf{T} : \mathcal{J} \times R \to \mathbb{R}^l$ を用いて，式 1 ように重みを得る工程と定式化することができる．ただし，$\mathcal{J}$ はあらゆる適合性判定済みのクエリ・文書ペアを含む集合のべき集合で，R はあらゆる適合性判定の基準 r を含む集合である．

$$\mathbf{w}_i = \mathbf{T}(J_i, r_i) \tag{1}$$

ソースドメイン m_i $(1 \leqq i < n)$ ではこのようなランキング学習手法を適用し，J_i を利用してモデル f の振る舞いを r_i に近づけるように最適なパラメータ $\mathbf{w}_i$ を学習することで性能の良いランキングモデルを構築することができる．しかし，先に述べたように，ターゲットドメイン m_n には $\mathbf{w}_n$ の学習に必要な適合性判定済みのクエリ・文書ペアの集合 J_n の要素が存在せず，従来のランキング学習手法を適用することができない．本論文では，このようなターゲットドメイン m_n においてもランキングモデルの最適な重み $\mathbf{w}_n$ を得て，性能の良いランキングモデルを構築するという問題を解く．

3.2 重み予測モデル

本論文で提案する重み予測モデルを用いて上記の問題を解くフレームワークを図 1 に示す．主なアイデアは，ソースドメインとターゲットドメインで共通して利用できる要素からドメイン特徴を抽出し，ドメイン特徴からそのドメインのモデルの最適な重みを予測することである．ドメイン特徴とは，各ドメインがどのようなものかを示す特徴である．各ドメインから 1 つずつ抽出し，抽出した特徴が各ドメインで異なることが望ましい．ドメイン特徴としては 文書のトピックなどのドメイン内の文書から抽出することができる特徴，クエリの長さの平均値などのドメイン内のクエリから抽出することができる特徴， クエリと関連する文書の多寡などのドメイン内の文書・クエリペアから抽出することができる特徴など様々な特徴を利用することができる．本研究で利用したドメイン特徴については次節で詳しく述べる．Macdonald らの研究で，データ数が同程度のドメインを用いて実験すると，あるドメインで学習したランキングモデルをそのまま他のドメインで利用した場合ランキングタスクにおける性能が低下すると報告されている [14]．このことから，ランキングモデルの最適な重みはドメインごとに異なることが示唆される．そこで，ドメインの特徴をうまく表現し，ドメイン特徴とランキングモデルの重みの関係性を学習すれば，ドメイン特徴からランキングモデルの最適な重みが予想できると考えた．例えば，ドメイン特徴としてクエリの長さの平均値を利用し，ランキングモデルの特徴である文書の URL に含まれる “/” の数の重みを予測するを場合を考える．クエリが長いことはより具体的な情報要求を表現していることを示し，クエリの長さの平均値が大きいドメインは具体的な情報要求が多いドメインであると期待される．また，URL に含まれる “/” の数は URL の階層構造の深さを示し，URL の階層構造が深い文書は，より具体的な情報や特定のトピックに焦点を当てている傾向にある．上記の傾向から，より具体的な情報要求に関連する文書は URL の階層構造が深い文書である可能性があり，したがって，クエリの長さの平均値が大きいドメインほど文書の URL に含まれる “/” の数に対する重みが大きいといった関係性があると考えられる．

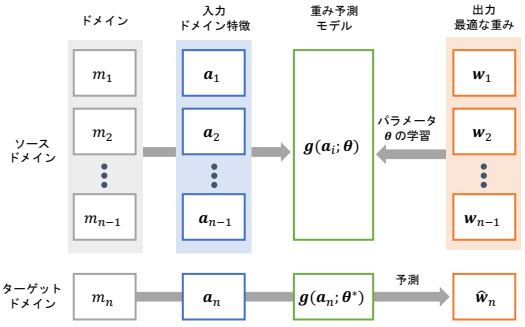

図 1　提案するフレームワークの概要図.

ここからは具体的な手法について説明する．まず，特徴抽出器 $\boldsymbol{\psi} : \mathcal{M} \to \mathbb{R}^m$ を用いてドメイン特徴 $\mathbf{a}_i = \boldsymbol{\psi}(m_i)$ を抽出する．そして，重み予測モデル $\mathbf{g} : \mathbb{R}^m \to \mathbb{R}^l$ を用いて， 式 2 に示すように対象のドメインのモデルの最適な重みの予測値を得る．$\mathbf{g}$ は最適化されたパラメータ $\boldsymbol{\theta}^*$ を持つモデルである．

Algorithm 1 ターゲットドメインでランキングモデルの重みを得るアルゴリズム.

Input: $\mathcal{M}$
Output: $\hat{\mathbf{w}}_n$

```
S ← {}
for i = 1 to n − 1 do
    a_i ← ψ(m_i)
    w_i ← T(J_i, r_i)（式 1）
    S ← S ∪ {(a_i, w_i)}
end for
a_n ← ψ(m_n)
θ* ← arg min_θ L(θ; S)（式 4）
ŵ_n ← g(a_n; θ*)（式 2）
```

$$\hat{\mathbf{w}}_i = \mathbf{g}(\mathbf{a}_i; \boldsymbol{\theta}^*) \tag{2}$$

パラメータ $\boldsymbol{\theta}^*$ は，複数のソースドメインにおいて，$\hat{\mathbf{w}}_i = \mathbf{g}(\mathbf{a}_i; \boldsymbol{\theta})$ が $\mathbf{w}_i$ に近づくように $\boldsymbol{\theta}$ を学習することで得る．ドメイン特徴からランキング学習モデルの重みを予測する問題は，最適な重み $\mathbf{w}_i$ をラベル，ドメイン特徴 $\boldsymbol{a_i}$ をデータの特徴とする式 3 で示すデータセット S を利用して多次元の回帰問題を学習したモデルで解くことができる．

$$S = \{(\mathbf{a}_i, \mathbf{w}_i) | i = 1, 2, ..., n-1\} \tag{3}$$

そのため，パラメータ $\boldsymbol{\theta}^*$ は式 4 のように，多次元の回帰問題に利用できる損失関数 L に対する損失を各ドメインで最小化するような $\boldsymbol{\theta}$ を学習することで決定する．

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}; S) \tag{4}$$

このような重み予測モデルを用いることで，ターゲットドメインにおいても性能の良いランキングモデルを構築することができるようになる．提案手法の全体をまとめたアルゴリズムを Algorithm 1 に示す.

3.3 データセットの分割によるドメインの作成

重み予測モデルを学習するためには複数のソースドメインが必要である．公開されているデータセットを人手で収集し，ソースドメインとして活用する方法もあるが，この方法では学習に十分な数のドメインを用意するのが難しい．そこで，本論文では多種多様なデータを含むデータセットを分割することで，複数のドメインを用意する方法を採用する．

データセットを分割する方法は複数考えられるため，分割を評価するための指標が必要である．先述の通り，ランキングモデルの最適な重みはドメインごとに異なることが確認されている．そこで，以下のような仮定を置く．まず，分割後の各ドメインのデータセットでランキング学習を行い，ドメインごとに最適化されたランキングモデルの重みを得る．この重みが各ドメインで異なれば異なるほど，各ドメインの性質も異なり良い分割であると言える．

この仮定に基づいて分割を評価するための指標として，各ドメインのモデルの各特徴に対するの重みの分散の和（以降，単に**重みの分散の和**と呼ぶ.）が利用できる．ある分割でできた複数のドメインでランキングモデルの重みを学習した結果重みの分散の和が大きいということは，ランキングモデルの重みが各ドメインで大きく異なることを示す．先述の仮定に基づくと，ランキングモデルの重みが各ドメインで大きく異なることは，ドメインの性質が大きく異なることを示す．したがって，重みの分散の和は分割の評価指標として妥当である．より具体的には，l 次元の重みに対して，式 5 が大きいほど良い分割であるとして評価する．ここで，W は分割後の各ドメインで学習したモデルの重み $\mathbf{w}_i$ の集合，w_{ij} は重み $\mathbf{w}_i$ の j 番目の要素を指す．また，$\bar{w}_j$ は各重みの j 番目の要素の平均値である．

$$\sum_{j=1}^{l} \frac{1}{|W|} \sum_{\mathbf{w}_i \in W} (w_{ij} - \bar{w}_j)^2 \tag{5}$$

本研究では，同一のクエリを持つクエリ・文書ペアは同一のドメインに属するという仮定のもと，以下の手順でドメインを分割した．まず，データセットの全クエリ・文書ペア (q, d) から特徴 $\mathbf{x}_{q,d}$ を抽出する．次に，同一のクエリを持つクエリ・文書ペアの特徴を，平均を取ることで 1 つにまとめる．具体的には，$\mathbf{x}_q = \frac{1}{n_q} \sum_{i=1}^{n_q} \mathbf{x}_{q,d_i}$ を各クエリの特徴とする．ここで，n_q はあるクエリ q に対して紐づけられた文書の数を示し，$(q, d_1), (q, d_2), ..., (q, d_{n_q})$ という形でクエリとクエリに紐づけられた文書のペアを表現する．その後，$\mathbf{x}_q$ の集合 $X^{(q)}$ に対して，k-means 法 [16] を用いてクラスタリングを行う．すると，$\mathbf{x}_q$ のクラスタ $X_1^{(q)}, X_2^{(q)}, ..., X_k^{(q)}$ ができる．$\mathbf{x}_q$ はあるクエリを代表する特徴であるため，$\mathbf{x}_q$ のクラスタはクエリのクラスタであると考えることができる．あるクラスタに含まれる全クエリの集合をクエリ集合 Q，$q \in Q$ に紐づく全文書の集合を文書集合 D とすることで，1 つのクラスタから 1 つのドメインを得ることができる．上記の方法では，$\mathbf{x}_{q,d}$ として様々な特徴セットを利用することが可能である．実際に利用した特徴に関しては，次節で詳しく述べる．

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とその統計情報について説明する．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．本実験では，まずデータセットを分割する際に効果的な特徴を明らかにする．次に，重み予測モデルを利用してドメイン適応することにより構築したランキングモデルの性能の有効性を明らかにする．最後に，重み予測モデルに入力するドメイン特徴として有効な特徴を明らかにする．

4.1 データセット

3 節で提案した方法で我々のタスクに適したデータセットを作成するにあたり，分割元となるデータセットとして AOLIA [13] を採用した．AOLIA は，AOL Query Log [19] を利用して作られたデータセットである．AOL Query Log には 2006 年の

3 月から 5 月の間に AOL Search[1]のサーチエンジンに実際にユーザが入力したクエリと，クエリを入力した日時，そのクエリに対してユーザが実際にクリックした文書の URL の情報などが匿名化されて含まれる．AOL Query Log に含まれる文書の URL からタイトルや本文を含む実際の文書の情報を得ることでクエリと文書のテキストが利用できる大規模なデータセットを構築することが可能である．AOLIA は，URL から文書の情報を得る際に Internet Archive[2]を利用しており，URL のリンク切れを回避できるため利用できる文書の量が多くなるほか，クエリが入力された当時の状態に近い文書の情報を利用することができる．また，AOLIA データセットは他のデータセットに比べてクエリを多く含んでおり，分割後のドメインでも多くのクエリを利用できるほか，クエリや文書のテキストが含まれるため，ドメイン特徴として様々な特徴を抽出し利用できるという利点がある．

通常のランキングタスクは主に 1 つの言語を対象として行われるため，AOLIA データセットから英語以外のクエリと文書を取り除いた．ここで，言語の検出にはデータセットを作成した論文でも利用された FastText [8] のモデル [3]を利用し，モデルが英語であると予測した確率が 0.6 未満のクエリと文書を取り除いた．また，タイトルまたは本文が空の文書も取り除いた．

AOLIA にはクエリ・文書ペアに対して人間が下した適合性判定データは含まれておらず，ランキングタスクのデータセットとして活用するには AOLIA に含まれるクエリログからクエリ・文書ペアの適合性を推定する必要がある．しかし，クエリに対して実際に提示されたランキングの情報やクリックされていない文書の情報が含まれていないため，既存のクリックモデルを活用してクエリ・文書ペアの適合性を推定することが難しい．そこで，本研究では以下の方法で適合・不適合の二値からなる適合性を付与した．まず，ユーザが入力したクエリに対してクリックされた文書を適合とみなした．次に，ユーザが入力したクエリに対してフィルタリング後の文書集合中の文書のランキングを算出し，ランキングの上位からクリックされていない文書を 10 件分収集し不適合とみなした．クエリログの収集時と似たランキングの中でクリックされなかった文書は不適合である可能性が高いと考えられるため，ランキングの算出には当時も利用されていたと考えられる単語ベースのランキングモデルである BM25 [24] を用いた．実際に，BM25 を利用して AOL Query Log に含まれるクエリに対する文書の検索結果を再現して利用する例がある [6].

フィルタリング後の文書数は 1,163,269 件で，フィルタリング後の文書集合中に正例を持つクエリは 3,688,945 件あった．その中からランダムに抽出した 360,000 件のクエリに対し，上記の方法で適合文書と不適合文書を決定した結果，4,319,209 件の適合性が付与されたクエリ・文書ペアからなる分割元のデータセットが構築された．構築されたデータセットの統計情報を表 1 に示す.

表 1　作成したデータセットの統計情報.

	データ数	クエリ数	文書数
分割元のデータセット	4,319,209	360,000	868,981
分割後のデータセット（平均値）	8,638	720	1,737

上記の分割元のデータセットを，次節で述べる特徴に基づいて k-means 法で 500 分割し，500 個のドメインを得た．500 分割すると，表 1 にも示したように，1 ドメインあたり平均 8,638 件のクエリ・文書ペアが割り当てられる．ランキング学習のデータセットである LETOR データセット [23] の OHSUMED コーパスには 16,140 件のクエリ・文書ペアが含まれ，通常，5-Fold 交差検証で利用される．5-Fold 交差検証で訓練・検証・評価を行う場合，一般的に，各 Fold で訓練データとして全体のクエリ・文書ペアの件数の 5 分の 3 の約 9,500 件のクエリ・文書ペアを利用することになる．各 Fold の訓練データを 1 つのドメインと捉えれば，分割後のデータセットのクエリ・文書ペアの件数はこの件数に近いため，妥当な件数であると言える．500 個のドメインを作成した後，それらをランダムに分割し，350 個のドメインを重み予測モデルの訓練に用いるソースドメイン，75 個のドメインをハイパーパラメータのチューニングなどの用途で検証に用いるターゲットドメイン，残りの 75 個のドメインを最終的な評価に用いるターゲットドメインとした．このようにして得た 500 個のドメインからなるデータセットを，我々のタスクに適したデータセットとして活用した.

4.2 実験設定

本節では，本実験における実験設定について説明する.

a） 利用した特徴

まず，データセットを分割するための特徴，ドメイン特徴，および，ランキング学習の特徴として利用した特徴について説明する．本実験で利用した特徴の詳細を表 2 に示す．利用した特徴は大きく分けてクエリ特徴（Q），文書特徴（D），クエリ・文書特徴（Q-D）の 3 種類である．これらの特徴はいずれもクエリ・文書ペア (q, d) に対して算出される．以下，表 2 で用いた数式について説明する．クエリ q および文書 d は単語の多重集合として扱い，$q_i \in q$ はあるクエリを構成する各単語を示す．また，各文書 $d \in D$ に含まれる全単語の多重集合を C とする．$df(t)$ は文書頻度で，ある文書集合 D においてある単語 t が出現する文書の件数を示す．$c(t, V)$ は単語の頻度で，ある単語 t が単語の多重集合 V において出現する回数を示す.

クエリ特徴は，検索時のクエリのみから得られる特徴である．表中の IDF (Inverse Document Frequency) は，ある文書集合中である単語が現れる文書の割合の逆数，ICF (Inverse Collection Frequency) は．ある文書集合内の全単語に占めるある単語の割合の逆数である.

文書特徴は，文書のみから得られる特徴である．文書のリーダビリティスコアとしては ARI (Automated Readability Index) [25] を利用した．文書のトピックとしては Open Directory

1：https://search.aol.com/
2：https://archive.org/
3：https://fasttext.cc/docs/en/language-identification.html

表 2　データセットを分割するための特徴，ドメイン特徴，ランキング学習で用いる特徴として利用した特徴.

番号	特徴	種類
1	クエリに含まれるトークンの数（$\|q\|$）	Q
2	$\sum_{q_i \in q} \mathrm{IDF}(q_i)$ ただし，$\mathrm{IDF}(q_i) = \log\left(\frac{\|D\|}{df(q_i)+1}\right)$	Q
3	$\frac{1}{\|q\|}\sum_{q_i \in q} \mathrm{IDF}(q_i)$	Q
4	$\max_{q_i \in q} \mathrm{IDF}(q_i)$	Q
5	$\sqrt{\frac{1}{\|q\|}\sum_{q_i \in q} \left(\mathrm{IDF}(q_i) - \overline{\mathrm{IDF}}\right)^2}$ ただし，$\overline{\mathrm{IDF}}$ は $\mathrm{IDF}(q_i)$ の平均値	Q
6	$\sum_{q_i \in q} \mathrm{ICF}(q_i)$ ただし，$\mathrm{ICF}(q_i) = \log\left(\frac{\|C\|}{c(q_i,C)+1}\right)$	Q
7	$\frac{1}{\|q\|}\sum_{q_i \in q} \mathrm{ICF}(q_i)$	Q
8/9	文書の本文/タイトルに含まれるトークンの数（$\|d\|$）	D
10/11	文書の本文/タイトルにおける $\log\|d\|$	D
12	文書の本文に含まれるセンテンスの数	D
13	文書の本文のリーダビリティスコア	D
14	文書の本文のトピック	D
15	文書の URL のドメインの長さ	D
16	文書の URL のドメインの頻度	D
17	文書の URL に含まれる “/” の数	D
18/19	文書の本文/タイトルにおける $\sum_{q_i \in q \cap d} c(q_i, d)$	Q-D
20/21	文書の本文/タイトルにおける $\sum_{q_i \in q \cap d} \log\left(c(q_i, d) + 1\right)$	Q-D
22/23	文書の本文/タイトルにおける $\sum_{q_i \in q \cap d} \frac{c(q_i,d)}{\|d\|}$	Q-D
24/25	文書の本文/タイトルにおける $\sum_{q_i \in q \cap d} \log\left(\frac{c(q_i,d)}{\|d\|} + 1\right)$	Q-D
26/27	文書の本文/タイトルにおける $\sum_{q_i \in q \cap d} \log\left(\frac{\|D\|}{df(q_i)}\right)$	Q-D
28/29	文書の本文/タイトルにおける $\sum_{q_i \in q \cap d} \log\left(\log\left(\frac{\|D\|}{df(q_i)}\right)\right)$	Q-D
30/31	文書の本文/タイトルにおける $\sum_{q_i \in q \cap d} \log\left(\frac{c(q_i,C)}{\|D\|} + 1\right)$	Q-D
32/33	文書の本文/タイトルにおける $\sum_{q_i \in q \cap d} \log\left(\frac{c(q_i,d)}{\|d\|} \cdot \log\left(\frac{\|D\|}{df(q_i)}\right) + 1\right)$	Q-D
34/35	文書の本文/タイトルにおける $\sum_{q_i \in q \cap d} c(q_i, d) \cdot \log\left(\frac{\|D\|}{df(q_i)}\right)$	Q-D
36/37	文書の本文/タイトルにおける $\sum_{q_i \in q \cap d} \log\left(\frac{c(q_i,d)}{\|d\|} \cdot \frac{\|D\|}{c(q_i,C)} + 1\right)$	Q-D
38/39	文書の本文/タイトルにおける BM25 のスコア（BM25）	Q-D
40/41	文書の本文/タイトルにおける $\log$(BM25)	Q-D
42/43	文書の本文/タイトルにおける LMIR.DIR のスコア	Q-D
44/45	文書の本文/タイトルにおける LMIR.JM のスコア	Q-D
46/47	文書の本文/タイトルにおける LMIR.ABS のスコア	Q-D

Project[4]に含まれるトピックのうち，後述のデータセットで利用された 13 のトピックを用いた．これらのトピックを AOLIA データセット中の各文書に対して推定するために Open Directory Project のデータセット [5]を用いて，テキストが 13 のトピックそれぞれに分類される確率を予測するモデルを学習した．学習に際して，テキストには spaCy のモデル [6]を利用してトークナイズと原形化処理を施し，学習データにおいて原形化されたトークンの集合を語彙として扱った．また，各トークンの TF-IDF を値とする語彙次元のベクトルを各テキストの特徴として抽出した．トピックを予測するモデルにはロジスティック回帰モデルを利用し，文書が各トピックに属する確率を出力するように学習した．URL やハイパーリンクのネットワークに関する文書特徴もランキング学習で良く利用される特徴であるが，AOLIA データセットからはネットワークに関する情報が得られなかったので，URL のみから得られる特徴を独自に 3 つ利用した．

クエリ・文書特徴は，クエリと文書のペアから得られる特徴で，クエリと文書の類似度に関するものが多い．また，ランキング学習の特徴として主に活用されるのはこの特徴である．番号 40 から 47 における LMIR (Language Model for Information Retrieval) は言語モデルによって推定されるクエリ尤度であり，それぞれスムージングの方法が異なる．クエリ・文書特徴の選定と実装は日本語の検索質問ドメインにおけるランキング学習用のデータセットである OpenLiveQ [10] のものを参考にした．上記のデータセットで利用されている 15 のクエリ・文書特徴を文書のタイトルおよび本文の 2 つのフィールドに適用し，30 個の特徴を用意した．

b）　ランキング学習モデル

ランキングモデルが活用する特徴は 1 通りに固定した．ランキングモデルが活用する特徴としては OpenLiveQ データセットを参考に，表 2 のクエリ・文書特徴の全て（番号 18 から 47）と文書特徴のうちトークンの数に関連する特徴（番号 8 から 11）を利用した．また，URL に関する 3 つの特徴（番号 15 から 17）も利用し，合計 37 次元の特徴を用いた．

本実験では，ランキングモデルとして RankLib[7]の Coordinate Ascent を利用した．このモデルは単純な線形のパラメータを持つため，重みの予測が行いやすいと考えた．分割後の各ドメインでランキングモデルを構築する際や，ベースラインとして全ソースドメインのデータを利用してモデルを構築する際は，以下の手順で学習を行った．まず，与えられたドメインのデータセットを 8:1:1 の割合で学習データ，検証データ，評価データに分割した．その後，学習データを用いて上記のモデルを nDCG@10 が最大化するように学習した．1 つのドメインに対してモデルのパラメータを複数回学習し，検証データで最も性能の良かったパラメータを評価に採用した．

c）　重み予測モデル

分割後の 500 個のドメインを利用して式 3 で示すデータセットを得て重み予測モデルの学習に利用した．ドメイン特徴は，データセットを分割する際のクエリの特徴 $\mathbf{x}_q$ としてクエリ・文書ペアの特徴をクエリごとにまとめて平均を取ったように，クエリ・文書ペアの特徴をドメインごとにまとめて平均を取ることによって抽出した．また，重み予測モデルとしてはランダムフォレストによる多次元回帰のモデルを用いた．多層パーセプトロンと線形回帰による多次元回帰のモデルでも実験を行ったが，ランダムフォレストの性能を超えることはなく，かつ結果の傾向も類似していたため，ランダムフォレストによる多次元回帰のモデルの結果のみを記載する．損失関数としては平均二乗誤差を利用し，ハイパーパラメータは検証用のターゲットドメインにおけるデータを用いて nDCG@10 を最大化するように決定した．

d）　評　　価

評価実験では，まず，データセットの分割に用いる特徴セットととして何が有効であるかを明らかにするための実験を行った．具体的には，データセットの分割に用いる特徴セットを変更して実際に複数の分割を行い，評価指標として重みの分散の和を利用することで比較した．

次に，提案手法で予測した重みを活用した**提案モデル**と以下

4：http://www.odp.org/homepage.php

5：https://www.kaggle.com/datasets/lucmichalski/dmoz-bert-multiclass-web-classification-dataset/data

6：https://github.com/explosion/spacy-models/releases/tag/en_core_web_sm-3.7.1

7：https://sourceforge.net/p/lemur/wiki/RankLib/

で紹介する 2 つのモデルのランキングタスクにおける性能を比較した．我々はベースラインとしてドメインの差異を考慮せず大量のデータを用いて学習した**汎用モデル**を利用した．汎用モデルは，学習用の 350 個のソースドメインのデータと検証用の 75 個のターゲットドメインのデータをそれぞれ 1 つにまとめることで学習を行った．提案モデルと汎用モデルが利用するデータの量や種類は同じであるため，性能を左右する大きな要素はドメインの差異を考慮したか否かである．したがって，汎用モデルは提案手法の有効性を示すためのベースラインとしてふさわしい．また，評価用のターゲットドメインで直接学習したモデルを理想的な振る舞いをする**理想モデル**として利用した．この実験では，ドメイン特徴として，データセットの分割に用いた特徴セットと同じものを利用した．

最後に，ドメイン特徴として何が有効かを判別するために，ドメイン特徴として利用する特徴セットを変更した際の重み予測モデルの性能の変化も計測した．

ランキングタスクの評価指標には，一般的に利用される評価指標である nDCG@10，ERR@10，MAP を利用した．より具体的には，75 個の評価用のターゲットドメインにおける各評価データでモデルを評価した場合の各指標の平均値を用いて比較した．各指標の 75 ドメインにおける平均値を算出する際に，汎用モデルは全ての評価ドメインで同じモデルを利用しているが，提案モデルと理想モデルは各評価ドメインごとに特化したモデルを利用している点に注意されたい．

4.3 実験結果

4.3.1 データセットの分割に用いる特徴セットの比較

まず，データセットの分割に用いる特徴のセットを決定するために行った実験の結果を表 3 に示す．Q は表 2 におけるすべてのクエリ特徴，D はすべての文書特徴，Q-D はすべてのクエリ・文書特徴を意味し，複数の特徴セットを組み合わせて利用した場合は，&で示している．

表から分かるように，データセットの分割に用いる特徴セットとしてクエリ・文書特徴を利用した場合に最も重みの分散の和が大きくなっている．3.3 節での仮説に従えば，クエリ・文書特徴を利用した場合に最もドメインの性質が異なるようにデータセットを分割できたと言える．また，クエリ・文書特徴を用いない特徴セットを利用した場合，特徴を用いずランダムに分割した場合よりも重みの分散の和が小さくなっている．特徴を用いずランダムに分割した場合，分割後のドメイン間で性質の差が小さいと考えられるため，ランダムな分割を下回るこれらの特徴セットではうまくデータセットが分割できないと思われる．さらに，クエリ・文書特徴と他の特徴を組み合わせて利用した場合，クエリ・文書特徴のみを利用した場合よりも重みの分散の和が小さくなっている．このことから，クエリ・文書特徴と他の特徴を組み合わせるのは効果的でないと考えられる．先述の通りクエリ特徴や文書特徴がドメインを分割をする上で効果的な特徴でないと思われるため，クエリ・文書特徴と組み合わせるとクエリ・文書特徴を単体で利用したときよりも分割に対する効果が低下してしまうことが原因として考えられる．

表 3　分割に用いた特徴セットごとの重みの分散の和.

分割に用いた特徴セット	重みの分散の和
特徴を用いずランダムに分割	0.0658
Q	0.0620
D	0.0649
Q-D	**0.0710**
Q&D	0.0643
Q&Q-D	0.0674
D&Q-D	0.0682
Q&D&Q-D	0.0690
ランキング学習に用いた特徴	0.0705

最後に，ランキング学習に用いた特徴を利用した場合，文書特徴とクエリ・文書特徴を組み合わせて利用した場合よりも重みの分散の和が大きくなっており，全体でも 2 番目の大きさである事が分かる．このことから，ランキング学習に用いた特徴を利用した場合，文書特徴とクエリ・文書特徴を組み合わせて利用した場合よりも効果的な分割ができると考えられ，全体で比較してもドメインの分割に有効であると考えられる．ランキング学習に用いた特徴は，文書特徴のうち文書の本文に含まれるセンテンス数，本文のリーダビリティスコア，本文のトピックを除いたものに，クエリ・文書特徴を組み合わせたものである．このことから，文書特徴の中でも特にランキング学習で利用していない上記の特徴がドメインの分割に対する効果の低下を引き起こしていると予想される．

以降の実験ではドメインの性質が最も異なるようにデータセットを分割できたと考えられるクエリ・文書特徴を用いた際の分割を採用して分析を行った．

4.3.2 モデル間の性能比較

次に，モデル間の性能を比較するために行った実験結果を表 4 に示す．この結果から，提案手法はドメインの差異を考慮せず学習した汎用モデルよりもすべての指標で高い性能を示した事が分かる．したがって，ターゲットドメインの適合性判定データが利用できない場合に，ドメインの差異を考慮して重みを算出することは効果的であると考えられる．また，提案手法はターゲットドメインで直接学習した理想モデルよりもすべての指標で高い性能を示した．このことから，提案手法は理想モデルよりもランキングタスクにおける性能が高いと考えられる．各テストドメインのモデルの重みを予測する際に，理想モデルは各テストドメインでテストドメイン 1 個分のデータしか活用していないのに対し，提案手法は学習用のソースドメイン 350 個分のデータを活用しており，この活用したデータ数の差が結果に影響したのではないかと予想する．

手法間の性能の差異を統計的に評価するために，nDCG@10 のスコアを対象に二元配置分散分析およびテューキーの範囲検定で検定を行った．二元配置分散分析により，3 つのモデル間に有意差があることが明らかになった（$F(2, 28886) = 59966, p < 0.05$）．その上で，テューキーの範囲検定では，すべてのモデルのペア間に統計的に有意な差が認められた（$p < 0.05$）．

表 4　各モデルの性能比較を行った実験結果.

モデル	nDCG@10	MAP	ERR@10
提案モデル	**0.823**	**0.773**	**0.454**
理想モデル	0.817	0.762	0.449
汎用モデル	0.810	0.759	0.447

表 5　ドメイン特徴を変更した際の提案モデルのスコア.

ドメイン特徴	nDCG@10	MAP	ERR@10
Q	0.817	0.762	0.449
D	0.817	0.762	0.449
Q-D	**0.823**	**0.773**	**0.454**
Q+D	0.820	0.770	0.453
Q+Q-D	0.822	0.771	0.453
D+Q-D	0.821	0.771	0.453
Q+D+Q-D	0.821	0.770	0.452
ランキング学習に用いた特徴	0.822	0.771	0.453

4.3.3　ドメイン特徴として用いる特徴セットの比較

最後に，ドメイン特徴として用いる特徴セットを変更した際の提案手法の性能の変化を表 5 に記す．この結果から，クエリ・文書特徴のみをドメイン特徴として利用したモデルがすべての指標で最も高い性能を示した事が分かる．このことから，クエリ・文書特徴がドメイン特徴として最も効果的であり，最もドメインの特徴を捉えることができていると考えられる．しかし，この結果はドメインの分割に用いる特徴とドメイン特徴が一致していることが影響している可能性もあり，一般的な結果ではないとも考えられる．また，表からは 4.3.1 節の実験結果と同じように，クエリ・文書特徴を用いない特徴セットは他の特徴セットと比べて性能が低いことが読み取れる．更に，クエリ・文書特徴と他の特徴を組み合わせて利用した場合，クエリ・文書特徴のみを利用した場合よりも性能が低下していることも同様である．このことから，ドメインの分割に寄与する特徴とドメイン特徴として活用した際に有効な特徴には似たような傾向があり，クエリ・文書特徴が特に有効であると示唆される．ただし，クエリ・文書特徴を利用している特徴セット間での性能差はドメインを分割する場合よりも小さく，ほとんど数値的な差がない．このことから，ドメイン特徴はデータセットの分割と比べて，あまり効果的でない特徴を追加することから受ける悪影響が小さいと考えられる．また，表 4 と比較すると，どのドメイン特徴を利用した場合であっても，すべての指標で汎用モデルを上回っていることが分かる．このことから，効果の差はあれど，いずれのドメイン特徴もドメインの性質をある程度捉えることが可能であり，重み予測モデルの入力として有効であると考えられる．

手法間の性能の差異を統計的に評価するために，nDCG@10 のスコアを対象に二元配置分散分析およびテューキーの範囲検定で検定を行った．二元配置分散分析により，手法間に有意差が有ることが明らかになった（$F(7, 77031) = 157544, p < 0.05$）．その上で，テューキーの範囲検定では，クエリ・文書特徴を用いた手法とクエリ特徴を用いた手法，ランキング学習に用いた特徴を用いた手法とクエリ特徴を用いた手法の間に統計的に有意な差が認められた（$p < 0.05$）．

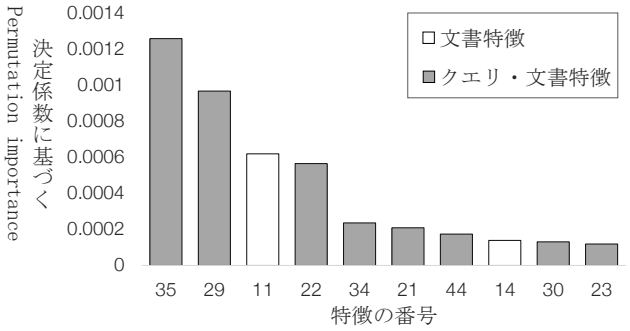

図 2　各ドメイン特徴に対する決定係数に基づく Permutation importance．特徴の番号は表 2 のものを利用している．

4.3.4　個別の特徴の分析

最後に，個別のドメイン特徴のうちどの特徴が有効であるかを分析した．上記の分析を行うために，表 2 のすべての特徴をドメイン特徴として利用した際の，決定係数に基づく Permutation importance を計測した．決定係数は回帰タスクにおける評価指標である．およそ 0 から 1 までの値を取り，この値が大きいとモデルがうまく予測を行える傾向にあることを示す．また，Permutation importance は入力データにおける特定の特徴の値をランダムに並び替えることによって，その特徴がモデルの予測性能にどの程度影響を与えているかを評価する指標である．決定係数に基づく Permutation importance は，重み予測モデルを用いて本来のデータで予測した場合の決定係数と特定の特徴の値を並び替えて予測した場合の決定係数の差を取ることで算出する．Permutation importance が高い特徴ほど値をランダムに並べ替えると予測性能が悪化すること，すなわち，予測において重要な役割を果たしていることを示す．

実験結果を図 2 に示す．ただし，重要度が高かった上位 10 件の結果のみを取り上げる．実験結果から，35 番目の特徴の Permutation Importance が最も高いことが分かる．このことから，35 番目の特徴は重み予測モデルにとって重要な特徴であると言える．35 番目の特徴は文書のタイトルにおける TF-IDF であり，この値をドメインごとにまとめた値に対する重みの重要度が高いことから，ドメインによってクエリと合致するタイトルを持つ文書の多寡が大きく異なる可能性が示唆される．Song らは TF-IDF と同じくクエリと文書の単語の一致度に基づいたスコアである BM25 の値をクエリごとにまとめた値の傾向がドメインによって異なると報告しており [26]，この傾向が本実験でも見られた．その他の結果としては，29 番目の特徴である文書のタイトルにおける IDF の対数の Permutation Importance が 2 番目に高く，ドメインによって入力するクエリの出現頻度（珍しさ）の傾向が大きく異なる可能性がある．また，11 番目の特徴である文書のタイトルに含まれるトークン数の対数の Permutation Importance が 3 番目に高く，ドメインによって文書のタイトルの長さが大きく異なる可能性がある．

最後に，図 2 から重要度の高い特徴にはクエリ・文書特徴が多いことが分かる．本研究では 4.3.1 節に記した通りクエリ・文書特徴の特徴セットに基づいてデータセットを分割することでドメインを作成しており，このことが結果に影響した可能性があると考えられる．

5 ま と め

本論文では，ターゲットドメインの適合性判定データを用いずに転移学習を行うトランスダクティブ転移学習という設定においてランキング学習タスクに取り組み，このタスクにおけるランキング学習モデルのドメイン適応手法を提案した．この手法では，ソースドメインとターゲットドメインで共通して利用できる要素からドメイン特徴を抽出し，ドメイン特徴からターゲットドメインにおける最適な重みを予測することでターゲットドメインにおいて性能の良いランキングモデルを構築することができると考えた．重みの予測は重み予測モデルを学習することで行い，学習に必要な複数のドメインは多種多様なデータを含むデータセットを分割することで用意する方法を採用した．実験では，データセットの分割に用いる特徴としてクエリ・文書特徴が最も効果的であることが判明した．クエリ・文書特徴を用いてデータセットを分割することで得たドメインで提案手法の性能評価をした結果，ドメインの差異を考慮せずに学習した汎用的なモデルよりも提案手法を用いたモデルの性能が高くなることが確認された．また，提案手法が利用するドメイン特徴としても，クエリ・文書特徴が最も効果的であることが明らかになった．今後の課題としては，よりパラメータが複雑なランキングモデルのパラメータを予測したり，より現実に即したドメインで評価を行うなどして提案手法の有効性を確認する必要があると考える．

謝辞 本研究は JSPS 科研費 JP23H03400 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] Olivier Chapelle, Pannagadatta Shivaswamy, Srinivas Vadrevu, Kilian Weinberger, Ya Zhang, and Belle Tseng. Boosted multi-task learning. *Mach. Learn.*, 85(1-2):149–173, 2011.

[2] Depin Chen, Yan Xiong, Jun Yan, Gui-Rong Xue, Gang Wang, and Zheng Chen. Knowledge transfer for cross domain learning to rank. *Inf. Retr. Boston.*, 13(3):236–253, 2010.

[3] Jianfeng Gao, Qiang Wu, Chris Burges, Krysta Svore, Yi Su, Nazan Khan, Shalin Shah, and Hongyan Zhou. Model adaptation via model interpolation and boosting for web search ranking. In *ENMLP*, pages 505–513, 2009.

[4] Wei Gao, Peng Cai, Kam-Fai Wong, and Aoying Zhou. Learning to rank only using training data from related domain. In *SIGIR*, pages 162–169, 2010.

[5] Xiubo Geng, Tie-Yan Liu, Tao Qin, Andrew Arnold, Hang Li, and Heung-Yeung Shum. Query dependent ranking using k-nearest neighbor. In *SIGIR*, pages 115–122, 2008.

[6] Qian Guo, Wei Chen, and Huaiyu Wan. AOL4PS: A Large-scale Data Set for Personalized Search. *Data Intelligence*, 3(4):548–567, 2021.

[7] Malay Haldar, Mustafa Abdool, Liwei He, Dillon Davis, Huiji Gao, and Sanjeev Katariya. Learning to rank diversely at airbnb. In *CIKM*, pages 4609–4615, 2023.

[8] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[9] In-Ho Kang and Gilchang Kim. Query type classification for web document retrieval. In *SIGIR*, pages 64–71, 2003.

[10] Makoto P. Kato and Takehiro Yamamoto. Overview of the ntcir-13 openliveq task. In *NTCIR Conference on Evaluation of Information Access Technologies*, 2017.

[11] Tie-Yan Liu. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, 2009.

[12] Bo Long, Sudarshan Lamkhede, Srinivas Vadrevu, Ya Zhang, and Belle Tseng. A risk minimization framework for domain adaptation. In *CIKM*, pages 1347–1356, 2009.

[13] Sean MacAvaney, Craig Macdonald, and Iadh Ounis. Reproducing personalised session search over the AOL query log. In *ECIR*, pages 627–640, 2022.

[14] Craig Macdonald, B Taner Dinçer, and Iadh Ounis. Transferring learning to rank models for web search. In *ICTIR*, pages 41–50, 2015.

[15] Craig Macdonald, Rodrygo L T Santos, and Iadh Ounis. On the usefulness of query features for learning to rank. In *CIKM*, pages 2559–2562, 2012.

[16] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297, 1967.

[17] Weijian Ni, Yalou Huang, and Maoqiang Xie. A query dependent approach to learning to rank for information retrieval. In *WAIM*, pages 262–269, 2008.

[18] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE TKDE*, 22(10):1345–1359, 2010.

[19] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In *Proceedings of the 1st International Conference on Scalable Information Systems*, page 1–es, 2006.

[20] Rama Kumar Pasumarthi, Sebastian Bruch, Xuanhui Wang, Cheng Li, Michael Bendersky, Marc Najork, Jan Pfeifer, Nadav Golbandi, Rohan Anil, and Stephan Wolf. TF-Ranking: Scalable TensorFlow library for Learning-to-Rank. In *KDD*, pages 2970–2978, 2019.

[21] Jie Peng, Craig Macdonald, and Iadh Ounis. Learning to select a ranking function. In *Advances in Information Retrieval*, pages 114–126, 2010.

[22] Tao Qin and Tie-Yan Liu. Introducing LETOR 4.0 datasets. *CoRR*, abs/1306.2597, 2013.

[23] Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. LETOR: A benchmark collection for research on learning to rank for information retrieval. *Inf. Retr. Boston.*, 13(4):346–374, 2010.

[24] Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, 2009.

[25] RJ Senter and Edgar A Smith. Automated readability index. *AMRL-TR. Aerospace Medical Research Laboratories*, pages 1–11, 1967.

[26] Ruihua Song, Ji-Rong Wen, Shuming Shi, Guomao Xin, Tie-Yan Liu, Tao Qin, Xin Zheng, Jiyu Zhang, Gui-Rong Xue, and Wei-Ying Ma. Microsoft research asia at web track and terabyte track of TREC 2004. *Text Retrieval Conference*, 2004.

[27] A Trotman, Jon Degenhardt, and S Kallumadi. The architecture of ebay search. *eCOM@SIGIR*, 2017.