

Web 閲覧・検索情報の構造化とその再利用

武田 裕介[†] 大島 裕明^{††} 田中 克己^{††}

[†] 京都大学工学部情報学科 〒 606-8501 京都府京都市左京区吉田本町

^{††} 京都大学大学院情報学研究科 〒 606-8501 京都府京都市左京区吉田本町

E-mail: †{takeda,ohshima,tanaka}@dl.kuis.kyoto-u.ac.jp

あらまし 本論文では、ユーザの Web 閲覧・検索行動を構造化して保存し、再利用する手法を提案する。タブブラウザの普及により、Web ページの閲覧を行う際に、同時に複数のページを開くことがある。また、以前に閲覧したページを再度閲覧することも多い。このような閲覧情報の再発見のためには、ブックマークなどの手法があるが、ページ単位でのブックマークでは、作業中の文脈やアクセスした複数の Web ページ間の関係が失われるという問題点がある。本研究では、タスクの再開をスムーズにするという目的のもと、ユーザが閲覧したページに対して再閲覧ページを自動推定し、保存する手法を提案する。あるページに対する再閲覧ページとは、ユーザが閲覧したページの内、そのページとの関連が強く、タスクの再開に有用なページを指す。また、現在閲覧しているページに応じて再閲覧ページを表示する手法を提案する。

キーワード 履歴、ブックマーク、再検索、再発見

1. はじめに

近年、タブブラウザの普及により、ユーザは複数のページを同時に開きながら、Web ページの閲覧や Web 検索を行うようになった。Web ページを閲覧しているといつの間にか大量のタブを開いてしまっていることがある。例えば、デジタルカメラの購入を検討するというタスクのために Web 検索を行い、ページを閲覧するとする。ユーザはデジタルカメラの性能比較のページを開く。サイバーショットのページと LUMIX のページを異なるタブで開き、互いに比較しながら閲覧する。価格を調べるために、価格.com のページを開く。また、Gmail や Outlook などのメールサービスのページや、サイボウズや Google カレンダーなどのスケジュール機能のページはよく参照するので長い間開かれたままになっている。デジタルカメラについて調べている間に他のタスクを行う必要がでてきて、デジタルカメラとは全く関係のないページを閲覧することもある。このように大量のタブやページを開きながら、Web ページを閲覧することが一般的に行われる。

また、あるタスクが終了したら、最終的にはそのタスクに関連するページを閉じることがある。しかし、終了したタスクでも、タスクを再開する必要があることがある。先ほどの例では、予算が変わってしまいデジタルカメラの購入を再検討する必要がでてくるといった場合である。サイバーショットのページを開いたとする。このとき、以前にサイバーショットのページと同時に比較しながら閲覧していた LUMIX のページを開きたいと思うこともあるだろう。しかし、以下のような問題が生じることがあると考えられる。

- 一から LUMIX のページを再検索し、以前の検索行動を繰り返してしまい、時間がかかる。
- どのように検索し、どのようにリンクをたどれば良いか覚えておらず、ページの再発見ができない。

閲覧した Web ページの再発見のためにはブックマークを利用したり、各 Web ブラウザが備えている履歴機能などなどを利用することが考えられる。しかし、ブックマークしていないページでも、もう一度閲覧したいと思うページはあるだろう。また、ユーザは大量のページを閲覧するので、履歴の中から目的のページを探し出すのは時間がかかることが多い。また、ブックマークや履歴機能はページ単位での情報の保存を行う。しかし、ページ単位の保存では作業中の文脈や、アクセスした複数の Web ページ間の関係が失われるという問題点がある。

そこで本研究では、ユーザが閲覧したページを単一で保存するのではなく、同時に開いていたという情報や、タブを切り替えた情報なども保存し、それらの中からあるページに対して関連の強いページがどれかを推定する手法を提案する。また、現在閲覧しているページが以下の場合に、以前に閲覧したページに対して推定した、関連の強いページを提示する。

- 以前に閲覧したことのあるページである
- 検索結果ページで、検索結果に以前閲覧したことのあるページが含まれている

先ほどの例で言えば、タスクの再開時にサイバーショットのページさえ発見できれば、LUMIX のページやデジタルカメラの性能比較のページが提示される。現在表示しているページに応じて、以前に閲覧した関連するページが提示されるので、ユーザは以前に閲覧したページに素早くアクセスできるようになる。ブックマーク、履歴機能のようにページを単一で保存するのではなく、ページに関連するページ集合を共に保存するということから、これを構造化保存と呼ぶ。

本研究の目的は、タスクの再開をスムーズにすることである。タスクの再開をスムーズにする上で有用であるページとして

- (1) 共閲覧ページ
- (2) 共検索ページ

を考慮する。共閲覧ページとは、あるページに対して、その

ページを閲覧している際に同時に参照・比較しながら閲覧しているとユーザが感じているページを指す。タスクの再開時に以前と同じように複数のタブを開き、互いに参照・比較しながら閲覧することはあるだろう。その状況を素早く再現できるという点で、共閲覧ページを表示することはタスクの再開に有用であると考えた。共検索ページとは、あるページに対して、そのページを開くに至った初めの情報要求が類似しているページを指す。ユーザは情報要求を検索エンジンに対してクエリという形で表すと言われている [6]。タスクの再開時に以前と同じように複数のクエリを順番に用いて何度も検索することはあるだろう。共検索ページを表示することで、何度も検索を行う必要がなくなり、タスクの再開がスムーズになると考えた。また、共閲覧ページや共検索ページを総じて再閲覧ページと呼ぶ。再閲覧ページは再閲覧するとタスクの再開に有用であるページを指す。共閲覧ページや共検索ページはタスクの再開をスムーズにする上で有用であるので、再閲覧ページの一部であると考えられる。

2. 関連研究

本研究に関連する研究としては、タブブラウザに関する研究や、ブックマークや履歴保存に関する研究や、再検索や再発見に関する研究が挙げられる。

タブブラウザに関する研究 [1], [5], [12] として以下のものがある。星加 [12] はタブブラウザにおいて、タブの生成、フォーカスの変更、タブの消去、ページの移動を監視し、タブブラウジングにおける、タブ操作の特徴的な 4 つのパターンを分析し、オートマトンを用いて抽出した。Dubroy ら [1] は、タブブラウザは複数のウィンドウを開く必要のある昔のブラウザと比べて「戻る」機能の使用が少なくなっていることなどを分析した。タブブラウザでは、容易に複数のページを開くことができるので、情報の整理は必要であり本研究の背景としてもその必要性は存在する。

複数の文書間、アプリケーション間の関連を自動的に保存する研究 [8], [9], [11], [13] としては、以下のものがある。内藤ら [13] は、Web ページやオフィス文書など、複数文書を同時に閲覧する際の文書間の関連について 6 つの属性に分け、各文書間の関連を求め可視化した。暦本 [8] は、デスクトップ環境を時間軸に記録していき、必要なときに当時の環境を回復できる *Time-Machine Computing* を提案した。渡辺ら [11] は、ファイルのテキスト類似性や作成時間、共起時間からファイルの関係性を抽出し、多次元クラスタリングを行い、ファイルの検索を行う *FileSearchCube* システムを提案した。Schmidt ら [9] は、閲覧履歴を視覚化する手法を提案し、再検索に有用であることを示した。いずれの研究においても、二つのオブジェクト間の関連の一つとして共起関係を用いている。本手法でも、共起関係を 1 つの指標として用いる。

ブックマークに関する研究 [2], [3] として、新しいブックマーク機能を提案したものがある。日野ら [2] は、*Small Knowledge* と呼ばれるブックマークの概念を提案した。これはユーザが指定した範囲の画像を用いてページをブックマーキングし、

ボードに自由に貼付けるものである。Hupp ら [3] は、*Smart Bookmarks* と呼ばれるブックマークを提案した。これは、ページ単一のブックマークではなく、どのようにしてそのページが開かれたかを自動的に推定してその情報と共に保存するものである。これらの研究のように、ページ単位での保存でないということは本研究においても重要である。ブックマーク機能はユーザが後で閲覧することがあると思ったページを保存することができる。しかし、保存しようと思っていないようなページでももう一度閲覧したいと思うページは存在する。

再検索、再発見に関する研究 [4], [7], [10] としては、以下のものがある。Tyler ら [10] は、再検索を行う際のクエリと元のクエリでは再検索を行う際のクエリの方がより良いクエリとなっていることを提示した。西本ら [7] は、あるページを再訪問しようとする際に、その手がかりから目的のページを見つけ出す手法を提案した。Kawase ら [4] は、再検索をする際に検索エンジンとソーシャルブックマークとソーシャルな付箋機能をそれぞれ用いたとき、付箋機能が一番短時間で再検索を行うことができると分析した。再検索、再発見を行う際にブックマークや履歴機能は有効である。そこで本研究では、ユーザがそれらの機能を使うことを意識せずに使用できるインタフェースを提示する。

3. 再閲覧ページ

あるページに対する再閲覧ページとは、ユーザが閲覧したページの内、そのページと関連が強く、そのページに関するタスクを行う際に再閲覧すると有用であるページを指す。再閲覧ページとしては様々なページが考えられるが、本研究では共閲覧ページと共検索ページを提案する。

3.1 共閲覧ページ

再閲覧ページの 1 つとして共閲覧ページを提案する。あるページに対する共閲覧ページとは、そのページを閲覧している際に互いに参照・比較しながら閲覧しているとユーザが感じるページを指す。例えば京都について詳しくないユーザが京都観光の計画を立てるためにタブブラウザを用いて図 1 のように複数のタブを開いて閲覧・検索行動をしたとする。このときユーザは、京都の観光マップのページ p_1 、京都の観光名所を紹介しているページ p_2 、各観光名所に関するページ p_3 、京都の飲食店に関するページ p_4 などを開く。ユーザはページ p_1 と他のページを交互に切り替えて、各観光名所の地理的な位置関係を確かめながら計画を立てた。ページ p_1 とページ p_2 はよく、フォーカスが切り替わっている。したがってページ p_1 はページ p_2 にとっての共閲覧ページだと言える。

あるページの共閲覧ページは、そのページと互いに参照・比較しながら閲覧しているとユーザが感じるページである。実際に参照・比較しながら閲覧したページだけを指すものではない。先ほどの例で言うと、ページ p_2 から京都の観光ルートについてのページ p_5 を開いて閲覧したとする。このとき、 p_1 と p_5 の間ではフォーカスが切り替わらなかった。ユーザはこのタスクを行っている際に、京都観光マップのページ p_1 と他のページをよく参照・比較しながら閲覧している。したがってページ p_5

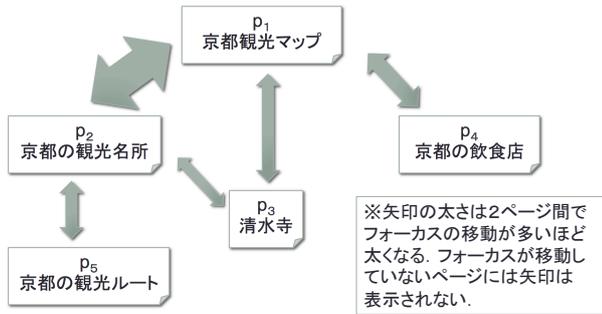


図 1 ページ $p_1 \sim p_5$ までのフォーカスの切り替え例

とも、参照・比較しながら閲覧したと感ずることもある。このとき、ページ p_5 にとって p_1 は共閲覧ページとなる。

複数のタブでページを同時に開き、それぞれを参照・比較しながら閲覧を行うことはよくあるだろう。タスクを再開する際に以前と同じように複数のタブでページを開き、以前と同じように互いに参照・比較しながら閲覧を行うということもよくあるだろう。複数のページをそれぞれ一から開き直していたのでは効率が悪い。共閲覧ページを推定し、表示することで、複数のタブでページを開いているという状況を素早く再現できるようになり、タスクの再開がスムーズになると考えた。

3.2 共検索ページ

再閲覧ページのもう 1 つの要素として共検索ページを提案する。あるページに対する共検索ページとは、そのページを開くに至った初めの情報要求が類似したページを指す。ユーザは情報要求を検索エンジンに対してクエリという形で表すと言われている [6]。例えば、デジタルカメラの購入を検討しようと「デジタルカメラ おすすめ」というクエリで検索を行い、おすすめのデジタルカメラについて記述されたページ p_6 を閲覧したとする。しかし、このページがユーザの要求を満たすものではなく、「デジタルカメラ 購入 ポイント」というクエリで検索し、要求を満たすページ p_7 が得られたとする。このとき、ページ p_6 とページ p_7 は開くに至った初めの情報要求が類似しているため共検索ページとなる。また、それぞれのページを開くために用いたクエリは類似している。ページ p_6 とページ p_7 は互いに参照・比較していないので共閲覧ページではない。

ユーザはタスクの再開時にページ p_7 を開こうと思ったとき、「デジタルカメラ おすすめ」というクエリで検索を行いページ p_6 を開くかもしれない。このときに、共検索ページであるページ p_7 を表示すれば、ユーザは何度も検索する必要がなく、目的のページを開くことが可能になる。もし、「デジタルカメラ 購入 ポイント」というクエリを忘れていた場合でもページ p_6 さえ見つけることができれば、ページ p_7 を探し出すことができる。

また、共検索ページはページを開くに至った初めの情報要求が類似しているページであるので、ページ内容が類似しているページだけを指すのではない。例えば、京都観光について調べようとして「京都観光」というクエリで検索を行い、観光案内のページ p_8 を開いたとする。そのページからリンクを複数回たどって、観光におすすめのデジタルカメラに関するページ p_9 を開いたとする。このとき、ページ p_8 とページ p_9 は開くに至っ

た初めの情報要求は、京都観光について調べるということで類似しているため共検索ページとなる。ページ p_9 を再検索しようとした時に「京都観光」というクエリを用いて検索を行ったのは覚えている。しかし、どうリンクをたどればページ p_9 を開くことができるか覚えていないということはあるだろう。このようにクエリは覚えているがどのようにリンクをたどればよいか覚えていないといった場合に、共検索ページを表示すれば再発見が困難なページを発見することができると思った。

4. 再閲覧ページの推定手法と表示

再閲覧ページ（共閲覧ページ、共検索ページ）を推定するために以下の 4 つの尺度を考慮する。

- (1) 共閲覧度
- (2) 共検索度
- (3) ページ重要度
- (4) 時間近接度

まず、あるページに対する再閲覧ページの候補ページ（周辺履歴ページ）を提案する。あるページに対する周辺履歴ページとはそのページが開かれていた時に同時に開かれていたページ（そのページを閲覧している途中で開かれたページも含む）や、そのページを開いた時刻と近い時刻に開かれた k 件のページ全てを指す。再閲覧ページを推定する際にその対象を全ての閲覧ページとしても良いが、計算コストが高くなる。また、全く違う時間に見ているページ間の関係を保存しても、関連の強いもの、あるいはタスクの再開に役立つものは少ないだろうと考えた。

4.1 共閲覧度

ページ p_1 のページ p_2 に対する共閲覧度は、ページ p_1 にとってページ p_2 がどれくらい共閲覧ページらしいかを表す。本研究では、共閲覧度をページの共起度を用いて計算する。ページ p_1 のページ p_2 に対する共起度は、ページ p_1 とページ p_2 間の共起度が大きいほど値が大きくなり、ページ p_1 の周辺履歴ページの各ページとページ p_2 間の共起度が大きいほど値が大きくなる。すなわち、以下のように計算する。

$$\text{CoBr}(p_1, p_2) = \alpha \cdot \text{CoOc}(p_1, p_2) + (1 - \alpha) \sum_{p_3 \in \text{Can}(p_1)} \text{CoOc}(p_2, p_3)$$

ここで $\text{CoBr}(p_1, p_2)$ はページ p_1 のページ p_2 に対する共閲覧度を指し、 $\text{CoOc}(p_1, p_2)$ はページ p_1 とページ p_2 間のページ共起度を表す。なお、ページ共起度の計算方法は後述する。また、 $\text{Can}(p_1)$ はページ p_1 の周辺履歴ページ全体を指し、 N は周辺履歴ページの数に指す。 α は $0 \leq \alpha \leq 1$ の係数である。ここで、式の第一項は 2 ページ間を実際に比較・参照したかを表す。第二項は、ユーザが 2 ページ間を比較・参照したと感ずたかを表す。これは周辺履歴ページ内でよく比較・参照されているページは、実際に比較・参照されていない場合でも、ユーザは比較・参照していると感じるという仮定に基づく。

ページ共起度

ページ共起度は単純には 2 ページ間でフォーカスが移った回数によって求めることができる。しかし、この求め方では、フォーカスの切り替えを間違えた場合に正しく共起度を付与で

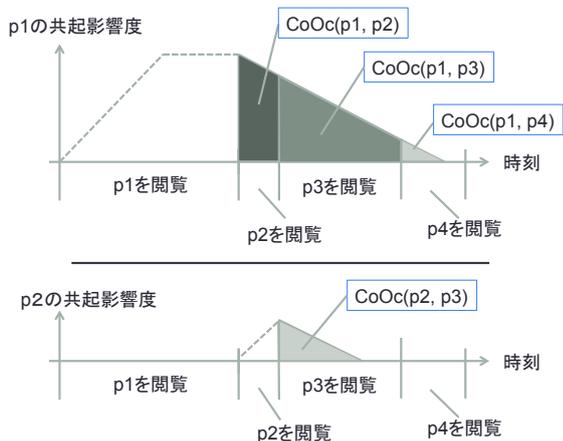


図 2 時間に基づく共起度の求め方

きない．そこで本研究では、時間に基づくページ共起度を求める手法を用いる．あるページを長い間閲覧しているページほど、そのページから長い時間が経ってから閲覧したページにも共起閲覧度を付与するというものである．ただし、長い時間が経ってから閲覧したページには大きな値を付与しない．

図 2 のように、ページ p_1 からページ p_4 まで順にフォーカスを切り替えて閲覧したとする．あるページに対して他のページがどれくらい共起関係にあるかということここでは共起影響度と呼ぶ．このとき、ページ p_1 の他のページに対する共起影響度は $m_{p_1} = at$ という式に従って増加する．一定時間以上閲覧しているページの共起影響度は一定値以上増加しないとする．したがって、 $m_{p_1} \geq 1$ となると、 $m_{p_1} = 1$ とする．そしてページ p_1 からフォーカスが移ると、ページ p_1 の共起影響度は、 $m_{p_1} = bt$ という式に従って減少する．

このとき、ページ p_1 のページ p_2, p_3, p_4 に対する共起度を図で色別に記された範囲の面積とする．ページ p_2 は少しの間しか閲覧していないため、ページ p_3 に対してのみ共起度が付与され、その値も小さい．

フォーカスが移っていた時間に基づいて共起度を求めることによって、フォーカスを移し間違えた場合にも対処することができる．例えば図 2 は、ユーザは p_1 から p_3 にフォーカスを移すつもりが、間違っ p_2 にフォーカスを移してから、 p_3 にフォーカスが移ったものとする．このとき、ページにフォーカスが移った回数によって共起度を定義する手法では、 $\text{CoOc}(p_1, p_3)$ には値が付与されない．しかし、ページにフォーカスが移っていた時間によって共起度を計算すれば、 $\text{CoOc}(p_1, p_3)$ は定義され、ある程度その値も大きい．また、 $\text{CoOc}(p_1, p_2)$ はあまり大きい値をとらない．

4.2 共検索度

ページ p_1 のページ p_2 に対する共検索度は、ページ p_1 にとってページ p_2 がどれくらい共検索ページらしいかを表す．ユーザは情報要求を検索エンジンに対してクエリという形で表す．したがって、本研究では共検索度を、ページを開くために用いたクエリを用いて計算する．ページを開くために用いたクエリとは検索結果から直接リンクして開いたページでは、その検索

に用いたクエリを指す．それ以外のページでは、そのページがリンクによって訪れたページならば、リンク元のページを開くために用いたクエリを指す．例えば「京都 観光」というクエリで検索を行い、そこからリンクによって複数のページを訪れたとする．このとき、それらの全てのページで開くために用いたクエリは「京都 観光」となる．

ページを開くために用いたクエリを用いて共検索度を計算するにあたって、以下の 2 つを考慮した．

- (1) クエリの文字列の類似
- (2) クエリで検索した際の検索結果の類似

クエリの文字列の類似度は、N-gram 法を用いてベクトル化し、コサイン類似度を用いて計算する．また、N-gram 法を用いてベクトル化を行う際にクエリをキーワード毎に分割し、キーワードの前後に N-1 個の意味のない文字を付加する．例えば「京都 観光」というクエリを 3-gram 法でベクトル化するとする．まず「京都」「観光」とキーワードに分割する．次に「\$\$京都\$\$」「\$\$観光\$\$」と意味のない 2 文字を前後に付加する．最後にこれらの語を 3-gram 法によって分割し、ベクトル化を行う．

クエリで検索した際の検索結果の類似度は、クエリの検索結果の上位 k 件のタイトルとスニペットを N-gram 法を用いてベクトル化し、コサイン類似度を用いて計算する．

ユーザはある情報要求のもと、複数のクエリを用いて Web 検索を行うことがある．このとき、文字列は類似していないが、検索結果だけ類似しているクエリを入力することもあるだろう．例えば、デジタルカメラの各機種について調べたいという情報要求のもと、「サイバショット」というクエリと「LUMIX」というクエリを用いて Web 検索を行う．このとき、2 つのクエリの文字列は類似していないが、その検索結果はある程度類似している．クエリの文字列の類似度とクエリで検索した際の検索結果の類似度を組み合わせることで、このような場合でも共検索度を正しく計算できると考えた．従って、共検索度を以下のように計算する．

$$\text{CoSe}(p_1, p_2) = \alpha \cdot \text{SimQ}(q(p_1), q(p_2)) + (1 - \alpha) \cdot \text{SimSR}(q(p_1), q(p_2))$$

ここで $\text{CoSe}(p_1, p_2)$ はページ p_1 とページ p_2 の共検索度、 $q(p)$ はページ p を開くために用いたクエリ、 $\text{SimQ}(q_1, q_2)$ はクエリ q_1 とクエリ q_2 のクエリ類似度、 $\text{SimSR}(q_1, q_2)$ はクエリ q_1 とクエリ q_2 の検索結果の類似度を指す． α は $0 \leq \alpha \leq 1$ の係数である．

4.3 ページ重要度と時間近接度

あるページの重要度とは、ユーザがそのページからどれだけ価値ある情報を得られたかを示す．価値ある情報が得られたページほど、タスクの再開時に必要であると考えた．例えばユーザがページの文章の一部をコピー&ペーストしたとすると、ユーザはそのページから価値ある情報を得たと推測できる．また、閲覧時間が長いだけでもユーザはそのページの情報をより多く得ていると考えられるので、価値のある情報を得ている可能性が高いと推測できる．本研究では、ページの重要度をユーザがページを実際に閲覧していた時間によって計算する．従って、以下のように計算する．

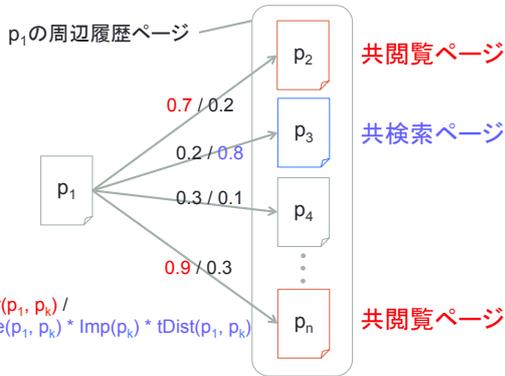


図 3 共閲覧ページ，共検索ページの推定

$$\text{Imp}(p) = \log(\text{FocusTime}(p))$$

ここで p はページ， $\text{Imp}(p)$ はページ p の重要度， $\text{FocusTime}(p)$ はページ p にフォーカスが当たっていた合計時間を指す。

ページ間の時間近接度とは，2つのページがどれだけ近い時間に開かれたかを示す．近い時間に開かれたページ同士のほうが，関連が強くタスクの再開時に必要であると考えた．ページ p_1 とページ p_2 の時間近接度を以下のように計算する．

$$\text{tDist}(p_1, p_2) = \frac{1}{\log(|t(p_1) - t(p_2)|)}$$

ここで， p はページ， $\text{tDist}(p_1, p_2)$ はページ p_1 とページ p_2 の時間近接度， $t(p)$ はページ p を開いた時間を指す．

4.4 再閲覧ページの推定と保存

本節では，あるページに対して再閲覧ページを推定する手法を述べる．推定方法の全体的なイメージ図を図3に示す．推定方法は以下になる．なお，共検索度が大きいページの中でも，ページ重要度と時間近接度が大きいほうがタスクの再開に有用であると考えた

- (1) あるページの周辺履歴ページに対して，共閲覧度，共検索度，ページ重要度，時間近接度を求める
- (2) 共検索度の大きさが上位 a 件のページを共閲覧ページと推定する．なお，2回以上訪問しているページに対しては，各訪問時毎に共閲覧度を求め，その値を足し合わせた値を共閲覧度とする．
- (3) 以下の式の大きさが上位 b 件のページを共検索ページと推定する．

$$\text{CoSe}(p_1, p_2) \cdot \text{Imp}(p_2) \cdot \text{tDist}(p_1, p_2)$$

なお， $p_2 \in \text{Can}(p_1)$ である．2回以上訪問しているページに対しては，各訪問時毎にこの値を求めて足し合わせた値を用いて共検索ページを推定する．

こうして推定された共閲覧ページと共検索ページを再閲覧ページとして保存する．このとき，ページを開くために用いたクエリなどの情報も共に保存する．

4.5 再閲覧ページの表示

4.4節で保存した再閲覧ページを，図4のようにグラフを用いて表示する．ノードは全てページを表している．まず，現在



図 4 Web 検索結果に併せて表示された再閲覧ページのグラフ

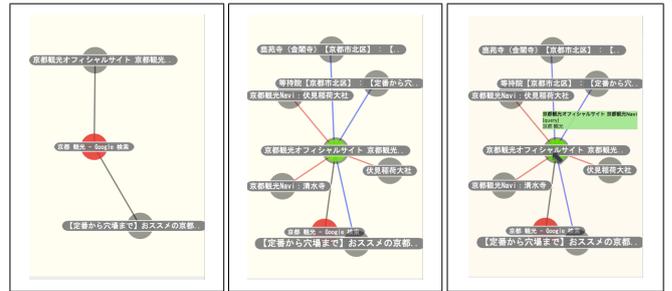


図 5 検索結果表示時 図 6 ノード展開時 図 7 マウスオーバー時

閲覧しているページのノードが表示される．また，現在閲覧しているページが，

検索エンジンの検索結果ページの場合

検索結果に以前に閲覧したことのあるページがあれば，そのページのノードと現在閲覧しているページのノードをつないで表示．

以前に閲覧したことのあるページの場合

そのページの再閲覧ページのノードと現在閲覧しているページのノードをつないで表示．

ノードの色は，

赤色 現在閲覧しているページのノード

黄緑色 最後に展開したノード

灰色 それ以外のノード

を意味している．共閲覧ページの関係は赤色のエッジで，共検索ページの関係は青色のエッジで示し，それ以外の関係は灰色のエッジで示す．各ノードに対しては以下の2つの操作が可能である．

- (1) ノードのページを開く
- (2) ノードを展開

ノードを展開することでそのノードが表すページの再閲覧ページのノードが表示される．

例えば，ユーザが Google の検索エンジンで「京都 観光」というクエリを入力して検索を行う．このとき検索結果にユーザが以前に閲覧したことのあるページが2件あった．このとき，図4のようにノードが検索結果の横に表示される．ノードにはページのタイトル名が記載されている．表示されるノードは図5のように，3つある．1つは検索結果ページのノードである．残りの2つのノードは検索結果のうち，ユーザが以前に閲覧したことのあるページのノードである．

図5のノードの1つを展開すると図6のようにノードが表示ページの見出しノードが表示される。また、各ノードをマウスオーバーすることによって、図7のようにそのノードが表示ページを開くために用いたクエリなどの情報が表示される。

ユーザが再検索をしようと通常の検索行動を行った際に1ページでも以前に閲覧したページを見つけることができれば、その再閲覧ページが表示されるので、素早く以前に閲覧したページを参照することが可能になると考えた。また、グラフを用いて表示しているので、ページ間の関係を容易に把握できる。表示されたページ群を見ることで以前の検索行動を思い出すことも可能であると考えた。

5. 実験と評価

前章で述べた提案手法の有用性を検証するため、ユーザ実験を行った。本実験の目的は、提案手法を用いることでタスクの再開がスムーズになるかを検証することである。

5.1 様々なログの取得

本手法を実装する為に、ユーザの Web 閲覧・検索行動のログを収集する。ログの収集には Web ブラウザの Firefox^(注1)を用いた。収集するログは、Firefox が元々閲覧履歴として記録している情報とタブ状態履歴である。タブ状態履歴とは、タブブラウザを用いて Web ページの閲覧を行う際のタブの生成、タブのフォーカスの切り替え等のタブ状態の変化を記録したものである。

Firefox は履歴情報としてユーザが訪問したページに対して

- (1) ページのタイトル・URL
- (2) 訪問日時
- (3) リンク元のページ

を1つの組として保存し、IDを用いて管理している。

5.1.1 タブ状態履歴

Firefox の拡張機能を用いてタブ状態履歴収集ツールを作成し、タブ状態履歴を保存する。タブ状態履歴収集ツールは、タブの生成、タブのフォーカスの変更、タブの消去、ページの移動のいずれかのイベントが発生した際に、ブラウザ上の全てのタブに対して

- (1) タブ ID
- (2) ページのタイトル・URL

を取得し、時刻、イベントの種類、イベントが生じたタブ、フォーカスされているタブの情報と共に保存するものである。

5.1.2 取得できる情報

収集したログより、最終的には訪問した各ページに対して、以下の情報が取得できる。

- (1) ページのタイトル・URL
- (2) ページを開いた/閉じた時刻
- (3) ページにフォーカスが当たった時刻
- (4) ページからフォーカスが外れた時刻
- (5) ページを開くために用いたクエリ

この情報以外にも、同時にタブとして開いていたページ群なども取得できる。2回以上訪問しているページについては、各訪問毎に上記に列挙した情報を取得しており、IDにより識別する。

5.2 インタフェースの実装

本実験では、Firefox の拡張機能を用いて、4.5 節に示した再閲覧ページを表示するインタフェースを実装した。再閲覧ページをサイドバーに表示する仕組みである。表示される情報はページが切り替わるたびに切り替わっていくのではなく、ユーザの操作によって切り替わる。なお、各ページに対して、表示する再閲覧ページ、再検索ページはそれぞれ3件ずつとした。

5.3 実験の概要

実験は行っていたタスクを途中で中断して、後で再開するという状況を想定した。タスクは複数のタブを開いて検索するようなものを用意した。実験には、20代の男性4人に参加してもらった。なお、全員情報検索には慣れている。

実験の内容を以下に述べる。実験はユーザにタスクを前半と後半に区切って行ってもらった。タスクの内容を表1に記す。なお、前半と後半でタスクの内容は変わらない。タスクの前半では何も用いずに Web ページの閲覧や Web 検索を行ってもらい、タスクの後半では、本インタフェースを使用する場合としない場合に分けてタスクを再開してもらった。タスクの後半は、タスクの前半が終わってから、一定の時間をあけてから再開してもらった。なお、タスクは一人につき6つ行ってもらった。具体的には以下の手順で実験を行った。

- (1) まず、本インタフェースについて説明を行い、操作方法等に慣れてもらう。
- (2) 次に各タスクに関して Web ページの閲覧や Web 検索を順番に10分間ずつ行ってもらい、10分経てばそのタスクが途中で全てのページを閉じてもらい、次のタスクに移ってもらう。
- (3) 次に前半と同じ順番で各タスクに関して Web ページの閲覧や Web 検索を再開してもらい、10分間ずつ行ってもらい、各タスクが終了したときにそのタスクに関してのアンケートに回答してもらう。
- (4) 全体を通してのアンケートに回答してもらう。

各タスクの前半は Web 閲覧・検索行動を集めるために行う。各タスクの後半は、3つのタスクで本インタフェースを使用してもらい、残りの3つのタスクでは、本インタフェースを使用しない。被験者によって本インタフェースを使用するタスクを変えた。なお、本インタフェースを使用しないタスクでは履歴機能等の使用は可能とした。また、順序効果がでないように、被験者によってタスクを行う順番を変えた。

アンケートの項目は各タスクが終了した際のもをを表2に、全てのタスクが終了した際のもをを表3に記す。

Q1, Q4~Q6 はリッカート尺度(1~5, 5が最も良い)に従って回答してもらった。また、Q2, Q3, Q6, Q7 は自由記述欄を設けた。

5.4 結果と考察

5.3 節の実験で得た結果を元に考察を行う。

(注1): <http://www.mozilla.jp/firefox/>

表 1 タスク内容

タスク番号	タスク内容
T1	奈良の寺社を観光したい。奈良の有名な寺社をできるだけ廻ることができるように予定を立ててください。ただし、10 時に JR 奈良駅集合、19 時に JR 奈良駅解散とする。
T2	デジタルカメラを購入したい。予算は 3 万円で、いくつか候補を挙げて、それぞれの違いを説明してください。
T3	名古屋駅近くでホテルを予約したい。3 月 20 日に宿泊予定である。値段は 1 泊 1 万円前後で、あなたが泊まりたいと思うホテルを探してください。
T4	東京ディズニーランドへ 0 泊 3 日で夜行バスを使って行きたい。出発は 2 月 21 日の夜で、チケット代も含めてできるだけ安いものを探してください。
T5	京都大学付近で、賃貸物件を探してください。家賃は 5 万円以内で、トイレ・バス別が最低条件です。あなたが住みたいと思う部屋を探してください。
T6	TOEFL を受験することになった。実際に勉強するとして良い参考書を調べてください。

表 2 各タスク終了時のアンケートの質問項目

質問番号	質問
Q1	タスクが提示した要件を完全に満たせましたか？
Q2	今回のタスクの再開にあたって、直面した問題のうち次の中で当てはまるもの全てにチェックをつけてください 1. タスクの前半でどのようなページを閲覧したか思い出せなかった 2. タスクの前半で閲覧したページの再発見に時間がかかった 3. タスクの前半で閲覧したページを再発見できなかった 4. タスクの前半で同時に開いていた複数のページを開き直すのに時間がかかった 5. タスクの前半で用いたクエリを思い出せなかった 6. タスクの前半で用いたクエリは覚えていたが、どのようにリンクを辿れば目的のページを発見できるか覚えていなかった

表 3 全てのタスク終了時のアンケートの質問項目

質問番号	質問
Q3	普段使用しているブラウザは何ですか？
Q4	普段から履歴機能をよく使いますか？
Q5	本インタフェースを用いることは以下の各項目の問題に対してどの程度有用でしたか？なお、該当する問題が生じなかった場合は、未記入としてください。 1. 以前に行ったタスクでどのようなページを閲覧したか思い出せない 2. 以前のタスクで閲覧したページの再発見に時間がかかる 3. 以前のタスクで閲覧したページの再発見ができない 4. 以前のタスクで同時に開いていた複数のページを開き直す 5. 以前のタスクで用いたクエリを思い出せない 6. 以前のタスクで用いたクエリは覚えていたが、どのようにリンクを辿れば目的のページを発見できるか覚えていない
Q6	本インタフェースを使用して、有用な情報を得られましたか？得られたら、それは具体的にどのようなことですか？
Q7	本インタフェースについて思ったことはありますか？

5.4.1 各タスク終了時のアンケートに基づく考察

Q1 の結果を本インタフェースを使用した場合と使用していない場合に分けてその平均をとったものを表 4 に示す。表 4 が示す通り、本インタフェースを使用するか否かによってタスクの達成度は差異がなかった。これは、タスクの内容が 20 分あればある程度達成することができるものであり、全体的にタスクの達成度が高くなってしまったために、差異がみられなかったと考えられる。

次に Q2 の回答から、タスクの再開にあたって直面した問題数をタスク毎に Q1 と同様にインタフェース使用の可否によって分け、平均をとったものを表 5 に示す。表 5 が示す通り、本インタフェースを使用した場合の方が使用しない場合より、

表 4 タスクの達成度の平均 (1-5)

	インタフェース使用	不使用
T1	2.50	4.50
T2	4.50	4.00
T3	4.50	5.00
T4	4.50	4.50
T5	4.50	3.50
T6	5.00	4.00
平均	4.25	4.25

表 5 タスク再開時に直面した問題数の平均

	インタフェース使用	不使用
T1	0.50	1.50
T2	1.00	1.00
T3	0.50	2.50
T4	1.00	3.00
T5	1.00	0.50
T6	1.00	1.50
平均	0.83	1.67

表 6 直面した問題毎の問題発生割合

	インタフェース使用	不使用
Q2-1	0	0.333
Q2-2	0.167	0.500
Q2-3	0.083	0.083
Q2-4	0.250	0.250
Q2-5	0.250	0.250
Q2-6	0.083	0.250

表 7 Q3 の結果

Firefox	3	質問	Q4	Q5-1	Q5-2	Q5-3	Q5-4	Q5-5	Q5-6	Q6
Chrome	1	平均値	2.5	4.25	4.5	4	3.75	2.25	4.333	4

表 8 Q4-Q6 の各評価の平均

	インタフェース使用	不使用
タスク前半の平均閲覧ページ数	25.92	20.08
タスク後半の平均閲覧ページ数	23.92	20.25
後半に閲覧したページの内、前半でも閲覧したページの割合	0.300	0.177
タスク前半の平均使用クエリ数	3.83	3.41
タスク後半の平均使用クエリ数	4.75	3.00
後半に使用したクエリの内、前半でも使用したクエリの割合	0.227	0.223

表 9 ユーザのタスクにおける閲覧ページ数と使用クエリ数

	インタフェース使用	不使用
タスク前半の平均閲覧ページ数	25.92	20.08
タスク後半の平均閲覧ページ数	23.92	20.25
後半に閲覧したページの内、前半でも閲覧したページの割合	0.300	0.177
タスク前半の平均使用クエリ数	3.83	3.41
タスク後半の平均使用クエリ数	4.75	3.00
後半に使用したクエリの内、前半でも使用したクエリの割合	0.227	0.223

タスクの再開に当たって直面した問題が少なくなっていることが分かる。また、Q2 の各問題毎の問題発生割合を表 6 に示す。Q2-1 に関しては、本インタフェースを用いることによって以前の閲覧時の情報を思い出せるということが分かる。Q2-2 に関しては、本インタフェースを用いることによって、素早く以前に閲覧したページを開くことができるということが分かる。Q2-6 に関しては、本インタフェースがページの再発見に有用だということが分かる。

Q2-3 ~ Q2-5 に関しては差異がでなかった。Q2-3、Q2-4 に関しては、タスクの時間が 10 分と短く、タスクの前半と後半を行った間隔が短かったため、大量のページを閲覧し、情報を整理することができなくなるという状況を完全に再現することができなかったためであると考えられる。

5.4.2 全てのタスク終了時のアンケートに基づく考察

普段用いているブラウザは表 7 のように、Firefox が 3 人、Chrome が 1 人であった。普段 Chrome を用いていると回答した人も Firefox を使うことはあると言っており、普段用いているブラウザによる実験結果の差異はないと考えられる。また、Q4 ~ Q6 の各質問に対する評価の平均を表 8 に示す。

Q4 の評価が示すように、普段から履歴機能を積極的に使用する人は少ないと考えられる。しかし、ユーザは頻繁に再検索を行うので、再検索を支援する機能は必要であると考えられる。また、Q5、Q6 の評価が示すように、本インタフェースは再検索や再発見を行う上で概ね有用であると考えられる。

5.4.3 全体の考察

実験結果についての全体的な考察を述べる。表 9 にユーザの閲覧ページ数と使用クエリ数を乗せる。タスクの後半で閲覧したページの内、前半でも閲覧したページの割合が本インタ

フェースを用いた場合の方が大きくなっている。従って、本インタフェースは以前に閲覧したページにアクセスしやすくなっていると考えられる。実験で得られたアンケートの自由記述欄を見ると「前回どのページにアクセスしたか分かりやすかった」や「以前開いていたページを再度開きたい時にいちいち探さなくてよかった」といった回答が見受けられた。5.4.1節や5.4.2節で示した評価結果にもあるように、本インタフェースは以前に閲覧したページを素早く開くという点や、以前に閲覧したページを思い出すという点で有用であると考えられる。

「入力したクエリを忘れてしまった時に、支援が欲しい」といった回答も見受けられた。これはQ2やQ5のクエリを思い出せないという問題に対する評価にも表れている。以前に閲覧したことのあるページを最低でも1つ見つけないと、関連するページを表示できないので、この問題について対処できていないと考えられる。また、「ノードの展開を繰り返すと、グラフが見づらい」や「ページのタイトルが類似しているときにどれがどのページか分かりにくい」といった回答も見受けられたように、表示方法については見直す必要がある。

6. 結 論

本研究では、タスクの再開をスムーズにするために、ユーザが閲覧したページを単一で保存するのではなく、同時に開いていたという情報や、タブを切り替えたという情報などを保存し、それらの中からあるページに対して関連の強いページがどれかを推定する手法を提案した。タスクの再開に有用であるページとして、共閲覧ページと共検索ページを提案し、それらを共閲覧度、共検索度という尺度を主に用いて推定した。また、現在閲覧しているページに応じて、共閲覧ページと共検索ページをグラフを用いて表示する方法を示した。

提案手法の有用性を検証するために、ユーザ実験を行った。実験では、提案手法が以前閲覧したページに素早くアクセスできるということや、以前に閲覧したページを思い出すことができるという点で有用であるということが分かった。また、以前に検索を行ったクエリを覚えていない場合には対処できないという問題点も見つかった。

今後の課題としては、まず、実験で発見した問題を改善することが考えられる。具体的にはユーザの再発見・再検索をクエリを入れるところから支援するといったことや、インタフェースの改良がある。実験の被験者が4人と十分な数とは言えないので、さらなるデータをとることを検討している。また、共閲覧ページや共検索ページが本当にタスクの再開に役立ったかを調べる必要もある。

謝 辞

本研究の一部は、文部科学省科学研究費補助金（課題番号24240013, 24680008）によるものです。ここに記して謝意を表します。

- [1] Patrick Dubroy and Ravin Balakrishnan. A study of tabbed browsing among mozilla firefox users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 673–682. ACM, 2010.
- [2] Akiko Hino and Katsumi Tanaka. Small knowledge: a concept of enhanced bookmarks with personal context. In *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication*, pp. 115–122. ACM, 2009.
- [3] Darris Hupp and Robert C Miller. Smart bookmarks: automatic retroactive macro recording on the web. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pp. 81–90. ACM, 2007.
- [4] Ricardo Kawase, George Papadakis, Eelco Herder, and Wolfgang Nejdl. The impact of bookmarks and annotations on refinding information. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, pp. 29–34. ACM, 2010.
- [5] Bonnie Ma Kay and Carolyn Watters. Exploring multi-session web tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1187–1196. ACM, 2008.
- [6] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, Vol. 1, p. 5. Cambridge University Press Cambridge, 2008.
- [7] Ippei Nishimoto and Masashi Toda. Process-recollective refinding on the web. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 883–892. IEEE Computer Society, 2006.
- [8] Jun Rekimoto. Time-machine computing: a time-centric approach for the information environment. In *Proceedings of the 12th annual ACM symposium on User interface software and technology*, pp. 45–54. ACM, 1999.
- [9] Benedikt Schmidt, Sebastian Doeweling, and Max Mühlhäuser. Interaction history visualization. In *Proceedings of the 30th ACM international conference on Design of communication*, pp. 261–270. ACM, 2012.
- [10] Sarah K Tyler and Jaime Teevan. Large scale query log analysis of re-finding. In *Proceedings of the third ACM international conference on Web search and data mining*, pp. 191–200. ACM, 2010.
- [11] Yousuke Watanabe, Kenichi Otagiri, and Haruo Yokota. Filesearchcube: a file grouping tool combining multiple types of interfile-relationships. In *Web-Age Information Management*, pp. 386–397. Springer, 2010.
- [12] 星加拓人. タブブラウザ上のウェブアクセス履歴の分析. 修士論文, 法政大学, 2011.
- [13] 内藤稔, 大島裕明, 高橋亜希子, 田中克己. 複数文書閲覧時の文書間の意味的関係の抽出と提示による文書ナビゲーション. 第9回日本データベース学会年次大会, F8-4, 2011.