筑波大学大学院博士課程

システム情報工学研究科修士論文

スレッドに基づく
Web閲覧履歴検索インタフェース

小澤　崇記

(コンピュータサイエンス専攻)

指導教員　田中　二郎

2008年3月

概要

曖昧な入力で履歴検索を行うことができるスレッドに基づいた Web 閲覧履歴検索システムインタフェース THiS を開発した。今日、急速に技術が発展したことによって、計算機で扱えるデジタルデータの量が増大し、Web 上の情報も今なお増大している。そのため、必要な情報を膨大な Web 空間から抽出することは難しく、検索技術の差が情報獲得の機会の差として顕著に表れると考えられる。

その過程において、過去に閲覧した Web ページを再閲覧する状況も想定されるが、Web 履歴が膨大になるにつれ、検索が困難になる。様々な研究が今までに行われているが、情報検索と履歴検索はユーザにとって異なる活動であり、異なった支援が必要である。

履歴検索を行う際、ユーザは曖昧な記憶を基に検索を行うが、この時、キーワードや日付といった明確な基準を他の情報なしに引き出すことは難しい。そのため、インタフェース側で提示する情報を増やし、あいまいな入力に対応する必要がある。

本研究では、ユーザがリンクを辿って閲覧した Web ページ群をスレッドとして捉えて、提示と検索のためのクエリの両方に用いた。さらに、クエリとして使える入力を URL やキーワードだけではなく、Web ページ、キーワード群まで拡大することで直感的な入力を可能にした。

目 次

図 目 次

第1章　序論

1.1　背景

今日、急速にコンピュータ技術や情報通信技術が発展したことにより、計算機で扱えるデジタルデータの量が増大してきている。そのため、Web 上の情報は網羅性という観点からとても重要な位置を占めるようになった。しかし、情報が増えれば増えるほど、その網羅性が増す半面、情報を検索する立場にある人にとって必要な情報を膨大な Web 空間から抽出することが難しくなる。このため、検索を行う人の検索技術の差が情報獲得の機会の差として顕著に表れると考えられる。

さらに、その過程において、過去に閲覧した Web ページを再度閲覧する状況が想定される。しかし、Web 履歴が膨大になるにつれて過去に閲覧した Web ページを探すのは同様に困難になる。このため、一度アクセスした情報に後で再びアクセスする際に、ユーザのアクセス履歴やブックマークを使ったユーザ支援がこれまでに多数、提案されてきている。

その中で、従来より広く研究されている情報検索は、ユーザが未知の目標についての想定された手がかりを基に、探索的に検索するというものである。これに対して、履歴検索はリファインディングと同様、ユーザが既知の目標に対して過去の体験や記憶情報を基に、方向限定的に目標を検索するというものである。情報検索とリファインディングが、ユーザにとって全く異なる活動であるため、それぞれに適した形での支援が必要である。[17]

1.2　既存の履歴インタフェースとその問題点

Internet Explorer の場合、図 1.1 のように履歴データを時系列順に一覧表示できるが、表示されるものは週情報、タイトル、URL のみである。また、Firefox の場合、図 1.2 のように一覧表示されるが、日付情報と、タイトル、URL のみの表示である。ユーザはこの中から、週や日付といった時間情報とタイトルもしくは URL を手がかりにして探すことになる。しかし、URL とタイトルだけでは、探しているページについての情報が明確でないときに見つけだすことは難しい。また、時間情報によって検索する場合も閲覧した量が多ければ、検索は難しくなる。さらに、同一 URL の Web ページを閲覧した際、ユーザに提示される情報は時間情報以外全て同じ情報である。これも検索を難しくしている要因の一つだと考えられる。

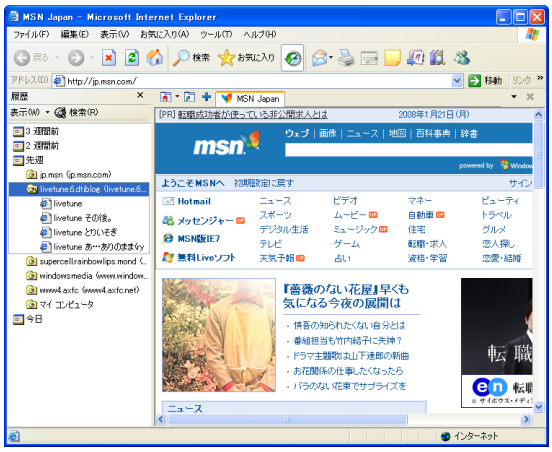

図 1.1: Internet Explorer における履歴検索

図 1.2: Mozilla Firefox における履歴検索

1.3 本研究のアプローチ

履歴検索を行う際、検索のキーとなるのはユーザの過去の記憶であるが、過去の記憶というものは時がたてばたつほど、曖昧なものになっていく。曖昧な記憶からキーワードや日付といった明確な基準を他の情報なしに引き出すことは難しい。そのため、インタフェース側は提示する情報を増やすことと、曖昧な入力に対応することの二つが考えられる。前者の場合、ユーザにとって何が有益な手がかりになりうる情報かをどう判断するかが焦点になる。仮に全ての情報を同時に見せるならば、情報量が多すぎて却って、ユーザが混乱することも考えられる。後者の場合、ユーザの頭の中にある情報をどのように表現させるかが焦点になる。

本研究では曖昧な入力で履歴検索を行うことができるスレッドに基づいた Web 閲覧履歴検索インタフェース THiS（Thread-based Web Browsing History Search）を提案する。ユーザの曖昧な入力に対応するために、履歴を Web ページ単位ではなく、スレッド単位で検索できるようにする。スレッドとは、ユーザがリンクを辿って閲覧した Web ページ群である。ユーザがリンクを辿る行為を一連の行動として捉えて、提示とクエリの両方に用いる。さらに、クエリへの入力を URL やキーワードだけでなく、Web ページ、キーワード群まで拡大することで、直感的な入力を可能にする。

1.4 本論文の構成

第 2 章では、本研究が提案するスレッド型 Web 閲覧履歴検索システムについて述べる。
第 3 章では、本インタフェースの実用例について述べる。
第 4 章では、関連研究を上げ、本研究と異なる点を述べる。
第 5 章では、実際のデータについて行った検証実験について述べる。
第 6 章では、本インタフェースについて考察を述べ、第 7 章でまとめる。

第2章　スレッド型Web閲覧履歴検索システム THiS

2.1　システム構成

本システムは、Web ページを取得する部分、Web ページの解析から Web 閲覧履歴の生成までを行う部分、検索インタフェース部分の 3 つから成る。まず、Web ブラウザの Web ページ取得プラグインがローカルディスクに保存する。その蓄積されたデータを用いて、Web ページの解析から Web 閲覧履歴の生成までを一つのプログラムが行う。作成された Web 閲覧履歴用のデータとローカルディスクに蓄積されているデータを検索インタフェースは読み込み表示する。

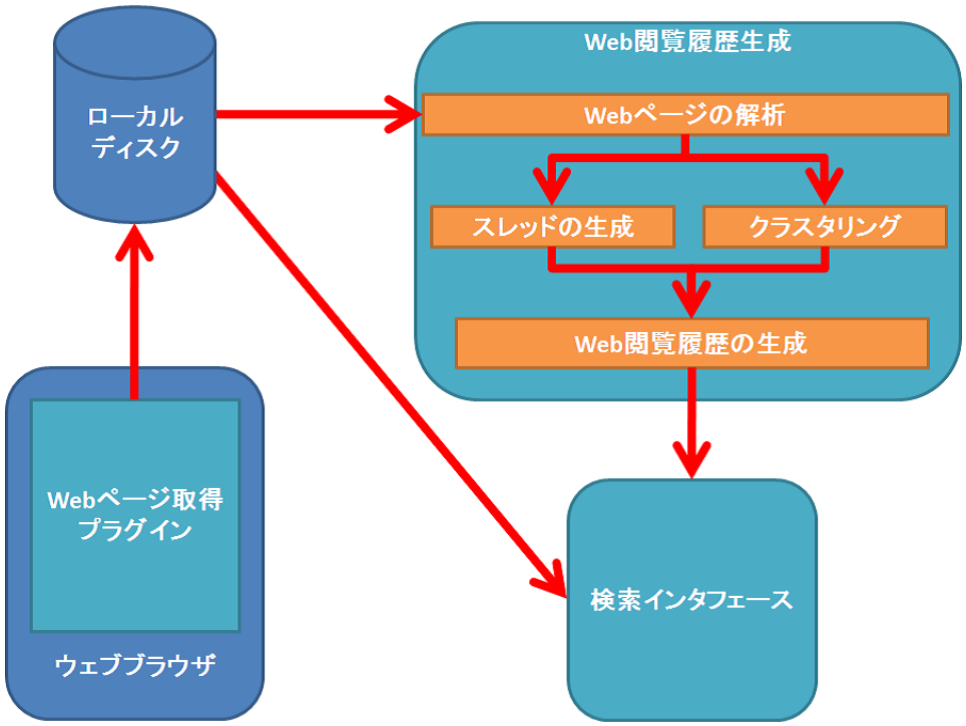

図 2.1: システム構成

2.2 ユーザが閲覧した Web ページの取得、及びブラウザ終了時刻の記録

ユーザが閲覧した Web ページの取得には Web ブラウザ Firefox の拡張機能である ScrapBook を使用した。ユーザが閲覧する、つまり Web ページが最前面に表示される度に ScrapBook を起動し、Web ページの全コンテンツと、最前面に表示された日時と URL、タイトルを同時にローカルディスクに保存しておく。また、ブラウザが終了した際、その日時を ScrapBook と同じ仕様で記録しておく。このように保存した全記録を Web 閲覧記録と呼ぶことにする。

2.3 Web ページの解析

Web 閲覧記録内の各 Web ページにおいて、URL、タイトル、リンク（アンカー URL とアンカーテキスト）、閲覧日時を取得し、文字数をカウントする。同時に、そのページが Google 等の検索ページであるのならば URL のクエリ部分を解析して、検索クエリを取得する。時系列的に一つ後のページの閲覧日時との差分から各 Web ページの閲覧時間を算出する。

また、各 URL の閲覧回数を数え、各ページがその URL において何回目の閲覧かを算出する。そして、以下の 3 つについて解析する。

- 一ページ前に閲覧した Web ページからリンクがあるかどうか。
- 閲覧時間が一時間以上であるかどうか。
- ブラウザを起動したときに開いたページであるかどうか。

算出された閲覧時間が一時間以上となっている Web ページに対しては、ユーザがその Web ページを見ていない、もしくは開いたまま放置していると考え、閲覧時間が一時間以内である全ページの平均値をその Web ページの閲覧時間に代替する。この時、代替する前の閲覧時間もデータとして残しておく。

2.4 スレッドの生成

現在処理している Web ページを P_t、時系列順において一つ前のページを P_{t-1} とする。

まず、新しくブラウザを起動した Web ページ、もしくは閲覧時間が一時間以上である Web ページを含む一番最近のスレッドを検出する。検出したスレッドの次のスレッドの最初のページから P_t までの間に、P_t と同一の Web ページがあるかどうか調べる。もし、あるならば、P_t がそのスレッドの最後のページであるかどうか調べる。

次に、各チェックを基にスレッドを生成していく。

1. P_{t-1} に P_t へのリンクがある場合は、最後に生成したスレッドの最後に P_t を追加する。リンクがない場合は、以下の処理を行う。

2. P_t で新しくブラウザを起動した場合、もしくは P_{t-1} の閲覧時間が一時間以上である場合、P_t のみの新しいスレッドを生成する。上記のどの条件にも当てはまらない場合は、以下の処理を行う。

3. ブラウザが閉じられた、もしくは閲覧時間が一時間以上である Web ページを含む一番最近のスレッドを検出する。

4. (3) で検出したスレッドの次のスレッドから P_t までの間で、P_t と同じ Web ページを含むスレッドを検出する。

5. (4) で検出したスレッドの中で一番新しいスレッドにおいて最後のページである場合、そのスレッドの最後に P_t を追加する。最後のページでない場合、そのスレッドの P_t と同じ Web ページまでと P_t から成る新しいスレッドを生成する。

6. 上記のどの条件にも当てはまらない場合、P_t のみの新しいスレッドを生成する。

図 2.2 の左は、時系列順に並べられた Web 閲覧記録であり、下に行くほど閲覧した日時が新しい Web ページである。図 2.2 の右は、これからスレッドを生成した結果である。線で結んであるページは同一 URL の Web ページである。

ユーザは一番目のページでブラウザを新しく起動し、ブラウザを閉じるまでに 22 ページ閲覧したとする。また、22 ページの閲覧時間は全て一時間未満であるとする。

- 2、3 ページ目：一つ前のページからリンクがあるため、一つ目のスレッドに追加されている。
- 4 ページ目：3 ページ目からリンクがなく、それまでに同じ Web ページがないので、新しいスレッドが生成されている。
- 5 ページ目：4 ページ目からリンクがあるため、2 つ目のスレッドに追加されている。
- 6、7 ページ目：4 ページ目と同じ理由でそれぞれ新しいスレッドが生成されている。
- 8、9 ページ目：一つ前のページからリンクがあるため、4 つ目のスレッドに追加されている。
- 10 ページ目：4 ページ目と同じ理由で新しいスレッドが生成されている。
- 11 ページ目：10 ページ目からリンクがなく、2 ページ目と同じ Web ページである。そして、2 ページ目を含むスレッドの中で一番新しいスレッドである一つ目のスレッドの最後のページと異なるページであるため、一つ目のスレッドの 2 ページ目までと、11 ページ目からなる新しいスレッドが生成されている。
- 12、13 ページ目：一つ前のページからリンクがあるため、6 つ目のスレッドに追加されている。

- 14 ページ目：13 ページ目からリンクがなく、10 ページ目と同じページである。そして、10 ページ目を含むスレッドの中で一番新しいスレッドである 5 つ目のスレッドの最後のページと同じページであるため、5 つ目のスレッドに追加されている。
- 15 ページ目：14 ページ目からリンクがなく、6 ページ目と同じページである。そして、6 ページ目を含むスレッドの中で一番新しいスレッドである 3 つ目のスレッドの最後のページと同じページであるため、3 つ目のスレッドに追加されている。
- 16 ページ目：4 ページ目と同じ理由で新しいスレッドが生成されている。
- 17 ページ目：16 ページ目からリンクがあるため、7 つ目のスレッドに追加されている。
- 18 ページ目：4 ページ目と同じ理由で新しいスレッドが生成されている。
- 19 ページ目：18 ページ目からリンクがなく、3 ページ目と同じページである。そして、3 ページ目を含む一番新しいスレッドである一つ目のスレッドの最後のページと同じページであるであるため、一つ目のスレッドに追加されている。
- 20 ページ目：19 ページ目からリンクがあるため、一つ目のスレッドに追加されている。
- 21 ページ目：20 ページ目からリンクがなく、10 ページ目と同じページである。そして、10 ページ目を含む一番新しいスレッドである 5 つ目のスレッドの最後のページと同じページであるため、5 つ目のスレッドに追加されている。
- 22 ページ目：21 ページ目からリンクがあるので、5 つ目のスレッドに追加されている。

2.5 クラスタリング

各 Web ページの特徴を抽出する際、文字情報の中にはページの特徴になりえない不用語もあるので、これらの語を省く必要がある。そのための手段の一つとして、形態素解析がある。形態素解析とは、文書の文字列を単体で意味が通る最小の文字列である形態素に分解し、品詞、語形変化、読みなどの情報を付加する処理である。表は「電子の歌姫初音ミクが、新語辞典『現代用語の基礎知識 2008』に収録されるらしい。」という文章に形態素解析器 MeCab を使って形態素解析を行った例である。

まず、各 Web ページの文字情報に対して、MeCab を用いて形態素解析を行う。その結果から名詞と未知語を抜き出す。名詞は最も特徴を表す品詞であると考え、また未知語は辞書に登録されていない固有名詞も含まれる可能性があると考え、選択した。抜き出した名詞と未知語から tf・idf 法とベクトル空間法を用いて各 Web ページの文書ベクトルを作成する。

tf・idf 法とは、「ある単語の、その文書における文書集合全体を考慮した相対的な重要度」を算出する手法である。文書 D_i の中の単語 t_j の重要度 w_{ij} を以下の計算式で求める。

$$w_{ij} = tf_{ij} \times idf_j$$

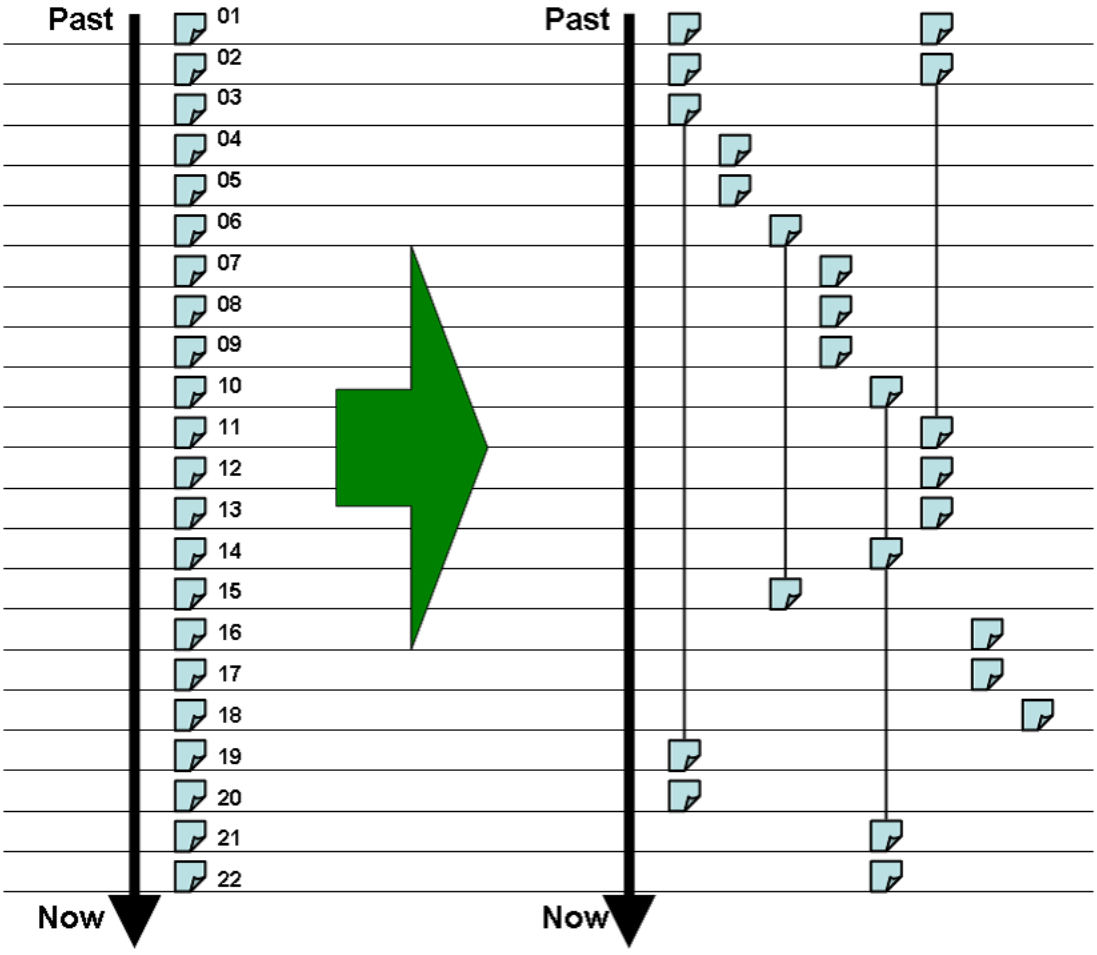
Past
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
Now
Past
Now

図 2.2: スレッドの生成

電子	名詞	一般	*	*	*	*	電子	デンシ	デンシ	○
の	助詞	連体化	*	*	*	*	の	ノ	ノ	
歌姫	名詞	一般	*	*	*	*	歌姫	ウタヒメ	ウタヒメ	○
初音	名詞	一般	*	*	*	*	初音	ハツネ	ハツネ	○
ミク	名詞	固有名詞	人名	名	*	*	ミク	ミク	ミク	○
が	助詞	格助詞	一般	*	*	*	が	ガ	ガ	
、	記号	読点	*	*	*	*	、	、	、	
新語	名詞	一般	*	*	*	*	新語	シンゴ	シンゴ	○
辞典	名詞	一般	*	*	*	*	辞典	ジテン	ジテン	○
『	記号	括弧開	*	*	*	*	『	『	『	
現代	名詞	一般	*	*	*	*	現代	ゲンダイ	ゲンダイ	○
用語	名詞	一般	*	*	*	*	用語	ヨウゴ	ヨーゴ	○
の	助詞	連体化	*	*	*	*	の	ノ	ノ	
基礎	名詞	一般	*	*	*	*	基礎	キソ	キソ	○
知識	名詞	一般	*	*	*	*	知識	チシキ	チシキ	○
2008	名詞	数	*	*	*	*	*			○
』	記号	括弧閉	*	*	*	*	』	』	』	
に	助詞	格助詞	一般	*	*	*	に	ニ	ニ	
収録	名詞	サ変接続	*	*	*	*	収録	シュウロク	シューロク	○
さ	動詞	自立	*	*	サ変・スル	未然レル接続	する	サ	サ	
れる	動詞	接尾	*	*	一段	基本形	れる	レル	レル	
らしい	助動詞	*	*	*	形容詞・イ段	基本形	らしい	ラシイ	ラシイ	
。	記号	句点	*	*	*	*	。	。	。	

図 2.3: 形態素解析の例

tf_{ij} とは、局所的重みとも呼ばれる文書 D_i の中での単語 t_j の出現頻度を表現している。文書 D_i に単語 t_j が多く出現すればするほど、tf_{ij} は大きな値となる。

idf_j とは、大域的重みとも呼ばれ、単語 t_j が全文書集合の中に出現すればするほど小さな値となり、珍しい単語であれば大きな値となる。

まとめると、ある文書 D_i における単語 t_j の重み（重要度）は、単語 D_i が文書 t_j においてよく出現し、かつ文書集合中において出現する文書数が少なければ大きくなるといえる。本研究では、idf_j を以下の数式によって求め、重みを決定した。

$$idf_j = \log \frac{N}{df_{t_j}} + 1$$

ベクトル空間法とは、文書やクエリ、カテゴリの内容を他次元空間上のベクトルとして表現する手法である。これには tf・idf 法を用いて得た重要度を適用する。m を文書集合全体の単語数、w_kj を文書 D_k 中の単語 t_j の重みとすると、文書 D_k はベクトル w_k で表現される。

$$w_k = [\, w_{k1}\ w_{k2}\ w_{k3} \cdots w_{km} \,]$$

作成した文書ベクトルを基に k-means クラスタリングを行う。

クラスタリングとは、分類を目的とする手法の一つであり、データに基づいて分類対象をいくつかのクラスター（グループ）に分類する。類似している者同士は同じクラスターに、類似していない者同士は異なるクラスターに分類される。その中で非階層的な手法の代表的な手法が k-means クラスタリングである。この手法では以下のプロセスでクラスターを作成する。

1. K 個のクラスターの中心をランダムに設定する

2. それぞれの個体を最も近い中心に割り当てる

3. クラスターごとに中心を計算しなおす

4. 全てのクラスターの中心が変化しなければ終了、それ以外は 2 からを繰り返す

本研究では、文書間の距離を類似度によって求めることによって、値の大きな文書同士ほど近くにあるようにクラスタリングがされる。各クラスタにおいて一番ベクトルの大きい単語の ID が若い順にインデックスを付ける。そして、各 Web ページにそのページが属しているクラスタのインデックスを付随する。それとは別に、各クラスタ間の類似度を算出しておく。

図 2.4 左は、時系列順に並べられた Web 閲覧記録であり、中央は 4 つにクラスタリングした結果である。そして、図 2.4 右では、図の説明上、順に一つ目のクラスタから、赤、青、黄、黒と色で区別している。実際にデータ上では、それぞれ 1、2、3、4 とインデックスがふられている。

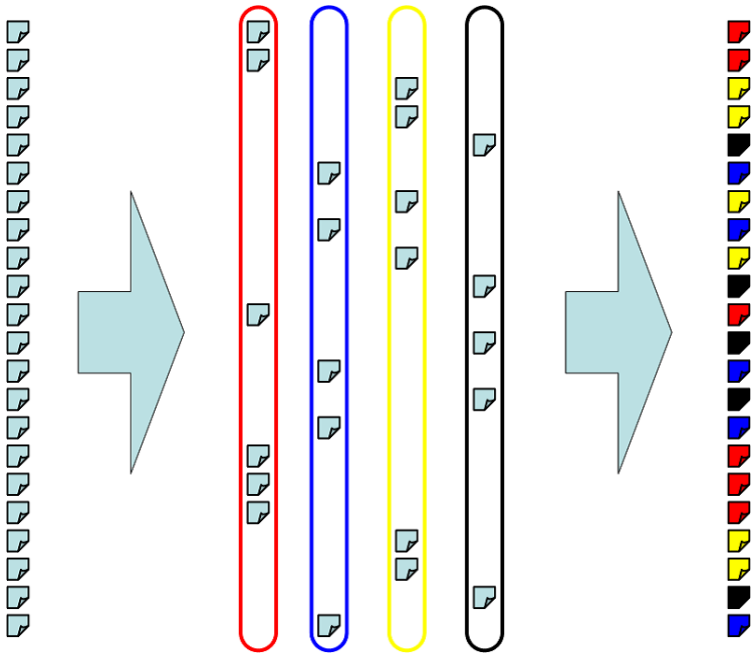

図 2.4: クラスタリング

2.6 Web 閲覧履歴の生成

スレッドと各スレッド内の Web ページの属するクラスタの ID を合わせることで、Web 閲覧履歴を生成する。Web 閲覧履歴中のパターンを基に、スレッド検索を行う。

図 2.5 左は、クラスタリングの結果であり、中央はスレッドを生成した結果である。図 2.5 右は生成された Web 閲覧履歴である。一つ目のスレッドから、スレッドにおけるパターンは

それぞれ、赤→赤→黄→黄→黄、黄→黒、青→青、黄→青→黄、黒→黒→黒→青、赤→赤→赤→黒→青、赤→赤、赤である。

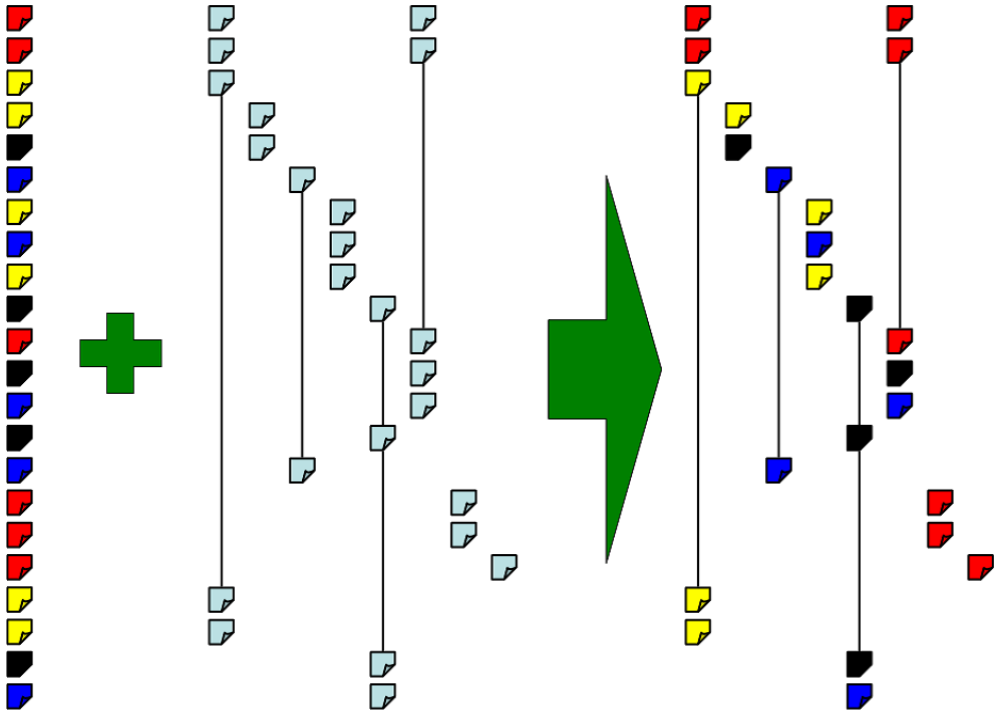

図 2.5: Web 閲覧履歴の生成

2.7 検索クエリ作成部

図 2.6 に本システムのイメージ図を示す。本システムは、クエリパーツ用ボタン、クエリパーツ用キーワード群、検索クエリ作成部、検索精度調整用スライダ、満足度算出方法選択用メニュー、Web 閲覧履歴提示部からなる。ユーザはボタンもしくはキーワード群からクエリパーツを選択し、検索クエリとしての閲覧パターンを作成する。システムは、そのパターンを基に Web 閲覧履歴からスレッドを検索して提示する。

ユーザは、検索クエリ作成部において、各クエリパーツボタンをクリックすることでそれぞれ Web ページクエリパーツ、キーワードクエリパーツを作成する。また、Web 閲覧履歴から Web ページを検索クエリ作成部にドラッグアンドドロップすることでも Web ページクエリパーツを作成することができる。そして、クエリパーツ用キーワード群をクリックする、もしくは検索クエリ作成部にドラッグアンドドロップすることで、パターンクエリパーツを作成する。これら 3 種のクエリパーツを組み合わせたり、並べ替えたりすることで、ユーザは検索クエリとしてのパターンを作成する。さらに、キーワードクエリパーツは左上のアイコ

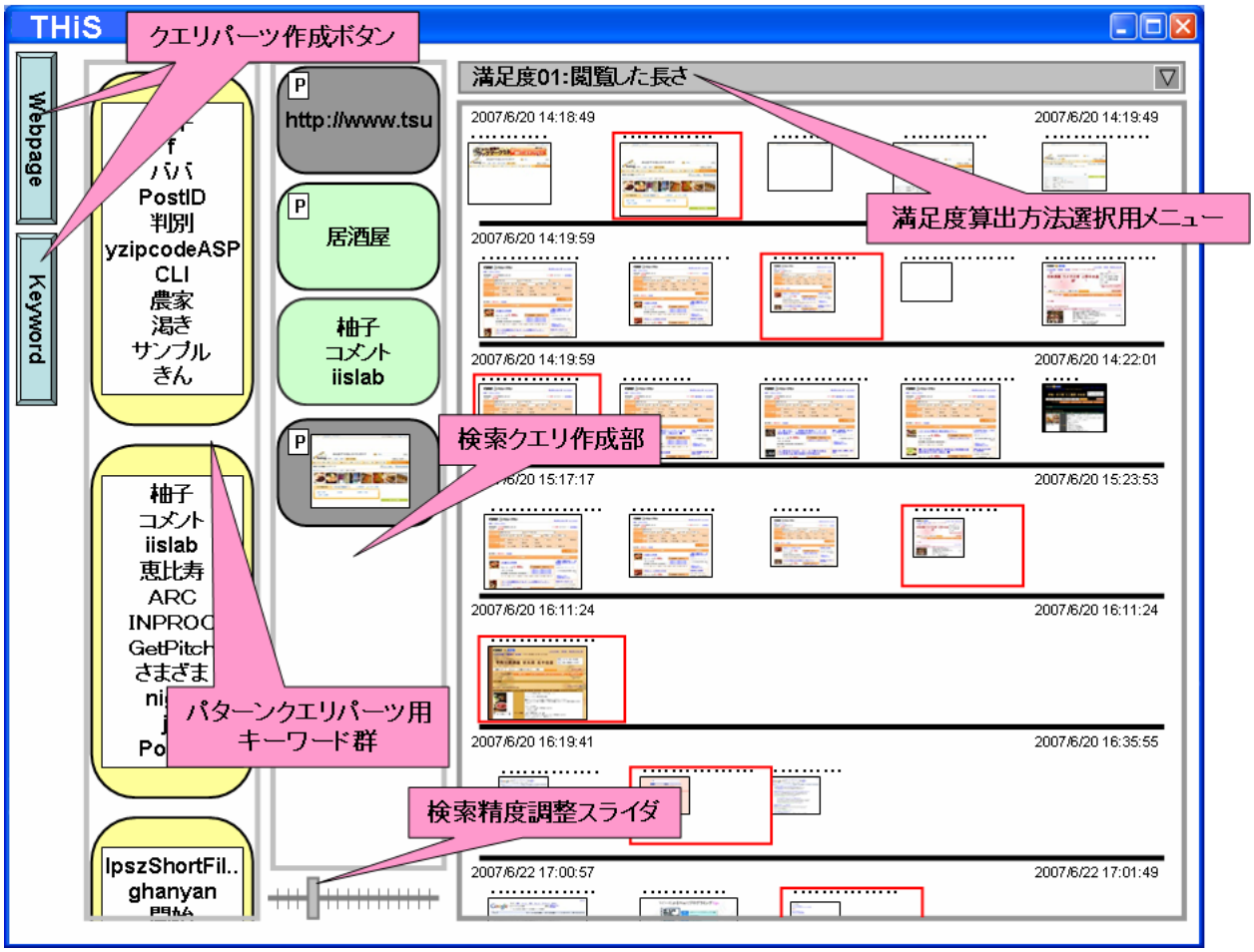

図 2.6: システムイメージ図

ンをクリックすることで、キーワード検索用のクエリ、もしくは検索ワード検索用のクエリにすることができる。それぞれのとき、左上のアイコンは「K」、「Q」となる。

2.7.1 Web ページクエリパーツ

Web ページクエリパーツの左上のアイコンは標準では「P（Patern）」である。

クエリパーツに入力された URL、もしくは履歴の ID からその Web ページを取得し、文字情報に対して形態素解析を行う。その結果から、名詞と未知語を抜き出し、その出現頻度をカウントすることで、ベクトルを作成する。ベクトルとクラスタリングによって得られた各クラスタのセントロイドとの類似度を求め、その値の一番大きなクラスタの ID をこのクエリパーツのパターンとする。

2.7.2 キーワードクエリパーツ

キーワードクエリパーツの左上のアイコンは、標準では「P（Patern）」である。

クエリパーツに入力されたキーワードを含むクラスタのうち、ベクトルの最も大きなクラスタの ID をこのクエリパーツのパターンとする。

2.7.3 パターンクエリパーツ

キーワード群は Web 閲覧記録をクラスタリングした際に、各クラスタにおいてベクトルが最も大きな 10 単語から成る。クラスタの ID がこのクエリパーツのパターンとなる。

2.7.4 パターンマッチング

Web 閲覧履歴のスレッドの中で検索クエリと同じパターンをもつスレッドを検索する。

図 2.7、図 2.8 にその仕組みを示す。両図とも左が Web 閲覧履歴であり、中央は検索クエリ、右はパターン検出結果である。図 2.7 では、検索クエリとしてパターン黄が与えられたとする。該当するページは 6 ページ検出され、そのページを含むスレッドは 3 つあることが分かる。検索クエリとしてのパターンに青が加わる（図 2.8）と、一致するパターンは一つ検出され、スレッドも一つに絞られていることが分かる。

また、スライダを調整することにより、類似パターンを含むスレッドまで検索範囲を拡大することができる。閲覧パターン p と q の類似度 dos_{pq} は、閲覧パターン p の m 番目 p_m、少ない方のパターンの数 n、クラスタ c_i と c_j の類似度 r_{ij} を基に以下の式によって求める。

$$dos_{pq} = \sum_{i=1}^{n} (r_{p_i q_i} \div n)$$

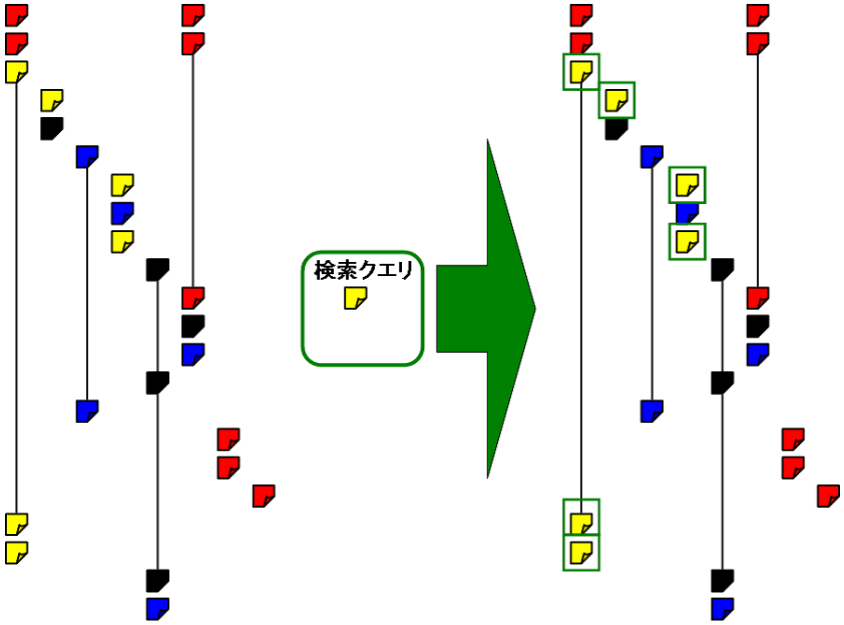

図 2.7: パターンマッチング（検索クエリ：黄）

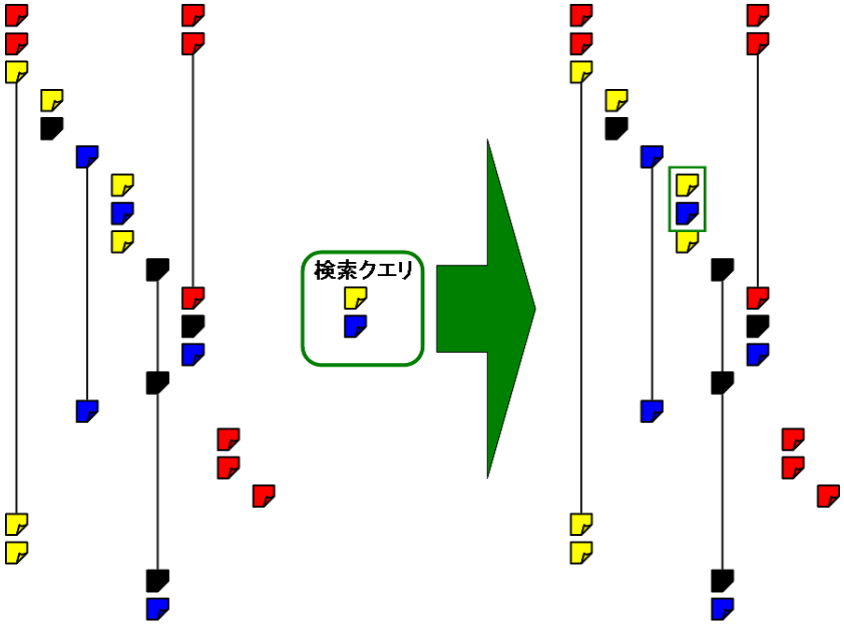

図 2.8: パターンマッチング（検索クエリ：黄 → 青）

2.7.5　URL 一致検索

Web ページクエリパーツの左上のアイコンが「U（URL）」のときは、URL 一致検索を行う。クエリパーツに入力された URL、もしくは履歴と同じ URL の Web ページを含むスレッドを検索する。

2.7.6　キーワード検索

キーワードクエリパーツの左上のアイコンが「K（Keyword）」のときは、キーワード検索を行う。クエリパーツに入力されたキーワードを含む Web ページを検出し、その Web ページを含むスレッドを検索する。

2.7.7　パターン遷移検索

検索クエリとしての閲覧パターン内のクラスタの遷移とスレッド内のクラスタの遷移からパターンマッチングを行う。類似度の算出方法はパターンマッチングと同様である。

図 2.9、図 2.10 にその仕組みを示す。図 2.9 は通常のパターンマッチングで赤 → 黄 → 青で検索した場合であり、一致するスレッドは一つしかないが、図 2.10 のように同パターンでも遷移検索にすると赤 → 黄 → 黄 → 黄 → 青のようなパターンを含むスレッドも検出され、一致するスレッドが 3 つに増えたことが分かる。

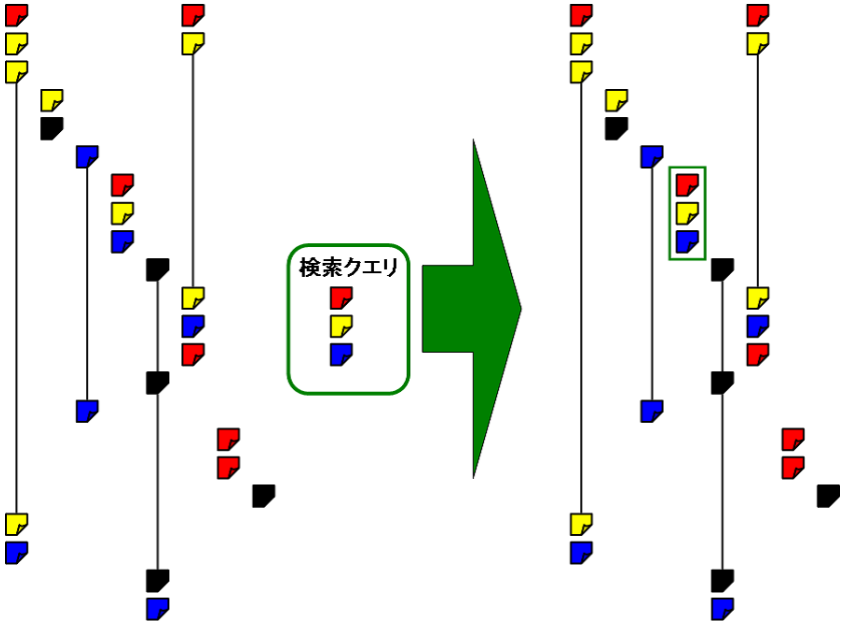

図 2.9: パターンマッチング（検索クエリ：赤 → 黄 → 青）

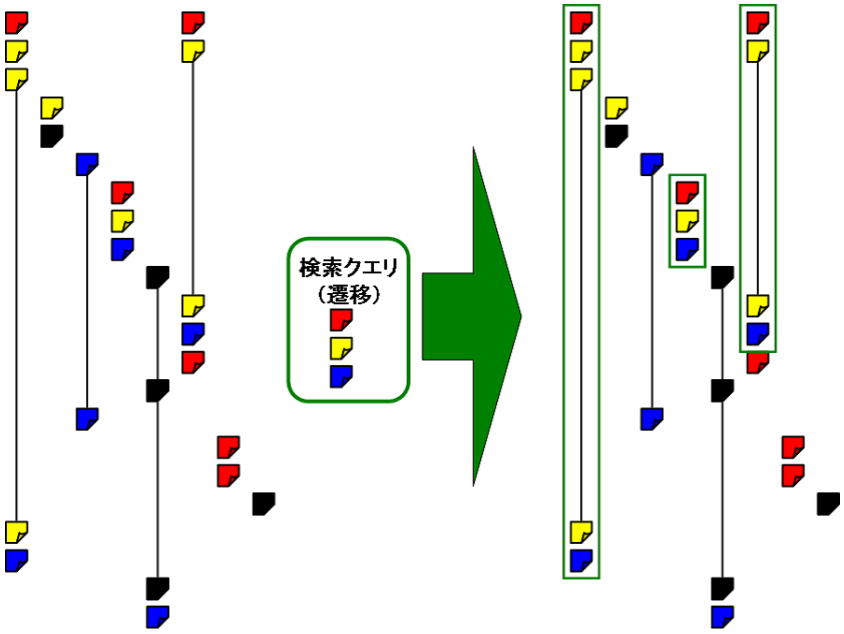

図 2.10: 図 2.9 をパターン遷移検索にした場合

2.7.8 検索ワード検索

キーワードクエリパーツのみ、もしくはキーワードクエリパーツがクエリパーツの一番上にあり、左上のアイコンが「Q（Query）」のときは、検索ワード検索を行う。検索ワード検索では、キーワードクエリパーツに入力されたキーワードで検索を行った検索ページを含むスレッドを検索する。

2.8 Web 閲覧履歴提示部

2.8.1 満足度算出

ユーザが選択した方法によって満足度を算出する。以下にその算出方法を示す。

1. 各 Web ページの閲覧した長さ
2. 各 Web ページを閲覧した回数
3. 各 Web ページをその時点までに閲覧した回数
4. (1) × (2)
5. (1) × (3)
6. (1) ÷ その Web ページの文字数

7. (2) ÷ その Web ページの文字数
8. (3) ÷ その Web ページの文字数
9. (1) × (2) ÷ その Web ページの文字数
10. (1) × (3) ÷ その Web ページの文字数
11. (1) ± (2) × 平均閲覧時間 ÷ 平均閲覧回数
12. (1) ± (3) × 平均閲覧時間 ÷ 平均閲覧回数
13. (1) ± (2) × 平均閲覧時間 ÷ 平均閲覧回数 ÷ その Web ページの文字数
14. (1) ± (3) × 平均閲覧時間 ÷ 平均閲覧回数 ÷ その Web ページの文字数
15. (1) × 検出された同一スレッドの数
16. (2) × 検出された同一スレッドの数
17. (3) × 検出された同一スレッドの数
18. (4) × 検出された同一スレッドの数
19. (5) × 検出された同一スレッドの数
20. (6) × 検出された同一スレッドの数
21. (7) × 検出された同一スレッドの数
22. (8) × 検出された同一スレッドの数
23. (9) × 検出された同一スレッドの数
24. (10) × 検出された同一スレッドの数
25. (11) × 検出された同一スレッドの数
26. (12) × 検出された同一スレッドの数
27. (13) × 検出された同一スレッドの数
28. (14) × 検出された同一スレッドの数

各スレッド内の Web ページのサムネイルは、満足度の値の大きさに比例して大きく表示する。ユーザが長く閲覧すればするほど、何回も閲覧すればするほど満足したと考える。また、文字数の多い Web ページは内容の良し悪しにかかわらず、閲覧時間が長くなると考えた。そして、Web ページ単位だけでなく、スレッド単位でも何回も見るほど満足度は高いと考えた。

2.8.2　表示部

検索結果として、一スレッドを一列で表示し、パターンが一致した部分をハイライトして提示する。スレッド内の Web ページの最大満足度によってスレッドの表示順番を変更する。

図 2.11 では、7 つのスレッドがそれぞれ最初から 5 ページ目までが表示されている。スレッドに関しては、スレッドの最初のページの閲覧開始日時と、最後のページの閲覧開始日時が提示される。また、スレッド内の各 Web ページはタイトルとサムネイルが提示される。検出されたスレッドのパターンと一致する部分は赤枠でハイライトして表示される。各サムネイルがクリックされると、各 Web ページがブラウザで開かれる。

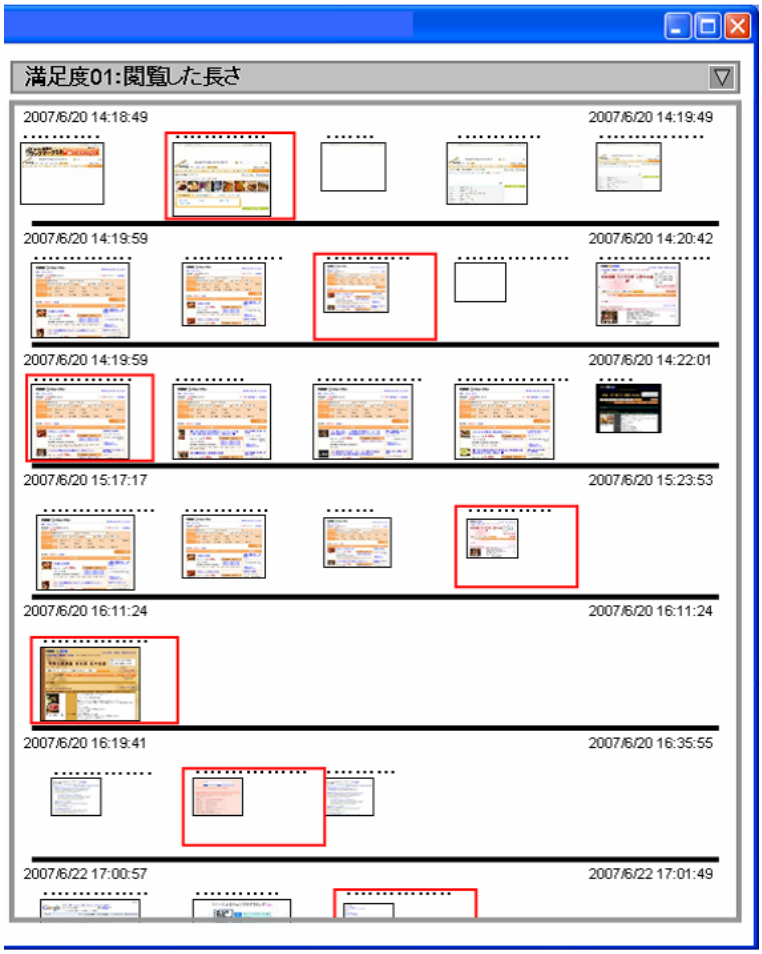

図 2.11: 表示部

第3章　使用例

この章では、具体的なタスクの例をあげて、本インタフェースの実用例について説明する。ユーザは過去に閲覧した Web ページの中から、以前探した飲食店の情報を探そうとしているとする。また、いつ探したのかも店の名前も店の場所も記憶が曖昧であり、Web ページの名前も覚えていなく、唯一覚えているのは、何かしらのポータルサイトからリンクを辿ったことだけであるとする。

まず、ユーザは Web ページクエリパーツボタンを押し、検索クエリ作成部に Web ページクエリパーツを作成し、URL を入力する。システムは、入力された URL の Web ページを取得し、パターンを生成する。そして、そのパターンを含むスレッドを検出して提示する。画面内に提示するスレッドは 7 つ、各スレッド内のページは 5 ページまでとする。

図 3.1 はシステムを起動したときの画面である。

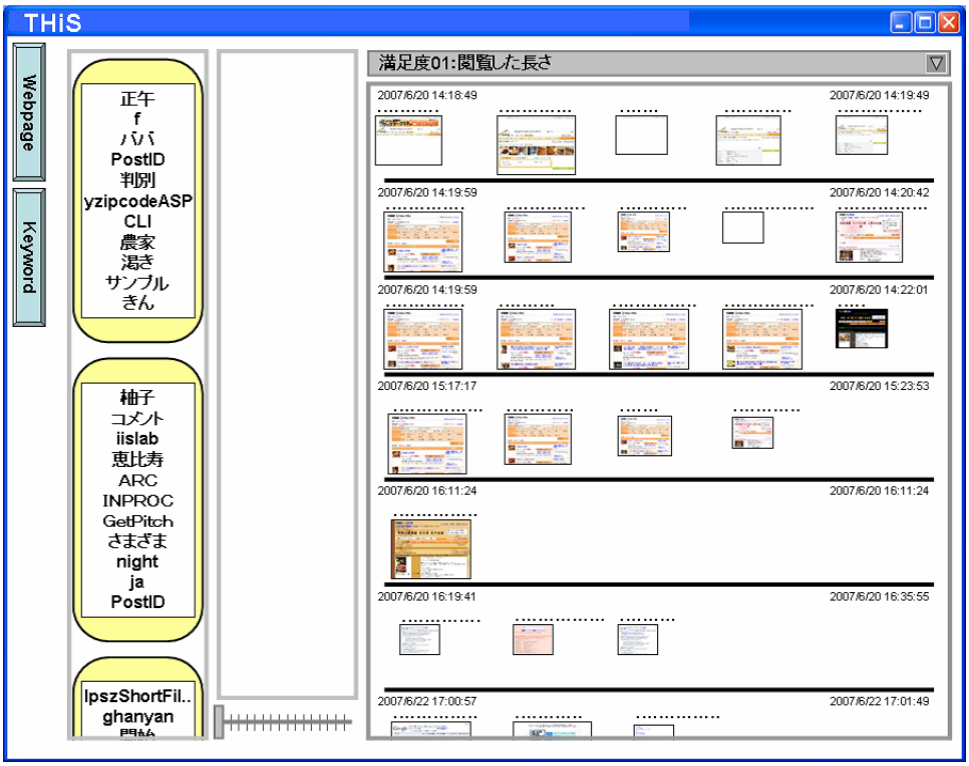

図 3.1: システム起動直後の図

図 3.2 は、システムにツクナビの URL「http://www.tsukunavi.com/」の Web ページクエリパーツ 1 つからなるパターンを与えた例である。この Web ページクエリパーツは、左上のアイコンが「P」なので、入力された URL の Web ページを基に作成したパターン（A) となっている。

そして、検索結果はスレッド内最大閲覧時間順に縦に並んでいる。また、スライダが一番左にセットされているため、検索クエリのパターンと完全に一致する部分が赤枠でハイライトされている。一番上のスレッドは 2007 年 6 月 20 日 14 時 18 分 49 秒から 2007 年 6 月 20 日 14 時 95 分 49 秒までのものであり、6 ページあるが、そのうち最初の 5 ページが提示されていて、2 番目のページがパターンと一致している。検出されたスレッドのパターンが一致している箇所へ（から）辿った Web ページが図 3.2 から分かる。一般の履歴検索インタフェースでは、このような曖昧な入力はサポートされていない。

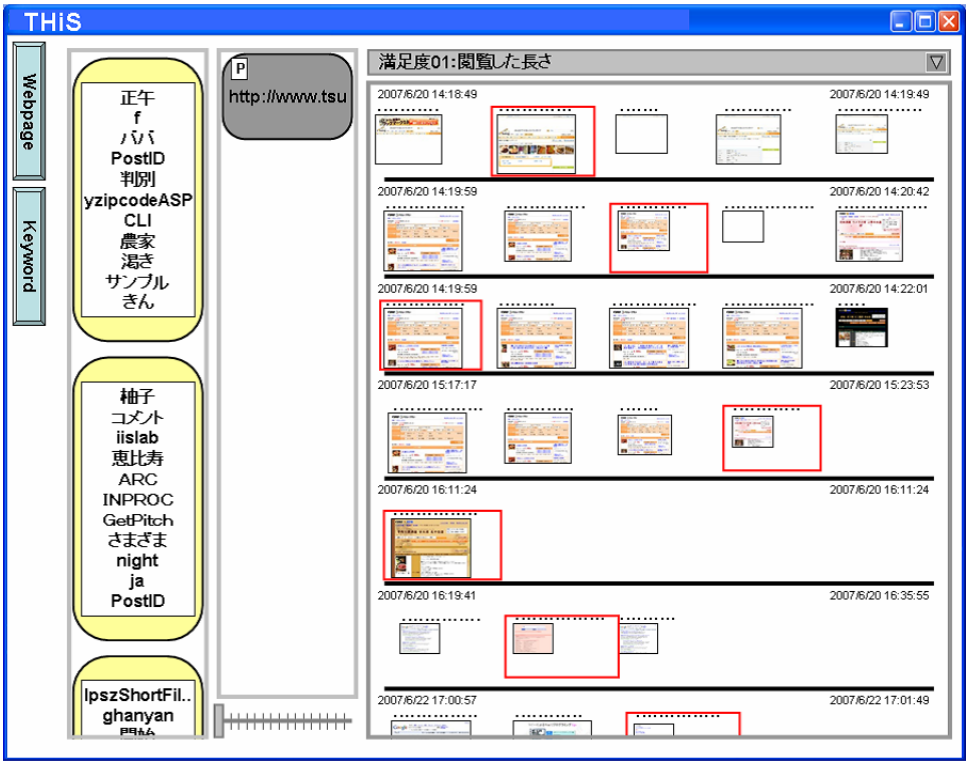

図 3.2: 使用例 1（クエリパーツ：1 つ）

さらに、ユーザは検出されたスレッドのパターンが一致されている個所の、次に閲覧した Web ページにぐるなびのページが多いことから、検索結果の中からぐるなびの Web ページを一つ、検索クエリ作成部にドラッグ＆ドロップしたとする。この時、履歴のパターンである B が検索クエリに追加される。システムは 2 つの連続したパターンを含むスレッドを検出して提示する。

図 3.3 は、システムに図 3.2 の状態から履歴から作成された Web ページクエリパーツを一

つ追加した例である。検索クエリは A → B であり、検索結果はパターン AB と完全一致する部分をもつスレッドとなっている。検索クエリと一致する部分が赤い枠でハイライトされているが、図 3.2 と違い、枠が 2 ページを囲んでいることが分かる。

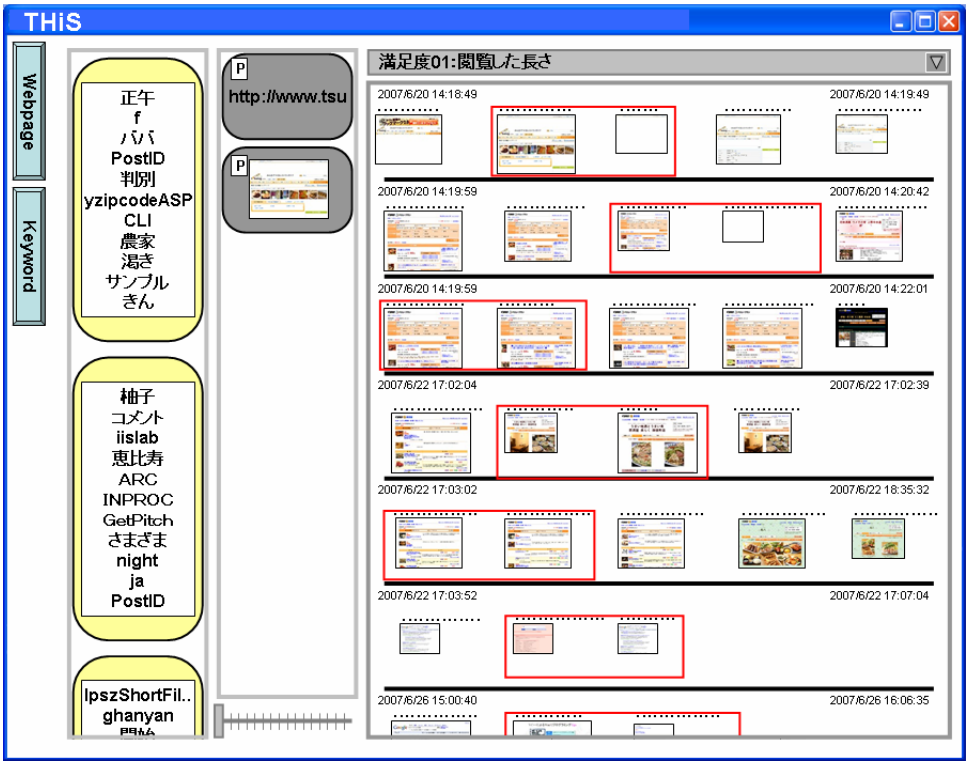

図 3.3: 使用例 2（クエリパーツ： 2 つ）

そして、ユーザは検索結果を絞り込むために、キーワード群の中で東京に近いキーワード「恵比寿」を含むものを追加する。この時、選択されたクエリパーツ用キーワード群により、検索クエリ作成部にパターンクエリパーツが追加され、キーワード群からなるパターン（C）がクエリに追加される。システムは 3 つの連続したパターンを含むスレッドを検出して提示する。図 3.4 は、システムに図 3.3 の状態からパターンクエリパーツを一つ追加した例である。検索クエリは A → B → C であり、検索結果は、パターン ABC と完全一致する部分を持つスレッドとなっている。3 つから成るパターンによってさらに検索結果が絞られたことが分かる。

さらに、ユーザは探している店が居酒屋であることを検索結果から思い出し、「居酒屋」というキーワードをもつキーワードっクエリパーツを検索クエリ作成部に追加し、クエリに追加する。この時、検索クエリ作成部にキーワードクエリパーツが追加され、キーワード「居酒屋」を基にしたパターン（D) がクエリに追加され、システムは 4 つの連続したパターンを含むスレッドを検出して提示する。

図 3.5 はその結果の図であり、パターン（ABCD）と完全一致する 4 ページ分がそれぞれ

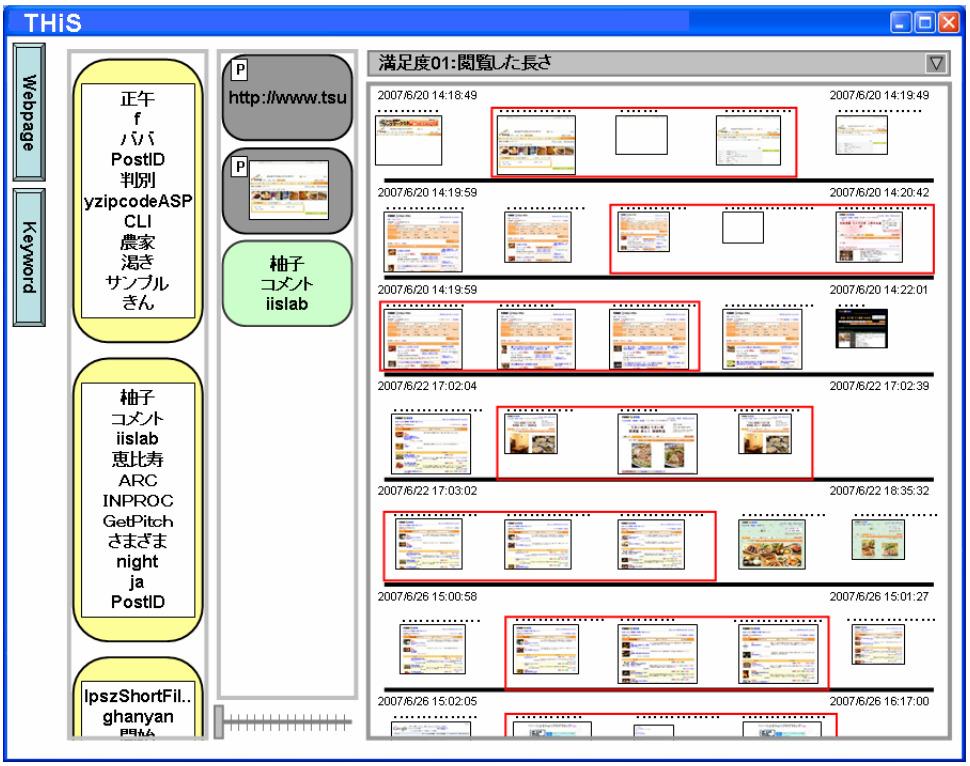

図 3.4: 使用例 3（クエリパーツ： 3 つ）

のスレッドでハイライトされている。また検索結果が絞られていることも分かる。本インタフェースを使うことで、ユーザは検索しながら、ぐるなびからリンクを辿った Web ページで東京近辺の居酒屋を探していたことを思い出すことができ、目的の店のページ尾をもう一度見つけることができるようになる。また、検索クエリ内のクエリパーツの順序をいれかえたり、パターン一致検索精度を下げるなどすることでより多くの情報を得ることもできる。

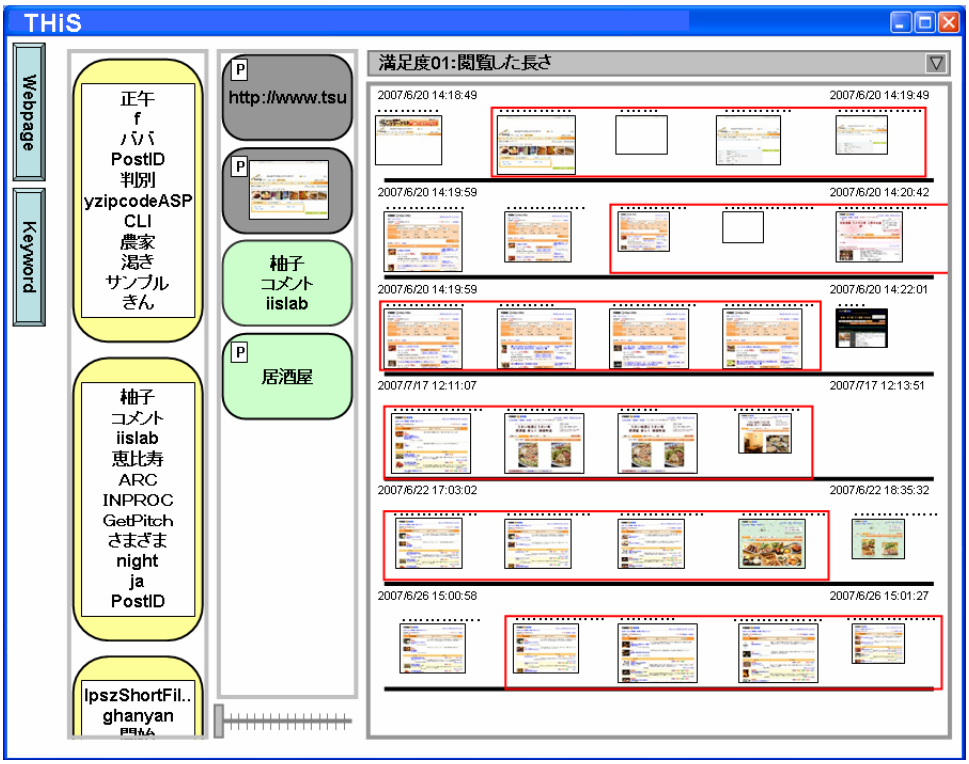

図 3.5: 使用例 4（クエリパーツ：４つ）

第4章　関連研究

Web 履歴を扱う研究として、安川らの研究 [12] があげられる。彼らの研究では、個人用のプロキシを用いて、ローカルにユーザが閲覧した（ブラウザが読み込んだ）ページと URL、参照日時、タイトル、先頭文字列、キーワード、ハッシュ値を保存し、再利用が可能なようにアーカイブとメタデータという形で保存している。検索インタフェースはないが、ローカルに Web ページを保存する手法が本研究と似ている。しかし、保存するタイミングがブラウザが読み込んだ時にした場合、複数ブラウザを立ち上げている時や、タブブラウザで閲覧している時、ページやブラウザを切り替えたことまで取得することができない。本研究は、フォーカスが最前面に来た時に保存するため、ユーザの記録をより鮮明にとることができると考えられる。

同様にブラウザが読み込んだページを保存して活用する研究として、西本らの研究 [2] と森田らの研究 [1] がある。前者は、推薦システムであり、ユーザが閲覧中の Web ページから辿ったページを提示している。後者は、検索システムであり、ユーザが集中して作業した期間を検索することができ、検索結果に閲覧時刻や回数、印刷の有無なども合わせて表示している。前者は、Web ページとリンクを検索のキーとしているところが似ているが、本研究とはリンクを用いて履歴を分ける点で類似しているが、推薦と検索という観点の違いがある。後者は、時刻や回数などから重要度を提示する観点は同じだが、期間ではなくリンクによって Web ページを分けている点で異なる。また、キーワードだけでなく Web ページや、キーワード群などでより曖昧な記憶を基にでも検索できるところが違うと言える。

履歴を検索するインタフェースとして、Google Web History[6] があげられる。これは、ブラウザが読み込んだページの URL と時刻のみを記録し、キーワードもしくは日時から検索を行うものである。しかし、キーワードや日時などの明確な基準を想起することは過去のものであればあるほど難しく、また URL のみの保存のため、Web ページが閲覧時の内容と変わっている可能性がある。本研究はユーザが閲覧した Web ページをローカルに保存している点で異なると言える。

Web ページだけにとどまらず、デスクトップ上で動作しているアプリケーションを含めた履歴をとり、その履歴を検索するインタフェースについての研究がある。暦本氏 [13] は、変更があるたびに履歴をとり、時間とキーワードによって検索することができる。この研究が、履歴を時系列一つにおいて考えているのに対し、本研究はスレッドという概念を使っている点で異なると言える。また近藤 [11] らの研究は、状態をスクリーンショットという形で定期的にとったものを日付から検索できる。

村上らの研究 [4, 14] では、専用のブラウザを作成し、ブラウザが読み込んだページを記録

し、その履歴の文章と画像によって情報を整理する。そして、その内容からさらに履歴を検索するものである。本研究は、スレッドの概念を用いている点が異なると言える。

永井らの研究 [7] では、検索時に閲覧してきたページからキーワードをハイライトして提示している。推薦と検索という観念が違い、本研究は履歴として全ページをとっている点が異なるといえる。

白井らの研究 [9] は、過去に閲覧したページを保存し、閲覧中のページと内容が類似している、または時間的に近いものを推薦している。本研究は、スレッドの概念を用いている点で異なるといえる。

Emmanuel Frécon らの研究 [15] は、ユーザが閲覧した Web ページを保存し、リンクを基に 3D で提示している。ユーザは、ブラウジングしながらページ間をつないでいる線を辿りながら探していく。本研究は、提示手法が 2D である点と、閲覧時間などで満足度を設定している点で異なるといえる。

第5章　検証実験

本手法における満足度の妥当性はそれぞれの値、閲覧時間、閲覧回数、文字数が妥当であるかどうかにかかわる。さらに、スレッドの妥当性は、そのスレッド内のＷｅｂページ数、スレッド内の最大閲覧時間等のパラメータが妥当であるかどうかにかかわる。

ここでは、著者の 2006 年 6 月 7 日 16 時 21 分 39 秒～2006 年 12 月 1 日 12 時 32 分 18 秒まで、及び 2007 年 6 月 19 日 16 時 3 分 24 秒～2007 年 12 月 14 日 17 時 24 分 47 秒までと研究室の同僚 A の 2007 年 6 月 20 日 13 時 1 分 51 秒～2007 年 10 月 10 日 20 時 16 分 33 秒の自由な利用によるＷｅｂ閲覧記録を調べた。

履歴総数はそれぞれ、6884 ページ、7147 ページであり、生成されたスレッドの数は 2993 個、2930 個である。スレッド内の最大ページ数はそれぞれ、47 ページ、33 ページであり、平均スレッド内ページ数はそれぞれ、およそ 2.3 ページ（6884÷2993）、およそ 2.4 ページ（7147÷2930）である。抽出されたキーワードの数は 160055 個である。また、最大閲覧時間はそれぞれ、3570 秒、3540 秒であり、このうち、平均閲覧時間は 103 秒、69 秒であった。同一 Web ページの最大閲覧回数は 247 回、264 回であり、平均閲覧回数はそれぞれ、およそ 1.9 回（6884÷3546）、およそ 2.2 回（7147÷3182）であった。最大文字数はそれぞれ 468314 文字、616848 文字であり、平均文字数はそれぞれおよそ 68 文字（468314÷6884）、およそ 86 文字（616848÷7147）であった。

図 5.1～図 5.3 はそれぞれ閲覧した長さ、閲覧した回数、文字数ごとのページ数である。各図において、赤は著者、青は同僚 A のデータである。閲覧時間の図を見ると、両者とも 0 から 500 秒の間におよそ 50 %の履歴が存在し、50 %が 500 秒以上に散らばっている。そのため、閲覧時間でサムネイルの大きさを決定した場合、全てが同じサイズであったり、異なるサイズであるような一目でわからなくなるという結果にはならない。

また、閲覧回数の図を見ると、大きく三つに分類することができる。100 回まで、200 回近辺、250 回近辺である。明確に各ページ間に差異が現れるため、ページの評価の基準となりうると考えられる。

そして、文字数の図を見てみると、著者のデータに関して言えば、3 分の 1 が 200000 以上であるが、同僚 A のデータではほとんどが 0～10000 以内にあることが分かる。このため、ページの評価の基準とするにはデータ間に差異がないため、使いづらいといえる。

図 5.4 はスレッドと各スレッド内のページ数の関係を表している。横軸にページ数、縦軸にスレッド数を置いている。スレッドを生成した結果、およそ 3 分の 1 がページが一つしかないスレッドであり、3 分の 1 がページ数 2 である。そして、その後に 3 ページ、4 ページの順にだんだんと少なくなっているのが分かる。ページ数が 1 であるスレッドにはブラウザを起

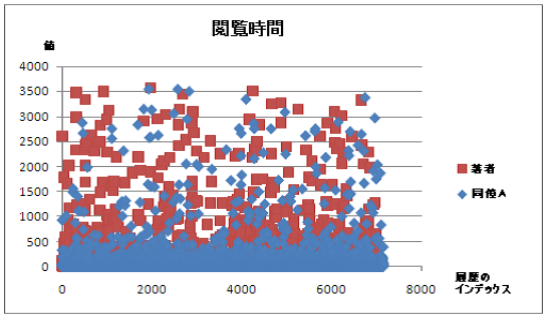

図 5.1: ページと閲覧時間

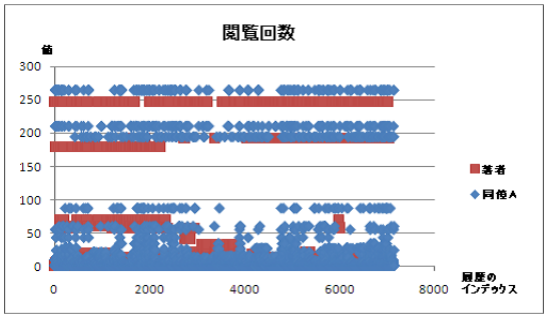

図 5.2: ページと閲覧回数

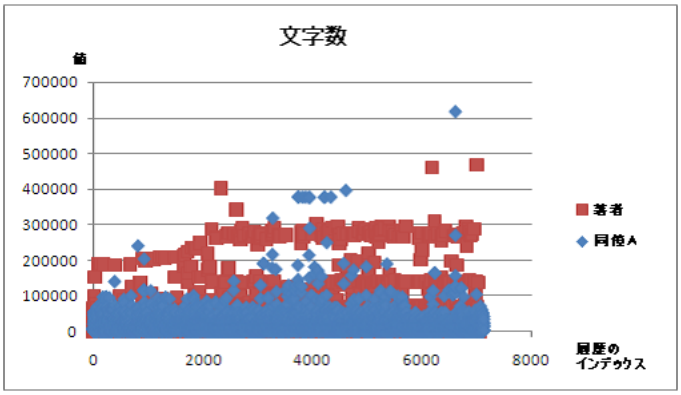

図 5.3: ページと文字数

動した際に最初に表示されるスタートページのみのスレッドも含まれる。本インタフェースでは連続したパターン、もしくはその遷移をもって、スレッドを検出するが、2 ページ以上のスレッドが全履歴の 3 分の 2 以上存在するので、本研究で提案しているスレッドの概念は有効であるといえる。

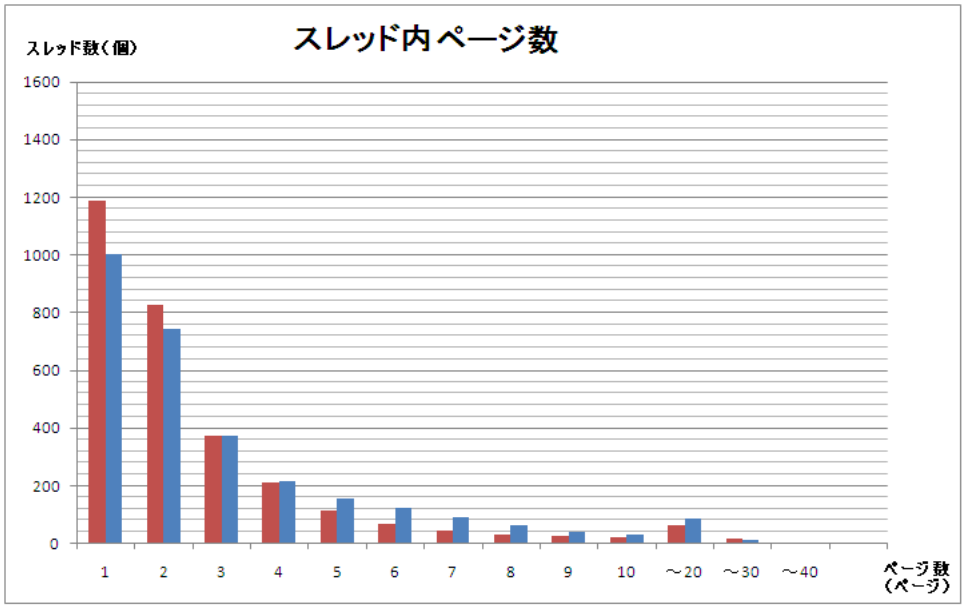

図 5.4: スレッドとページ数

次に、本研究で提案している満足度について実測した。満足度（1）と（2）に関しては、前述で検証しているので省略する。（3）同一 URL において何回目の閲覧であるかという値であるが、これは同じページでも情報が更新され、何度も訪れているのであれば、その情報のほうが満足したのではないかと推測した。図 5.5 を見ると、横軸に時系列、縦軸に閲覧回数を置いているが、右上に向かった線と 0 から 50 回に収まる線に分けることができる。新しく何回も見ているページほど、値が大きくなっているので、ユーザの評価になりうると思われる。

（4）の閲覧時間に閲覧回数を掛け合わせた値（図 5.6）と、（5）の閲覧時間と何回目の閲覧であるかという値を掛け合わせた値（図 5.7）は、両方とも両者ともページのほとんどが下部にあるが、1 %ほどが上の値に散らばっているのが分かる。繁雑に全体を見るときには基準となりえないが、検索結果から絞り込む際には有効な値になるのではないかと考えられる。

そして、図 5.8、図 5.9、図 5.10 が表している満足度（6）、（7）、（8）の値だが、ほぼすべての値が下部にまとまっており、ページの評価の値になりえないだろう。値のちいさな（1）、（2）、（3）を値のおきな文字数で割ったがためにこのような結果になったと思われる。

（9（図 5.11））に関しては、（4）と似た値となっているが、さらに絞り込む際に使用できると考える。しかし、（10（図 5.12））に関しては前述の 3 つと同様ページの評価にはなりえない値である。

（11（図 5.13））と（12（図 5.14））は図においてそれぞれ（2）、（3）と似た形になってい

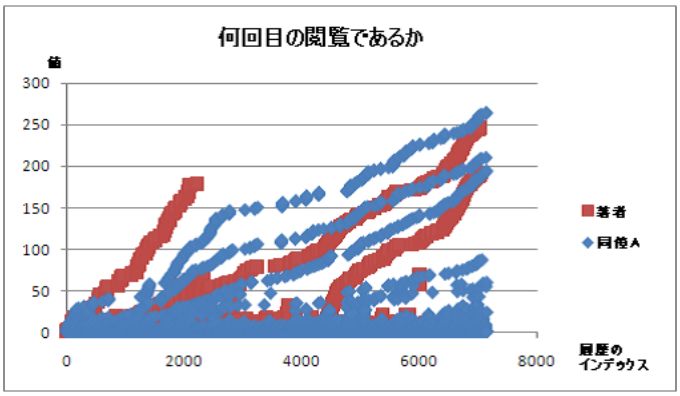

図 5.5: 同一 URL において何回目の閲覧であるか

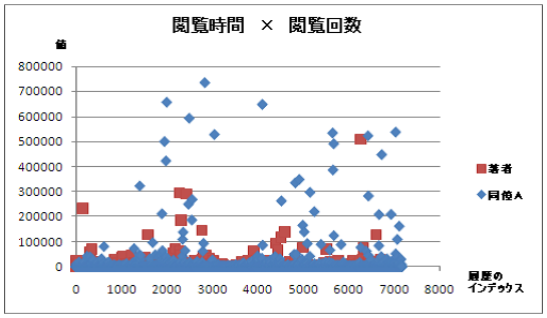

図 5.6: 満足度（4）

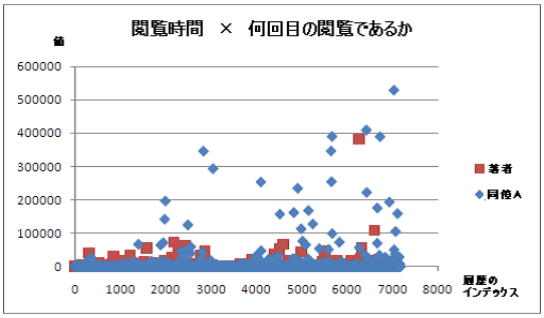

図 5.7: 満足度（5）

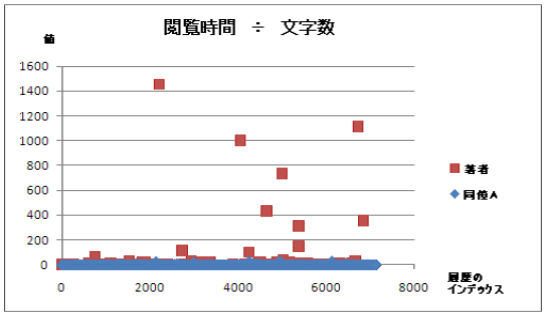

図 5.8: 満足度（6）

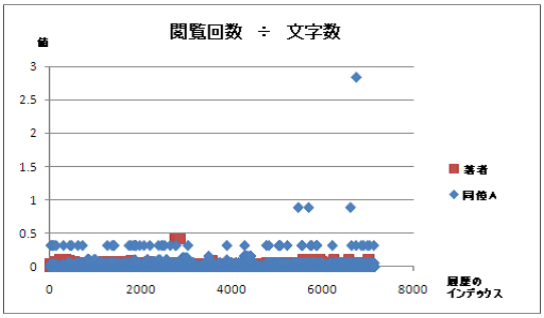

図 5.9: 満足度（7）

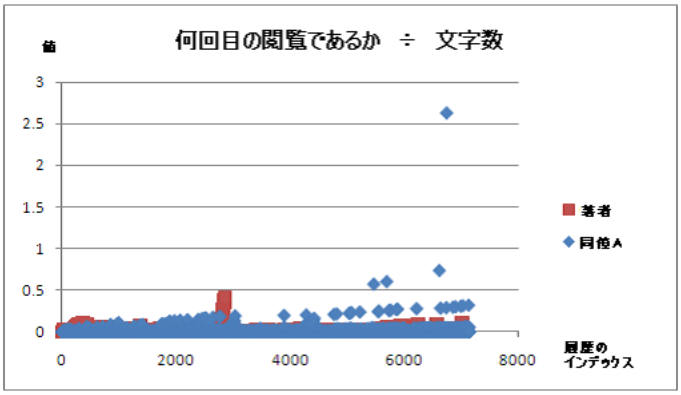

図 5.10: 満足度（8）

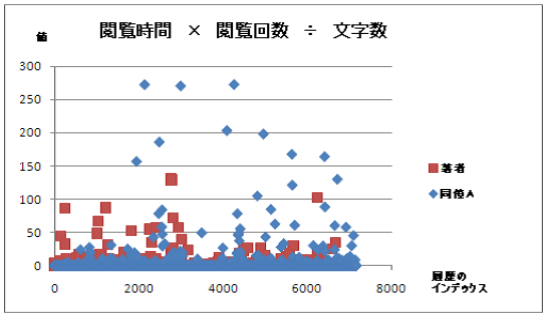

図 5.11: 満足度（9）

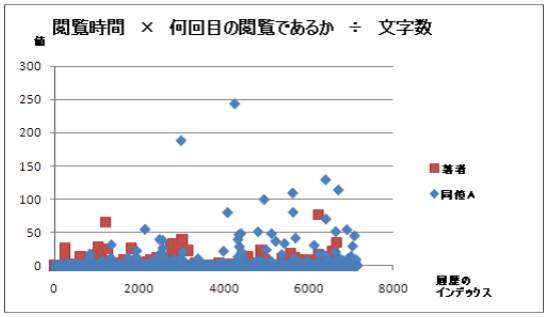

図 5.12: 満足度（10）

るが、著者と同僚 A のデータでは最大値が異なるという結果が出ている。これはブラウジングの個々のスタイルの差が出たのではないかと考えられる。

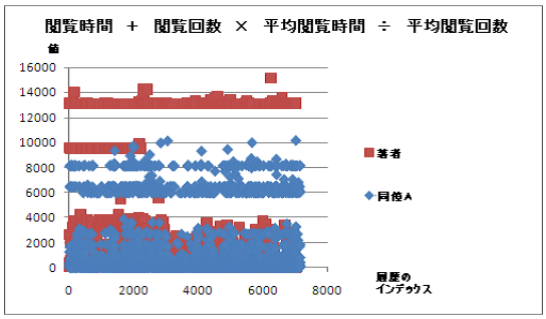

図 5.13: 満足度（11）

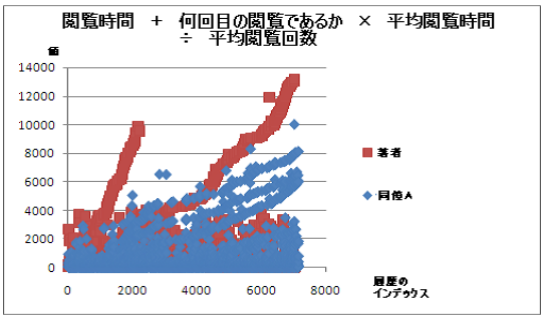

図 5.14: 満足度（12）

（13（図 5.15））と（14（図 5.16））は（8）、（9）、（10）と同様、値にばらつきがほぼないので、ページの評価とはなりえないだろう。

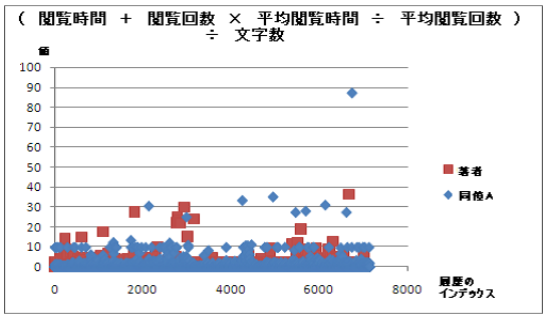

図 5.15: 満足度（13）

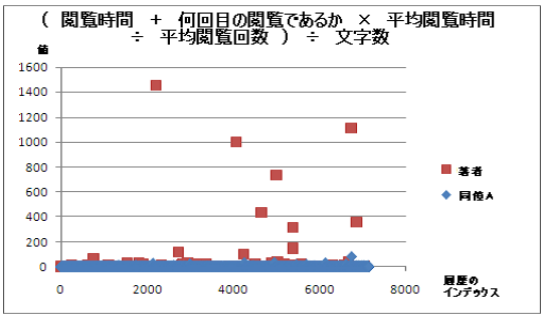

図 5.16: 満足度（14）

第6章　考察

本インタフェースの検索クエリ作成部は曖昧な入力に対応するため、入力できるものをキーワード、URL から Web ページ、キーワード群まで拡大してある。しかし、一番記憶があいまいである最初において、キーワードや URL、キーワード群、Web ページよりもさらに曖昧な入力ができることが好ましい。また、いくつか候補を提示することによって、ユーザの入力を支援することも考えられる。現在は、一番古い履歴が最初に表示されているが、ランダムでスレッドを選ぶか、異なるパターンの Web ページで始まるスレッドを選び提示することも考えられる。サムネイルをクリックすることで、Web ページを閲覧することができるが、本インタフェースのみで目的の Web ページを見つけだすようにするために、提示する情報のすべてが検索クエリになりうるようにするべきである。

また、ユーザの満足度については、前章の検証実験で（1）～（14）までの値について検証してあるが、（15）～（28）について検証を行っていないので実際にデータ検証を行うべきである。その中で、前章において文字数で割った場合、それぞれの値における差異が小さくなってしまうため、（20）～（24）、（27）、（28）は考慮する必要性はない。さらに、履歴として保存しておく対象を、Web ブラウザの状態（表示）や、マウスの移動、スクロール、キーボードによって打ち込まれた文字列、Web ブラウザの細かな動作（印刷など）も取得することで、よりユーザの満足度を確かなものにすることができる。ユーザの視線情報や思考までは測ることができないため、ブラウジングにおけるユーザの動作をすべて記録することはできない。限られた情報の中でどれだけユーザのニーズにこたえることができるかがこれからの課題である。

本インタフェースの表示部では、検出したスレッドの日付、タイトル、サムネイルを提示しているが、ユーザは Web ページを全体ではなく一部を覚えていることを考え、最も大きな画像を提示することも考えられる。さらに、現在、履歴の保存を Web ページ単位で行っているが、記事もしくは段落や情報単位で保存し、提示の際も情報単位で行うことでよりユーザの目的にかなった検索ができるようになる。

本研究では、履歴をクラスタリングする際、名詞と未来語によって行っているが、その 1 ページにおける単語量も差ながら、全体における単語数も多い。より長期間使うことで、さらに量が増大していくことと、現状として、前処理にかなりの時間を要することを考えると、ベクトルにする際に単語をより選別することも考えられる。また、Web ページを保存する際にも時間が 1～2 秒かかること、まったくの同一ページが何個も保存されることがあり、同一ページの場合の処理を考慮する必要があり、今後の課題となる。

第7章　まとめ

ユーザの曖昧な記憶からの入力に対応した Web 閲覧履歴検索手法を開発した。まず、履歴の表現方法として「スレッド」を開発した。これは、ユーザがリンクを辿る行為を一連の動作として捉えた表現手法である。クラスタリングとスレッドにより、履歴検索をより曖昧な入力で、Web 閲覧履歴を検索できるようになった。

謝辞

本研究を進めるにあたり、指導教官である田中二郎教授に様々な助言やご指導をいただきました。深く感謝いたします。また、志築文太郎講師にはグループミーティングなどにおきまして適切なご指導をいただきました。深く感謝いたします。有益な助言、ご指導を下さった三末和夫准教授並びに高橋伸講師に心から感謝いたします。また、田中研究室の皆様及びWAVE グループの皆様には研究の全般にわたり丁寧なアドバイスをいただきました。本当にありがとうございました。最後に、私を支えて下さった家族や友人に改めて感謝いたします。

参考文献

[1] 森田哲之, 日高哲雄, 田中明通, 加藤泰久. 記憶想起支援ツール『Memory-Retriever』. インタラクション 2007 論文集, pp.173-174, 2007.

[2] 西本一平, 戸田真志. Web リソースを対象としたリファインディング支援システム. インタラクション 2007 論文集, pp.63-64, 2007.

[3] 大澤亮, 高汐一紀, 徳田英幸. 俺デスク：ユーザ操作履歴に基づく情報想起支援ツール. インタラクション 2006 論文集, pp.219-220, 2006.

[4] 村上晴美, 平田高志. WWW からの情報獲得・整理支援-思考・興味空間ブラウザ-. 情報処理学会研究報告 Vol.2001, No.20, pp.167-174, 2001.

[5] 福田隼人, 松尾豊, 石塚満. ユーザ個人の閲覧履歴からのキーワード抽出によるブラウジング支援. 人工知能学会論文誌, Vol.18, No. 4E, pp.203-211, 2003.

[6] Google Web History. http://www.google.com/history/.

[7] 永井洋一. ユーザの閲覧履歴に基づくオンライン検索支援システム. 日本ソフトウェア科学会第 23 回大会論文集, pp.1- 9, 2006.

[8] 今枝誉明, 早瀬康裕, 松下誠, 井上克郎. 閲覧状態復元機能付き Web ブラウザの試作. 電子情報通信学会技術研究報告 SS, ソフトウェアサイエンス Vol.15, NO.25, pp.13-18, 2005.

[9] 白井良成, 中小路久美代, 山本恭裕. インタラクションヒストリによる Web ブラウジング拡張. インタラクション 2006 論文集, pp.223-224, 2006.

[10] 増井俊之, 塚田浩二, 高林哲. 近傍関係にもとづく情報検索システム. 第 11 回インタラクティブシステムとソフトウェアに関するワークショップ（WISS2003）論文集, pp.79-86, 2003.

[11] 近藤秀樹, 小出洋, 三宅芳雄. 活動履歴を活用するシステムの基本設計と暫時的開発. 情報処理学会論文誌 Vol.48, No.SIG12, pp.19-27, 2007.

[12] 安川美智子, 山田篤, 星野寛, 大瀬戸豪志, 上林彌彦. Web コンテンツの収集と再利用を支援する個人用アーカイブ. 情報処理学会研究報告 データベース・システム研究会報告, Vol.2003, No.5, pp.139-146, 2003.

[13] 暦本純一. Time-Machine Computing：時間指向ユーザインタフェースの提案. 第7回インタラクティブシステムとソフトウェアに関するワークショップ（WISS'99）論文集, pp.55-64, 1999.

[14] 村上晴美, 平田高志. 記憶を中心とする人生の記録 -ユーザの知識空間の作成による Web ブラウジングの履歴の想起支援-. 情報処理学会研究報告 人文科学とコンピュータ研究会報告, Vol.2004, No.7, pp.19-24, 2004.

[15] Emmanuel Frécon, Gareth Smith. WebPath - A Three Dimensional Web History. INFOVIS Proceedings of the 1998 IEEE Symposium on Information Visualization, pp.3-10, 1998.

[16] Andy Cockburn, Saul Greenberg, Steve Jones, Bruce Mckenzie, Michael Moyle. Improving Web Page Revisitation : Analysis, design, and evaluation. IT & Society., Vol.1, Issue 3, Winter 2003, pp.159-183, 2003.

[17] Capra, R., Pinney, M. and Pérez-Quiñones, M. Refinding Is Not Finding Again. Technical Report, Technical Report TR-05-10, Computer Science, Virginia Tech, pp.1-10, 2005.