DBSJ Japanese Journal
Vol. 15-J, Article No. 13, March 2017

同一閲覧・検索意図に合致する Web ページ群の発見と再訪問ページ推薦

RECOMMENDATION OF RE-VISITING PAGES BASED ON FINDING WEB PAGES FOR SAME BROWSING AND SEARCH INTENT

武田 裕介♡ 大島 裕明♢ 田中 克己♠

Yusuke TAKEDA Hiroaki OHSHIMA Katsumi TANAKA

本稿では，ユーザが過去に閲覧したが，再訪問したい **Web** ページを推薦する手法を提案する．**Web** ページの再訪問はしばしば行われるが，失敗することも多い．ユーザが再訪問を行おうとするときには，様々な検索結果や **Web** ページを閲覧することになる．そのときに閲覧されている **Web** ページは，ユーザが再訪問したいと思っているページと同じ情報要求を満たすものであると考えられる．推薦されるべきページは，ユーザが過去に閲覧したページであり，かつ，現在閲覧しているページ群と同じ情報要求を満たすページである．そのようなページを見つけるため，検索クエリの類似性，リンク構造，閲覧時刻の近接性といった，検索・閲覧行動を利用する．実験を行った結果，我々の提案方法が既存の手法よりも良い結果を示した．

This paper proposes a method for recommending Web pages that a user has browsed and wants to re-visit. Re-visiting Web pages is often happened and often failed. When a user tries to find such pages, the user browses search results and many Web pages. The search results and the Web pages have the same information need of the pages the user wants to re-visit. Pages for recommendation can be found from pages that the user has browsed and whose information need is the same as one of the currently browed pages. We leverage the query similarity, link structure, and temporal adjacency of pages for finding such pages. Experimental results show the proposed method overcomes the existing methods.

1 はじめに

近年の Web の発達に伴い，Web には大量の情報が蓄積されてきた．Web 上を検索・閲覧することで様々な情報を得ることが可能になった．以前に閲覧した Web ページを再び閲覧しようとすることも多い．訪問したページの内 44%が再訪問であったり [16]，33%のクエリが再発見のためのもの [19] であるという報告もある．このことからもユーザはページを日常的に再訪問していることが分かる．しかし，以前に発行した検索クエリを忘れたり，どのリンクをたどったかを忘れたりして，以前に閲覧したページの再訪問に失敗することがある．

♡ 非会員　京都大学大学院情報学研究科
takeda@dl.kuis.kyoto-u.ac.jp
♢ 正会員　京都大学大学院情報学研究科
ohshima@dl.kuis.kyoto-u.ac.jp
♠ 正会員　京都大学大学院情報学研究科
tanaka@dl.kuis.kyoto-u.ac.jp

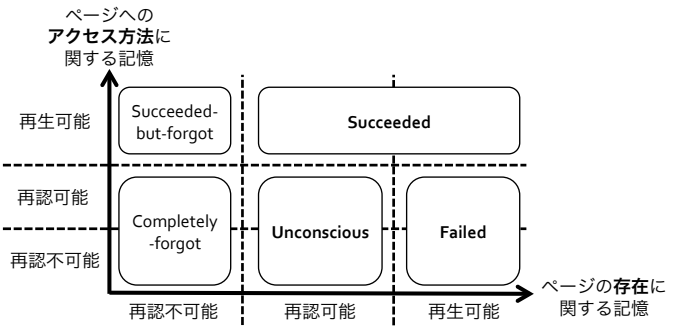


図 1: ページを閲覧したこととページへのアクセス方法の記憶に基づく閲覧ページの分類

ページを再訪問するためには，Web ブラウザのブックマーク機能を利用したり，閲覧履歴を参照したりするという方法がある．しかし，ブックマークしていないページであってももう一度閲覧したいと思うページは存在する．また，ページ単位のブックマークでは，作業中の文脈や閲覧した複数のページ間の関係が失われ，文脈を考慮した再訪問ができないという問題点がある．閲覧履歴を参照するにしても，ユーザは大量のページを閲覧するので，閲覧履歴の中から目的のページを探しだすことは困難であると考えられる．

本研究では，ユーザが過去に閲覧したページの中から，現在の情報要求と同じ情報要求を満たすために閲覧されたページを推定する手法を提案する．これは，現在閲覧しているページの**同位ページ**を推定する手法を提案するといえる．**同位ページ**とは同じ情報要求を満たすために訪問されたページを意味する．このようなページを推薦することで再訪問支援を行う．

例えばシンガポールの公共の場における喫煙の罰則について調べたユーザがいると想定する．このユーザがその 1 週間後に喫煙の罰則についての情報を見直すために，以前に訪れた禁煙の罰則に関するページの再訪問を試みた．しかし，以前とは少し異なる検索クエリを発行してしまい目的のページが検索結果ページに現れなかった．結果，以前に訪問したページの再訪問に失敗した．このとき，再訪問しようとしていたページは以前に「喫煙の罰則について調べる」という情報要求で訪問したページである．現在の情報要求も「喫煙の罰則について調べる」ということである．このようなことから，現在閲覧中のページの同位ページの中に目的のページが存在すると考えた．

ユーザがどのようなページの再訪問を試みて，どのような場合にページの再訪問に失敗するのかということをページに関する記憶に基いて図 1 のように分類した．ページの記憶量は以下の独立する 2 軸によって表すことができると考えた．(1) ページを閲覧したことに関する記憶 : そのページを閲覧したという行為に関する記憶を意味する．ページ内容を覚えているかどうかは関係ない．(2) ページのアクセス方法に関する記憶 : そのページを訪れるために用いたクエリや，どのようなリンクを辿ったかというアクセス方法に関する記憶を意味する．心理学の分野において忘却度合いは “再生” と “再認” を用いてしばしば分類されており [21]，本研究においてもこれに従った．以上の 2 軸を用いて閲覧したページを分類した．(1) **Failed** : ユーザはそのページを覚えており，再訪問を試みることがある．しかし，検索クエリやどのリンクをたどれば良いかといったアクセス方法を忘れてしまっており，再訪問ができずに苦戦する．ページ推薦を切望すると考えられるので本研究ではこのケースに注目した．(2) **Unconscious** : ユーザはそのページを訪問したことすら忘れてしまっている．しかし，そのペー

ジが推薦されれば訪問したことを思い出し（再認する），再訪問すると有用か判断できる．ユーザはページの存在を忘れているのでもし推薦されなくても不自由は感じない．(3) **Succeeded** : ユーザはそのページを再訪問することができる．しかし，時間がかかるためページ推薦による支援を望むことがある．もしページ推薦されなかったとしてもそのページに再訪問することができるので問題はない．Completely forgot ケースや Succeeded-but-forgot ケースに分類されるページをユーザは再認できないので，再訪問のための推薦としては望まないと考えた．

現在閲覧中のページと同じ情報要求で閲覧・検索されたページを推定するために，検索クエリやリンクによるページ遷移，閲覧時刻近接性といったユーザの検索・閲覧行動を利用する．実際の推薦アプリケーションとして使用する際には計算速度も重要である．閲覧ページをセッション毎に分割し，現在と関連するセッションで閲覧されたページから現在閲覧中のページと同じ情報要求で閲覧・検索されたページを推定する．これによって計算速度が早くなるだけでなく，精度の向上が想定される．

提案手法の有用性を確かめるためのユーザ実験から以下のような知見が得られた．(1) 現在閲覧中のページの同位ページの推薦は再訪問におおよそ有用である．(2) リンクの遷移と閲覧時刻近接性を用いた同位ページ推定手法はベースライン手法を上回った．(3) クエリの類似度を用いた同位ページ推定手法はセッション分割と関連セッションの取得が完全に成功したと仮定した場合にベースライン手法を上回った．(4) セッション分割と関連セッションの取得は精度の向上に有用である．

2 関連研究

Jones と Klinkner [8] はクエリを goal，mission，session に分類した．session は一定時間行動がなかった場合に分割されるものである．goal とは極小の情報要求を差し，1 つ以上のクエリによって構成される．mission とは関連する情報要求の集合を差し，1 つ以上の goal によって構成される．

再発見・再検索時のユーザの行動の分析を行った研究は以下のものが挙げられる [15, 3, 17, 1, 22, 6, 23, 18]．Nishimoto ら [15] は，再発見の際の三つの問題点を挙げた．それは最初の検索の際には重要だと思わなかったことが重要になること，閲覧のコンテキストが失われること，発見時と再発見時では知識が異なっていることである．Teevan [18] は以前と同じクエリを入力することができても，検索エンジンはいつも同じ結果を返すとは限らないという問題点を示唆している．Capra [3] は，再発見の種類について Exact，Path，Subset，Move の 4 種類に分類した．Tyler ら [22] は再検索を行う際のクエリと元のクエリでは，再発見を行う際のクエリのほうが良いクエリとなっていることを分析した．Pu ら [17] は発見時と再発見時の検索の行動を比較をした．再発見時にはより多くのクエリ，サーチエンジンを使用するという知見を得ている．Adar ら [1] は，ページが再訪問されるのが以前の訪問からどれくらい時間が経ってからなのかによって，ページが特徴付けられるということを分析した．Eirinaki ら [6] は PageRank とマルコフモデルを用いてユーザの行動を推定し，次あるいはその少し先に閲覧するであろうページを推薦する手法を提案した．Tyler ら [23] は以前のクエリと検索結果を考慮して検索結果を再ランキングする手法を提案した．

再発見のためのシステムに関する研究は以下のものがある [5, 7, 2, 10, 9],[11, 12],[13],[14, 20]．Kawase ら [10] は次に再閲覧するページを推定して表示するシステムを提案した．彼らはブラウザの履歴情報を利用して評価を行った．ユーザが閲覧したページが再訪問のときに，それ以前の閲覧ページを利用して再訪問したページを予測できるかということをログベースで評価した．本研究では，ユーザアンケートにより，再発見に困っている時とその時に推薦して欲しいページという情報を収集して評価を行った．Kawase ら [9] はまた検索エンジン，オンラインブックマーク，オンラインアノテーションという 3 つの再発見手法の違いを評価した．彼らはブックマークとアノテーションが検索を行うよりも有用であるという知見を得た．さらに，アノテーションはページ内容の再認という点でブックマークよりも有用であるという知見を得た．Morris ら [14] は閲覧履歴をトピックとクエリによって階層化し，表示するシステムを提案した．Aula ら [2] はサムネイルとタイトルを表示することがページの再認に有用であるという知見を得た．Teevan ら [20] は視覚的なスニペットを提案し，これがテキストスニペットよりも小さく再訪問に有用であると分析した．Li ら [11] や deng ら [5] はコンテキストを利用して閲覧履歴を改良する手法を提案した．Mackay ら [12] はランドマークを付けることができるブックマーク機能を提案した．ランドマークはページの気になる文章にハイライトを付けることができる機能であり，そのページを再訪問した際にもハイライトは表示される．Web ブラウザの Firefox[1]で使用されている Frecency アルゴリズム [7] はページをどれくらい最近閲覧したか，どの程度頻繁に閲覧したかということを考慮したページのスコアを付けるアルゴリズムである．このスコアはロケーションバーでの推薦においてランキングを付ける際などに利用される．

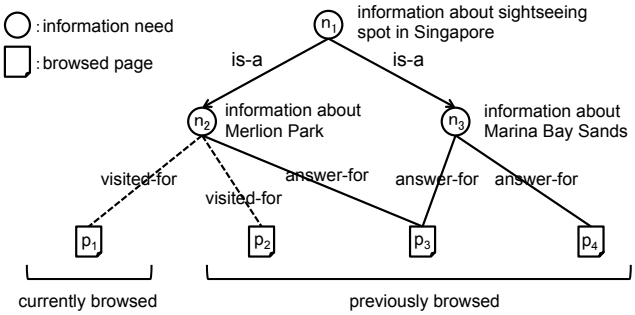


図 2: 情報要求と閲覧ページの関係を表すグラフ

3 再訪問のための同位ページ

3.1 問題定義

本研究において取り組む再発見問題について定義する．URI によって識別されるページ集合 $P = \{p_1, \ldots, p_m\}$ がある．これらはユーザの n 回のページ訪問 $V = \{v_1, \ldots, v_n\}$, $m \leq n$, $\mathrm{page}(v_k) \in P$ の間に閲覧されたページである．また，訪問順序の集合 $I = \{i(v_1), \ldots, i(v_n)\}$ がある．$i(v_k)$ は 1 から始まり，ページ訪問が行われるたびに 1 増加する．このとき，再発見に有用なページ順に $p_i \in P$ をランキングするという問題を解く．入出力は以下の通りとなる．

入力 $P = \{p_1, \cdots, p_m\}$, $V = \{v_1, \ldots, v_n\}$, $I = \{i(v_1), \ldots, i(v_n)\}$,
ブラウザにより取得されるユーザの全ての閲覧・検索行動

出力 $(P, \leq)$, $\leq \subset P \times P$,

$\leq$ は P の全順序を表す．この問題定義は Kawase ら [10] のものと類似している．彼らは次 (n+1) に閲覧するであろうページを推定しているのに対して，本研究は現在推薦してほしいページを推定することを目的としている．上記の問題を解くことで得られる上位 k 件のページを推薦する．

3.2 情報要求と閲覧ページ

本研究では現在閲覧中のページと同じ情報要求の元で閲覧・検索されたページに注目したページ再訪問手法を提案する．なぜ，このようなページを推薦するのが良いのかを説明するために，まずユーザの情報要求と閲覧したページの関係について述べる．ユーザの情報要求と閲覧したページの関係は図 2 のように表される．ユーザの今までの情報要求全ての集合を N，閲覧した全ての Web ページの集合を P とすると，その関係は有向非巡回グラフ $G = (N \cup P, E)$ によって表すことができる．ここでエッジ集合 E は，$E = E_{\mathrm{vis-for}} \cup E_{\mathrm{ans-for}} \cup E_{\mathrm{is-a}}$

[1] http://www.mozilla.jp/firefox/

と表される．それぞれのエッジ集合は以下のように表す．

$$E_{\text{vis-for}} := \{(n,p) \mid n \in N, p \in P,\ n \text{ is visited-for } p\} \subset N \times P$$
$$E_{\text{ans-for}} := \{(n,p) \mid n \in N, p \in P,\ n \text{ is answer-for } p\} \subset E_{\text{vis-for}}$$
$$E_{\text{is-a}} := \{(n_i,n_j) \mid n_i \in N, n_j \in N,\ n_i \text{ is-a } n_j\} \subset N \times N$$

$E_{\text{vis-for}}$ と $E_{\text{ans-for}}$ は無向エッジ集合であり，$E_{\text{is-a}}$ は有向エッジ集合である．ページと情報要求の間には visited-for 関係，answer-for 関係が存在する．ページ p と情報要求 n が visited-for 関係にあるとは，ページ p が情報要求 n を満たすために閲覧されたということを意味する．ページ p と情報要求 n が answer-for 関係にあるとは，ページ p と情報要求 n を満たすことができるということを意味する．

情報要求と情報要求の間には is-a 関係が存在する．情報要求 n_i に対して情報要求 n_j が is-a 関係にあるということは，n_i が n_j のサブクラスであることを意味する．従って，$(n_i,p) \in E_{\text{vis-for}}$ かつ $(n_i,n_j) \in E_{\text{is-a}}$ のとき，$(n_j,p) \in E_{\text{vis-for}}$ となる．これは visited-for 関係を answer-for 関係に置き換えても成立する．

例えば，マーライオンパークの情報を知るという情報要求 n_2 の元で以前にページ p_3 を閲覧し，情報を得たユーザがいるとする．このとき，n_2 と p_3 は answer-for の関係にある．このユーザが 1 週間後にマーライオンパークの情報を確認しようとページ p_1 を閲覧したとする．このとき，n_2 と p_1 は visited-for 関係にある．ページ p_1 が目的のページではなかったとする．このとき，ユーザは n_2 を満たすことのできるページを探している．n_2 と answer-for 関係にあるページ p_3 を推薦すればよい．answer-for 関係を汎化したものが visited-for 関係なので，p_1 と p_3 はどちらも情報要求 n_2 と visited-for 関係にある．

3.3 再訪問のための同位ページ

ページ p_i とページ p_j がある情報要求 n と visited-for 関係にあるとき，p_i と p_j は n において同位であると定義する．同位関係集合 $R_{\text{coordinate}}$ は下記のように表される．

$$R_{\text{coordinate}} = \{(p_i,p_j,n) \mid p_i,p_j \in P, n \in N, (p_i,n) \in E_{\text{vis-for}}, (p_j,n) \in E_{\text{vis-for}}\}$$

同位ページは，ある情報要求 n 上での同値関係とみなすことができ，反射律，対称律，推移律を満たす．本研究の目的は現在閲覧中のページの同位ページを以前に閲覧したページの中から推定することで達成されると考えた．前節で述べた例についても，p_1 と p_3 は同位ページの関係にある．

3.4 実際の同位ページの推定手法

ユーザの情報要求を正しく推定することは困難であり，情報要求の階層構造を求めることは難しい．実際に同位ページを推薦するにあたり，情報要求の階層構造とそれぞれの情報要求の内容には焦点をあてない．例えば，図 2 において，p_1 と p_3 は n_2 において同位である．実際に同位ページを推薦する際には，n_2 が n_1 の is-a 関係や n_2 がマーライオン公園の情報を調べるという情報要求であることを推定しない．p_1 と p_3 が何らかの情報要求の元で同位関係にあるということだけを推定する．

3.5 同位ページを推定するための検索・閲覧行動

以下に述べる 3 つの検索・閲覧行動を用いて同位ページを推定する．これらの検索行動によって閲覧されたページは同位ページである可能性が高いと考えた．図 3 は上記の 3 つの検索・閲覧行動による同位ページの推定方法を表している．図において p_1 を再訪問した際に p_2 もしくは p_3 を推薦することが再訪問に有用であると考えた．

クエリ類似度　情報要求は検索クエリという形で表現されるので，類似するクエリは同じ情報要求を達成するために発行されることが多いと考えられる．類似するクエリを用いて閲覧されたページは同位ページの可能性が高いと考えた．

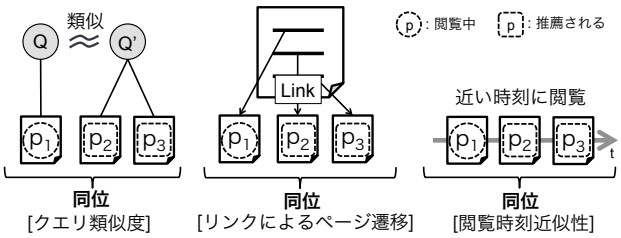


図 3: 同位ページを推定するための検索・閲覧行動

リンクによるページ遷移　あるページの複数のリンクをクリックしてページを訪問することがある．このとき同じ情報要求を達成するために複数のページが開かれることもある．これらのページは同位ページの可能性が高いと考えた．

閲覧時刻近接性　2 つのページを連続的に閲覧したとする．このとき，2 つ目に閲覧したページは 1 つ目のページの記憶が頭のなかに残っている状態で閲覧される．このとき，何らかの情報要求を満たすために頭の中に残っている 1 つ目のページの内容と 2 つ目に閲覧したページの内容を比べることがある．このように，近い時間に閲覧されたページは同位ページである可能性があると考えた．

4 同位ページの推定

現在閲覧中のページの同位ページを以下の流れで推定する．(1) ページ訪問集合をセッションに分割する (2) 現在のセッションに関連するセッションを選ぶ．(3) 選んだセッションの中から現在閲覧中のページの同位ページを推定する．

セッション $S_k \subset V$ はページ訪問集合の部分集合である．ページ訪問集合を分割することでセッション集合 $\mathcal{S} = \{S_1, \cdots, S_l\}$ が生成される．このとき，$S_1 \cup S_2 \cup \cdots \cup S_l = V$ であり，任意の $S_i, S_j \in \mathcal{S}$ において $S_i \cap S_j = \emptyset$ である．

現在閲覧中のページを p^{bro}，以前に閲覧したページを $p_k \in P$ とすると現在閲覧中のページの同位ページらしさ $\text{Coordinate}(p^{\text{bro}}, p_k) \subset [0,1]$ のスコアによって今までに閲覧したページ p_k をランキングすることが目標である．

4.1 セッション分割

閲覧ページをセッション毎に分割する利点は (1) 計算時間の短縮 (2) 精度の向上である．本手法ではユーザがページを閲覧するたびに推薦するページが変化する．従って計算時間は出来る限り短縮したほうが良い．セッション分割をすることによって計算の対象が少なくなるので，計算時間が減少する．現在のセッションに関連しないセッションを取り除くことで，推薦して欲しいページの候補となるページの数が減少する．これは結果的に精度の向上につながると考えられる．

しかし，セッション分割・推定には推薦してほしいページをとりこぼすという欠点がある．推薦して欲しいページを含むセッションを推定できなければ，そのページを推薦できない．関連するセッションを選ぶ際には厳しすぎる閾値を設定することは好ましくないと考えられる．本研究では 30 分以上ユーザの操作がない場合にセッション分割を行った．

4.2 関連するセッションの取得

現在のセッション S_{cur} に関連するセッションを取得するために，セッション毎に特徴ベクトル $f(S_k)$ を作成する．特徴ベクトルはセッションに属するページの単語頻出度の和によって表現する．$f(S_k) = \sum_{p_i \in S_k} \text{tf}(p_i)$ であり，$\text{tf}(p_i)$ はページ p_i の単語頻出度ベク

トルである．このとき，現在のセッションに関連するセッションに属するページ集合 P' はセッション間の類似度によって次のように求められる．

$$P' = \{p | p \in S_k, S_k \in S, S_k \neq S_{\text{cur}}, \\ \text{sim}(f(S_{\text{cur}}), f(S_k)) \geq \theta_{\text{session}}\}.$$

$\text{sim}(f_1, f_2)$ は 2 つの特徴ベクトル f_1，f_2 のコサイン類似度を表す．このようなページ集合 P' の中から現在閲覧中のページの同位ページを推定する．

4.3 同位ページの推定

同位ページを発見するために 2 部グラフである **need-page** グラフ $G = (N \cup P', E)$ を定義する．N は目的ノード集合であり，P' は以前に閲覧したページの集合の内，現在のセッションと関連するセッションとして取得されたセッションに属するページの集合である $(N \cap P' = \emptyset)$．直感的には目的ノードは情報要求を表す．$N = \{n_1, n_2, \cdots, n_l\}$ とすると E は $E \subseteq \{(n, p) | n \in N, p \in P'\}$ と表される無向エッジ集合である．目的ノード n とページノード p の間のエッジは，p と n が visited-for 関係にあるということを示す．もし，ある 2 ページが同じ目的ノードにエッジを貼っていたら，それらのページは同位関係にあると推定される．

この need-page グラフを用いて推定すべきことは現在閲覧中のページの同位ページがどのページであるかである．同位ページの推定のためにページ内容とユーザの閲覧・検索行動に関する以下の 3 つの仮定を立てた．(1) 2 ページ間の同位関係が複数の観点から推定されているなら，これらのページはより同位らしいと推定される．(2) 同位ページの同位ページは同位ページである可能性がある．(3) 類似するページは同位ページである可能性が高い．作成された 2 部グラフに対して Generalized Co-HITS アルゴリズム [4] を適用することでこれらの仮定を反映できる．グラフの作成に閲覧・検索行動から推定される同位関係，アルゴリズムの初期値としてページ内容から推定される同位関係を反映させることで，上記の仮定を満たした推定を行うことが可能である．

4.4 検索・閲覧行動に基づくグラフ構築

クエリ類似度，リンクによるページ遷移，閲覧時刻近接性それぞれ用いて 3 つの **need-page** グラフを構築する．目的ノード集合とエッジ集合について，各グラフについて述べる．ページノードは全てのグラフで現在のセッションに関連するセッションに属する，以前に閲覧したページ集合 P' となっている．

4.4.1 クエリ類似度に関するグラフ

2 組の検索クエリが類似しているとする．それらのクエリからリンクによる遷移を行って閲覧したページは同位ページらしいという仮定に基いてグラフを構築する．Q を現在のセッションに関連するセッションで発行されたクエリの集合とする．それぞれのクエリ $q \in Q$ に対して $\text{ans}_k(q)$ を上位 k 件の検索結果ページのリスト $[p_1, p_2, \cdots, p_k]$ とする．クエリ q に対する特徴ベクトル $f(q)$ は $s(p_1) \cdot s(p_2) \cdot \cdots \cdot s(p_k)$ から生成される tf ベクトルである．$s(p)$ はページのスニペットを指す．グラフ $G = (N \cup P', E)$ は以下のように定義される．

$$\begin{aligned} N &= \{n = (q_i, q_j) | q_i \in Q, q_j \in Q, i < j, \text{sim}(f(q_i), f(q_j)) \geq \theta_{\text{query}}\} \\ E &= \{(n, p) | q_i \in Q, q_j \in Q, i < j, n = (q_i, q_j), p \in \text{ans}^*(q_i) \cup \text{ans}^*(q_j)\} \end{aligned}$$

$\text{ans}(q)$ に対して $\text{ans}^*(q) = \text{ans}(q) \cup \{p' | p'$ は $\text{ans}(q)$ からリンク遷移によって訪問したページ $\}$ とする．後述する実験では $k = 10$ とした．$\theta_{\text{query}} \in [0, 1]$ は二つのクエリが類似しているかどうかを判断する閾値である．

4.4.2 リンクによるページ遷移に関するグラフ

リンクに関するグラフでは同じページからリンクによる遷移を行って閲覧したページは全て同位ページらしいと仮定して，グラフを構築する．従って，目的ノードはリンク元のページ数だけ作成される．同じページからリンクされたページに対して，同じ目的ノードとのエッジを張る．グラフ $G = (N \cup P', E)$ は以下のように表される．

$$\begin{aligned} N &= \{p \in P' | \exists p_i, p \rightarrow_{link} p_i, p_i \in P'\}, \\ E &= \{(n, p) | n \in N, p \in P', n \rightarrow_{link} p\}, \end{aligned}$$

$p_i \rightarrow_{link} p_j$ はページ p_i から p_j へのリンクによるページ遷移があったことを表す．

4.4.3 閲覧時刻近接性に関するグラフ

閲覧時刻近接性に関するグラフでは，あるページの内容に関する記憶が残っている内に閲覧されたページにエッジを張る．$\text{Memory}(p, t)$ をページ $p \in P$ の時刻 t における記憶量，$\text{Browsing}(t)$ を時刻 t において閲覧中のページ，$\text{isBrowsing}(p, t)$ を $\text{Browsing}(t) = p$ ならば 1，そうでないならば 0 を返す関数とする．グラフ $G = (N \cup P', E)$ は以下のように表される．

$$\begin{aligned} N &= \{p | p \in P'\} \\ E &= \{(n, p) | n \in N, p \in P', \text{Memory}(n, t) \cdot \text{isBrowsing}(p, t) \geq 0\} \end{aligned}$$

$\text{Memory}(p, t)$ は $\text{isBrowsing}(p, t)$ が 1 の時に値が増え，$\text{isBrowsing}(p, t)$ が 0 の時に値が減る関数である．人の記憶には限界があるのでこの関数の値には上限値がある．

4.5 Generalized Co-HITS アルゴリズムの適応

上記で構築した各グラフ $G = (N \cup P', E)$ に，Generalized Co-HITS アルゴリズムを適用する．Generalized Co-HITS アルゴリズムは，ページノード $p_i \in P'$ の値 $x_i \in [0, 1]$ と目的ノード $n_i \in N$ の値 $y_i \in [0, 1]$ をエッジにそって伝播していくアルゴリズムである．x_i の値が高いほど p_i が現在閲覧中のページの同位ページらしいと言える．具体的には以下の計算式によって x_i と y_i の値を更新する．

$$x_i = (1 - \lambda_p) x_i^0 + \lambda_p \sum_{n_j \in N} w_{ji}^{np} y_j,$$

$$y_j = (1 - \lambda_n) y_j^0 + \lambda_n \sum_{p_i \in P'} w_{ij}^{pn} x_i.$$

x_i^0 は p_i の初期値，y_j^0 は n_j の初期値であり，$\sum x_i^0 = \sum y_j^0 = 1$ である．w_{ji}^{np} および w_{ij}^{pn} はエッジの重みであり，w_{ji}^{np} は n_j から p_i へのエッジの重みである．また，$\sum_{p_i \in P'} w_{ji}^{np} = \sum_{n_j \in N} w_{ij}^{pn} = 1$ である．$\lambda_p \in [0, 1]$ および $\lambda_n \in [0, 1]$ は初期値の x_i^0，y_j^0 をどの程度重視するかを表すパラメータであり，値が小さいほど，x_i^0，y_j^0 を重要であるとみなす．上記のアルゴリズムは共通の目的ノードに対してエッジが張られたページノードの値が近いものになる．

4.5.1 初期値

ページノードの初期値はそのページが現在閲覧中のページと同じ情報要求の元で開かれたかを表す．ページが現在閲覧中のページと類似していればしているほど同じ情報要求の元で開かれたと考えた．ページの類似度を計算する際にはページの中身だけでなく，タイトルも使用した．これは，Web ページの中には Google Maps などの，主に Javascript で構成されたページがあり，ペー

ジの内容（HTML）がそのページの内容を正しく表していない場合があるからである．ページ p に対する特徴ベクトル $f(p)$ はページのテキストによって生成される tf ベクトルである．$t(p)$ をページ p のタイトルとするとページタイトル t に対する特徴ベクトル $f(t)$ をタイトルの N-Gram ベクトルとした．タイトルはページ本文に比べて短く，ページと同様に形態素解析を行ってベクトルを作ると次元数がとても小さくなるので N-Gram を用いた．実験では $N = 2$ とした．

ページ p と p' 間の類似度 $\mathrm{sim}'(p, p')$ は下記のように計算される．

$$\mathrm{sim}'(p, p') = (1 - a) \cdot \mathrm{sim}(f(p), f(p')) + a \cdot \mathrm{sim}(f(t(p)), f(t(p'))).$$

$a \in [0, 1]$ はページ内容とタイトルのどちらを重視するか決めるパラメータで，大きいほどタイトルを重視する．x_i と y_i の初期値は以下のように計算する．

$$\begin{aligned} x_i^0 &= \frac{\mathrm{sim}'(\mathrm{page}(v_n), p_i)}{\sum_{p_k \in P'} \mathrm{sim}'(\mathrm{page}(v_n), p_k)}, \\ y_i^0 &= \frac{1}{|N|}. \end{aligned}$$

$|N|$ は目的ノードの総数を指す．

4.5.2 エッジの重み

エッジの重みは以下のように計算する．クエリの類似度に関しては，全てのエッジの重みを均等に扱う．エッジを張る際にすでに閾値を用いて枝刈りをしているので，全てのエッジの重みが等しいと考えた．したがって次のように計算される．

$$\begin{aligned} w_{ji}^{np} &= \frac{1}{\mathrm{count}_{\mathrm{ef}}(n_j)}, \\ w_{ij}^{pn} &= \frac{1}{\mathrm{count}_{\mathrm{ef}}(p_i)}. \end{aligned}$$

$\mathrm{count}_{\mathrm{ef}}(p_i)$ は p_i に張られているエッジの本数を指す．

リンクに関するグラフではリンクの回数によって以下のようにエッジの重みを設定する．ページ A からページ B にリンクを 5 回，ページ C にリンクを 1 回，ページ D にリンクを 4 回行ったとする．この時，ページ B に対してページ D のほうがページ C よりも同位ページらしいと考えられる．このような理由から次のようにエッジの重みを設定する．

$$\begin{aligned} w_{ji}^{np} &= \frac{\mathrm{count}(p_j \rightarrow_{link} p_i)}{\sum_{p_k \in P'} \mathrm{count}(p_j \rightarrow_{link} p_k)}, \\ w_{ij}^{pn} &= \frac{\mathrm{count}(p_j \rightarrow_{link} p_i)}{\sum_{p_k \in N} \mathrm{count}(p_k \rightarrow_{link} p_i)}. \end{aligned}$$

$\mathrm{count}(p_i \rightarrow_{link} p_j)$ は，ページ p_i からページ p_j へのリンクによる遷移があった回数を指す．

閲覧時刻近接性に関するグラフでは，あるページの内容の記憶量の多さとページの閲覧時間を考慮してエッジの重みを設定する．例えばページ A を 30 秒閲覧して，次に別のタブに開いていたページ B をタブを切り替えることによって表示した．しかし，ページ B が想定していたページと異なっていた．そこで 3 秒閲覧しただけで別のタブに開いていたページ C をタブを切り替えることによって表示し，30 秒閲覧した．ページ B は少しの間しか閲覧されていないので，ページ A の内容と比較しながら閲覧されたとは考えにくい．ページ A とページ C はどちらも 30 秒間閲覧されており，ページ A の内容を頭の中に置きながらページ C を閲覧したと考えることができる．このようなことを考慮してエッジの重みを以下のように設定する．

$$\begin{aligned} w_{ji}^{np} &= \frac{\int_0^{t_{\mathrm{now}}} \mathrm{Memory}(n_j, t) \cdot \mathrm{isBrowsing}(p_i, t) dt}{\sum_{p_k \in P'} \int_0^{t_{\mathrm{now}}} \mathrm{Memory}(n_j, t) \cdot \mathrm{isBrowsing}(p_k, t) dt}, \\ w_{ij}^{pn} &= \frac{\int_0^{t_{\mathrm{now}}} \mathrm{Memory}(n_j, t) \cdot \mathrm{isBrowsing}(p_i, t) dt}{\sum_{n_k \in N} \int_0^{t_{\mathrm{now}}} \mathrm{Memory}(n_k, t) \cdot \mathrm{isBrowsing}(p_i, t) dt}. \end{aligned}$$

ここで，t_{now} は現在時刻を指す．

4.6 グラフの結合

それぞれのグラフを結合することでそれぞれの閲覧・検索行動を考慮したグラフを生成することができる．グラフを結合する際には概念ノードの数がグラフによって異なるので初期値とエッジの重みの与え方に工夫が必要である．$G = (N^Q \cup P', E^Q)$ をクエリ類似度に関するグラフ，$G = (N^L \cup P', E^L)$ をリンクによるページ遷移に関するグラフ $G = (N^F \cup P', E^F)$ を閲覧時刻近接性に関するグラフとする．このとき，3 つの閲覧・検索行動を考慮したグラフ $G = (N^{\cup} P', E)$ ををを作成する．なお，目的ノード集合は $N = \{n | n \in (N^Q \cup N^L \cup N^F)\}$ であり，エッジ集合は $E = \{(n, p) | (n, p) \in (E^Q \cup E^L \cup E^F)\}$ である．ページノードの初期値はそれぞれのグラフのページノードの初期値と同じである．目的ノード n_i の値 x_i は

$$x_i = \begin{cases} 1/3|N^Q| & (\text{if } n_i \in N^Q) \\ 1/3|N^L| & (\text{if } n_i \in N^L) \\ 1/3|N^F| & (\text{if } n_i \in N^F) \end{cases}$$

と表される．目的ノードからページノードへのエッジの重みはそれぞれのグラフの重みと同じである．ページノードから目的ノードへのエッジの重みは

$$w_{ij}^{pn} = \frac{{w'}_{ij}^{pn}}{\sum_{n_k \in N} {w'}_{ik}^{pn}}$$

となる．ここで ${w'}_{ij}^{pn}$ はグラフ結合前のエッジの重みを表す．

グラフの結合後，Generalized Co-HITS アルゴリズムを適用することで，ページノード $p_i \in P'$ の値としてに入力ページに対する同位ページらしさの値 x_i が求まるこれにより，全ての以前に閲覧したページ $p_k \in P$ と現在閲覧中のページとの同位ページらしさのスコア $\mathrm{Coordinate}(p^{\mathrm{bro}}, p_k)$ が決まる．$\mathrm{Coordinate}(p^{\mathrm{bro}}, p_k)$ は $p_k \in P'$ のとき x_k であり，$p_k \notin P'$ のとき 0 となる．このスコアを用いて閲覧ページをランキングし，推薦するページを決定する．

5 実験

提案手法が有効であるかを検証するためにユーザ実験を行った．目的はいつ，どのページを推薦して欲しいかというデータを取得することである．実験を行うために詳細な閲覧履歴や検索・閲覧行動を記録する Firefox の拡張機能を実装した．

5.1 実験内容

実験は 25 名の参加者を雇い実施した．被験者は，18 歳から 25 歳の学生であり，日本語を母国語としている．被験者が普段から使用しているパソコンに Firefox の拡張機能をインストールした．拡張機能は被験者の閲覧行動を記録するものである．実験は網羅的な情報を Web を閲覧・検索によって収集してメモを作成してもらい，その 1，2，4 週間後にそのメモを再現するためにもう一度 Web を閲覧・検索してもらうというものである．スケジュールは以下の通りである．

メモ再現の 4 週間以上前 ：拡張機能のインストール
被験者が普段から使用しているパソコンに Firefox の拡張機能をインストールする．この時点から，被験者の閲覧・検索履歴の取得が開始する．

メモ再現の 4 週間前 ：メモ作成のタスク 1
それぞれ，4 個のトピックに関して Firefox を用いて情報を集める．各トピック 30 分間の検索・閲覧を行う．各トピックで検索・閲覧した内容に関する紙のメモを作成する．

メモ再現の 2 週間前 ：メモ作成のタスク 2
タスク 1 と異なるトピックで同様に行う．

メモ再現の 1 週間前 ：メモ作成のタスク 3
タスク 1，2 と異なるトピックで同様に行う．関する紙のメモを作成する．

メモ再現日 ：メモ再現タスク
メモ作成タスクで情報を集めたトピックの内 4 トピックに関するメモを各トピック 20 分間で再現する．各トピックに関するメモの再現が終了するごとに，アンケートに答える．

トピックは合計 12 個用意した．12 個のトピックのうち 4 個は再訪問タスクを行わないダミートピックである．残りの 8 個のトピックは再訪問タスクを行うトピックであり，(1) Travel (2) Food (3) Health (4) Purposeless の 4 つのカテゴリに分類される．各カテゴリに 2 個のトピックが用意されている．各カテゴリのトピック例を以下に示す．

Travel 友達と温泉旅行へ行くために，日本の各温泉地の特徴を調べる

Food 美味しいわさびを買うために，各産地ごとのわさびの特徴を調べる

Health 簡単にできる花粉症対策を調べる

Purposeless 変わったデザインのウェブサイトを探す

これらのトピックは単一の答えがあるものではなく，多くの情報を探すことができるものとした．

メモを再現するトピックはメモ作成のタスク 1 から 3 のいずれか 2 つのタスクから，それぞれ 2 トピックずつ選んだ．例えばタスク 1 で行った 4 つのトピックから 2 つ，タスク 3 で行った 4 つのトピックから 2 つの合計 4 トピックという具合である．メモ再現時には，トピックの順序効果が出ないように考慮した．

アンケートはメモ再現時に閲覧しているページに対して，どのページ推薦して欲しいかという情報を得るために行った．また，推薦して欲しいページに関する記憶についても質問した．アンケート内容を以下に示す．

- メモ再現タスクでページを推薦して欲しかった場面を思い浮かべて下さい．そのときの状況について以下の質問に答えてください．
  - どのページを閲覧していましたか．
  - どのページを推薦して欲しかったですか．
  - 推薦して欲しいページのアクセス方法を覚えていますか．
  - 推薦して欲しいページを閲覧したことを覚えていますか．

また，推薦して欲しいページに関する記憶については，1 章で述べたように，アクセス方法に関する記憶と閲覧したことに関する記憶について再認不可，再認可，再生可のどれにあたるか聞いた．

5.2 評価

5.2.1 評価手法

実験では閲覧・検索ログよりユーザ毎に 3.1 で示した問題の入力にあたる，ユーザの閲覧・検索行動が得られる．また，アンケートにより，次の組の集合 $D_{\mathrm{cor}} = \{(v_1^{\mathrm{bro}}, p_1^{\mathrm{rec}}), \cdots\}$ が得られる． (1) $v_i^{\mathrm{bro}} \in V$: ページ p_i^{rec} を推薦して欲しい際に閲覧していたページ．(2) $p_i^{\mathrm{rec}} \in P$: 推薦して欲しいページ．この集合において，あるページを閲覧しているときに推薦して欲しいページの数が 2 つ以上になる場合がある．このとき，推薦してほしいページ全てを推薦することを考えている．評価を行うために D_{cor} から少し変更した集合 $D'_{\mathrm{cor}} = \{(v_1^{\mathrm{bro}}, P_1^{\mathrm{rec}}), \cdots\}$ を下記のように定義する．

$$v_i^{\mathrm{bro}} \in V^{\mathrm{bro}} \quad , \quad P_i^{\mathrm{rec}} = \{p_k^{\mathrm{rec}} | (v_i^{\mathrm{bro}}, p_k^{\mathrm{rec}}) \in D_{\mathrm{cor}}\}$$

ここで，$V^{\mathrm{bro}} = \{v_i | (v_i, p_i) \in D'_{\mathrm{cor}}\}$ は推薦して欲しい際に閲覧していたページのページ訪問集合である．上記の組を正解データとみなした．どのページを推薦するか決定するためにランキングを計算する際には，ページ訪問 v_b を現在閲覧中のページとみなして計算を行った（$\max(I) = i(v_b)$）．閲覧ページのランキングは提案手法と後述するベースライン手法を用いて行う．ページ $p_k^{\mathrm{rec}} \in P^{\mathrm{rec}}$ を正解ページとみなして，このランキングの平均適合率（AP）を求めた．これを正解各組毎に求めその平均値を評価値とした．これは一般に MAP（Mean Average Precision）と言われる評価値である．MAP を用いる理由は，推薦して欲しいページ全てを発見することを目的としているからである．

5.2.2 提案手法の適応

セッション分割と推定に関して，次の 3 つ状況で提案手法を適応した．

1. セッション分割と推定を行わない
2. セッション分割と推定を行う
3. セッション分割と推定が完全に成功した．

以上の各状況において，メモ作成タスクで閲覧したページに対して 4.4 節で示したようにグラフを構築する．グラフはそれぞれの単一の検索・閲覧行動に基づくグラフと全てを結合したグラフの計 4 つを作成した．構築したそれぞれのグラフに Generalized Co-HITS アルゴリズムを適用し，各ページに与えられた値順にランキングを行った．

5.2.3 ベースライン手法

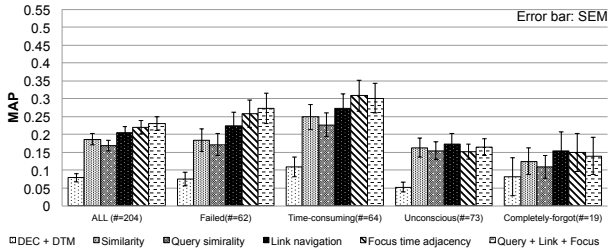
Kawase ら [10] の手法とページ類似度による手法を用いた．

Kawase らはランキング手法と伝播手法を組合せた手法を提案した．ランキング手法は (1) LRU：どれだけ最近閲覧したか (2) MFU：どれだけ頻繁に閲覧しているか (3) DEC：LRU と MFU の組み合わせの 3 つからなる，ページにスコアを付ける手法である．ランキング手法によって計算された値は伝播手法で遷移行列（TM）を使用することによって伝播される．遷移行列は直感的にあるページを閲覧した後にどのページを閲覧しているかを表す．これによってランキング手法によって高い値になったページの後に閲覧されたページのスコアが高くなる．Kawase らは遷移行列をセッション内でのページの閲覧順序を考慮して作成した．これらの手法を試したところ DEC+DTM を用いた手法がもっとも良い結果になった．DTM とは同じセッション内でページ x の後にページ y が閲覧された場合 x から y への値持ち，x と y の訪問順序が離れているほどその値が低くなる遷移行列を作成する手法である．

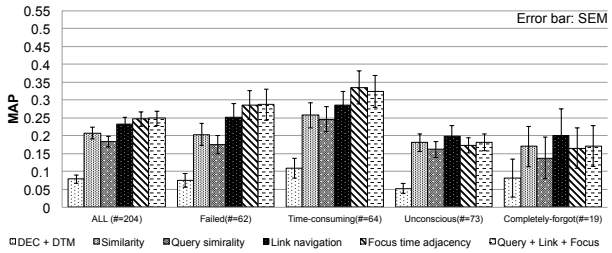
ページ類似度ベースの手法は現在閲覧中のページとの類似度順に閲覧ページをランキングするものである．これは 4.5.1 節で述べた提案手法のアルゴリズムにおける初期値と等価である．

5.2.4 評価結果

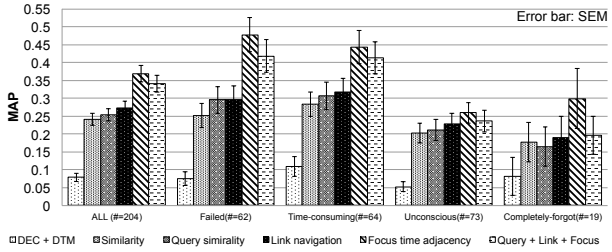
セッション分割と指定なし，セッション分割と推定あり，セッション分割と推定が正確にできたと仮定した場合の 3 種類の MAP に関するグラフを図 4 に示す．それぞれのグラフは推薦して欲しいページを記憶に関する分類にしたがって分類し，それぞれの分類毎に MAP を計算したものを表している．



(a) セッション分割と推定なし



(b) セッション分割と推定あり



(c) セッション分割と推定が完璧にできた場合

図 4: 推薦して欲しいページのタイプ別の MAP．各ブロックで左の 2 つがベースライン手法によるもの．

全体の評価　セッション分割と推定ありの図における一番左の 6 つの MAP について見る．これは，正解データセットの分割を行わずに，全ての正解データセットに対して，提案手法を全て適応した際の MAP を表している．閾値に関するパラメータは $\theta_{\text{session}} = 0.3$, $\theta_{\text{query}} = 0.6$ を用いた．ベースライン手法の MAP と比べて，リンクに関するグラフと閲覧時刻近接性に関するグラフ，全てをマージしたグラフの MAP の方が高い値をとった．しかし，クエリ類似度に関するグラフではベースライン手法を下回った．ユーザは 1ヶ月の間に平均 1000 ページを閲覧しており，240 ものクエリを発行している．この状況で，クエリ類似度に関するグラフを作成すると莫大な数の目的ノードが作成される．巨大すぎるグラフが作成されてしまい，HITS が上手に作用しなかったため，このような結果になったと考えられる．

上述の条件に対してパラメータ λ_p を 0 から 1 まで変化させた

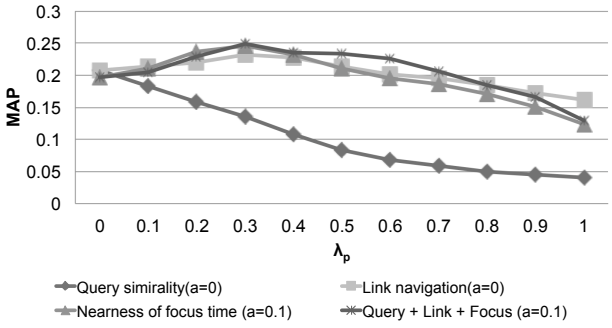


図 5: λ_p を変化させた時の MAP の値

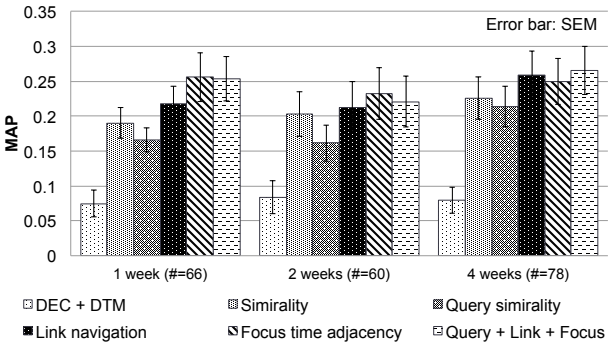


図 6: 再訪問期間別の MAP

際の MAP を図 5 に示す．図 5 においてクエリの類似度を用いたグラフ以外は上に凸なグラフになっている．これは，クエリの類似度を用いたグラフ以外は初期値と伝播手法の両方が重要であるということを意味する．

セッション分割に関する評価

図 4 における 3 つのグラフを比べると，セッションが正確にできればできるほど精度が上がっていることが分かる．つまり，セッション分割を行い，現在のセッションと関連するセッションを選ぶことが計算時間の短縮だけでなく，精度の向上にも大きく寄与していることが分かる．現在のセッション分割手法はとてもナイーブなものであるので，再考の価値はあると考えられる．セッション分割が完全にできた場合，クエリ類似度に関するグラフの手法でもベースラインを上回っている．これは，セッションが完全に分割されていると，作成される目的ノードの数が爆発的に増えないからであると考えられる．

ページの記憶に関する評価　推薦して欲しいページの分類別に MAP を見ると，Failed ケース，Succeeded ケースでは提案手法がベースライン手法を上回っている．しかし，Unconscious ケースや Forgot ケースではベースライン手法と変化があまり見られない．Unconscious ケースや Forgot ケースのページはメモ再現の際にその存在を再生できていないページである．存在を再生できていないページを探そうとすることはないので，そのようなページと類似するようなページにすらたどり着いていない場合も多く存在すると考えられる．存在を再生出来ていな場合，そのページが満たす情報要求も発生しなかったことも多いと考えられる．この場合，現在閲覧中のページの同位ページの中に正解ページが無い場合も多く考えられる．これらのケースの問題解決のためには，情報要求の階層構造や似て異なる度について考える必要がある．

再訪問間隔に関する評価　再訪問の期間別に計算した MAP を図 6 に示す．期間による MAP の目立った違いはないように見受けられる．また，再訪問の期間が 1，2，4 週間のとき，メモ作成タスクで閲覧した 38%，39%，30%のページがメモ再現タスクで再訪問されていた．もし以前に閲覧したページを再訪問できていなくとも，提案手法が目的のページを推定できていることが分かる．

5.3 考察

上述の実験では，提案手法の有用性が概ね示された．本研究には他に以下の問題が残されている．(1) ページ内容が一定時間毎に変わってしまう Web ページが存在する．このようなページを推薦しても，ユーザにとっては有用でない場合が多い．(2) 本研究ではページを推薦するタイミングについて議論していない．一つの方法として，ユーザが表示ページを切り替える度に推薦を行うということが考えられる．つねにページ推薦が行われるのは煩わしいというユーザもいる．また，推薦するタイミングによって推

薦して欲しいページの種類は異なる．適切なタイミングで適切なページを推薦する必要がある．(3) 推薦するページの対象は以前に閲覧したページに限る必要はない．以前に閲覧していないページを推薦するために，他のユーザの閲覧行動における同位関係を使用するということが考えられる．

6 おわりに

本論文では，再訪問に有用なページが現在閲覧中のページの同位ページであるを推薦する手法を提案した．我々は，再訪問に失敗しているユーザに注目した．現在閲覧中のページの同位ページを発見するために，まず閲覧ページをセッションに分割した．次に現在のセッションに関連するセッションを選び，選んだセッションに含まれるページの中から現在閲覧中のページの同位ページを推定した．同位ページの推定には，クエリ類似度，リンクによるページ遷移，閲覧時刻近接性という 3 つの閲覧・検索行動を使用した．実験の結果，提案手法がある程度有用であることがわかった．リンクによるページ遷移と閲覧時刻近接性を用いた手法はベースライン手法を上回った．クエリ類似度を用いた手法はセッション分割が完全になされたと仮定した場合には有効に働いた．セッション分割は計算時間の短縮と精度の向上に有用であることが分かった．

[謝辞]

本研究の一部は，JSPS 科研費（課題番号 16H02906,15H01718，24680008）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] E. Adar, J. Teevan, and S. T. Dumais. Large scale analysis of web revisitation patterns. In *CHI2008*, pages 1197–1206, 2008.

[2] A. Aula, R. M. Khan, Z. Guan, P. Fontes, and P. Hong. A comparison of visual and textual page previews in judging the helpfulness of web pages. In *WWW2010*, pages 51–60, 2010.

[3] R. G. Capra III. *An investigation of finding and refinding information on the web*. PhD thesis, Virginia Polytechnic Institute and State University, 2006.

[4] H. Deng, M. R. Lyu, and I. King. A generalized co-hits algorithm and its application to bipartite graphs. In *KDD2009*, pages 239–248, 2009.

[5] T. Deng, L. Zhao, and L. Feng. Enhancing web revisitation by contextual keywords. In *ICWE2013*, pages 323–337, 2013.

[6] M. Eirinaki, M. Vazirgiannis, and D. Kapogiannis. Web path recommendations based on page ranking and markov models. In *WIDM2005*, pages 2–9, 2005.

[7] https://developer.mdozilla.org/en-US/docs/Mozilla/Tech/Placesd/Frecency_algorithm.

[8] R. Jones and K. L. Klinkner. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *CIKM2008*, pages 699–708, 2008.

[9] R. Kawase, G. Papadakis, E. Herder, and W. Nejdl. The impact of bookmarks and annotations on refinding information. In *HT2010*, pages 29–34, 2010.

[10] R. Kawase, G. Papadakis, E. Herder, and W. Nejdl. Beyond the usual suspects: context-aware revisitation support. In *HT2011*, pages 27–36, 2011.

[11] I. Li, J. Nichols, T. Lau, C. Drews, and A. Cypher. Here's what i did: sharing and reusing web activity with actionshot. In *CHI2010*, pages 723–732, 2010.

[12] B. MacKay, M. Kellar, and C. Watters. An evaluation of landmarks for re-finding information on the web. In *CHI'05 extended abstracts*, pages 1609–1612, 2005.

[13] M. Mayer. Web history tools and revisitation support: A survey of existing approaches and directions. *Foundations and Trends in Human-Computer Interaction*, 2(3):173–278, 2009.

[14] D. Morris, M. Ringel Morris, and G. Venolia. Searchbar: a search-centric web history for task resumption and information re-finding. In *CHI2008*, pages 1207–1216, 2008.

[15] I. Nishimoto and M. Toda. Process-recollective refinding on the web. In *WI2006*, pages 883–892, 2006.

[16] H. Obendorf, H. Weinreich, E. Herder, and M. Mayer. Web page revisitation revisited: implications of a long-term click-stream study of browser usage. In *CHI2007*, pages 597–606, 2007.

[17] H.-T. Pu and X.-Y. Jiang. A comparison of how users search on web finding and re-finding tasks. In *2011 iConference*, pages 446–451, 2011.

[18] J. Teevan. The re: search engine: simultaneous support for finding and re-finding. In *UIST2007*, pages 23–32, 2007.

[19] J. Teevan, E. Adar, R. Jones, and M. A. Potts. Information re-retrieval: repeat queries in yahoo's logs. In *SIGIR2007*, pages 151–158, 2007.

[20] J. Teevan, E. Cutrell, D. Fisher, S. M. Drucker, G. Ramos, P. André, and C. Hu. Visual snippets: summarizing web pages for search and revisitation. In *CHI*, pages 2023–2032, 2009.

[21] E. Tulving and F. I. Craik. *The Oxford handbook of memory*, pages 8–9. Oxford University Press, 2000.

[22] S. K. Tyler and J. Teevan. Large scale query log analysis of re-finding. In *WSDM2010*, pages 191–200, 2010.

[23] S. K. Tyler, J. Wang, and Y. Zhang. Utilizing re-finding for personalized information retrieval. In *CIKM2010*, pages 1469–1472, 2010.

武田 裕介 Yusuke TAKEDA
2014 年京都大学工学部情報学科卒業．2016 年同大大学院情報学研究科社会情報学専攻博士前期課程修了．同年，ヤフー株式会社に入社．以来，情報検索の研究開発に従事．

大島 裕明 Hiroaki OHSHIMA
京都大学大学院情報学研究科社会情報学専攻特定准教授．2007 年京都大学大学院情報学研究科博士後期課程修了．博士（情報学）．主に情報検索，ウェブマイニング，デザイン学の研究に従事．情報処理学会，電子情報通信学会，日本データベース学会，ACM 各会員．

田中 克己 Katsumi TANAKA
京都大学大学院情報学研究科社会情報学専攻教授．1976 年京都大学大学院博士前期課程修了．京大工博．主にデータベース，マルチメディアコンテンツ処理，ウェブ検索の研究に従事．IEEE Computer Society，ACM，人工知能学会，日本ソフトウェア科学会，情報処理学会，日本データベース学会各会員．