

# ユーザの動作履歴を基にしたデータ間関連度とデータ着目度算出機構の構築

大澤 亮<sup>2</sup> 出内 将夫<sup>2</sup> 高汐 一紀<sup>2</sup> 徳田 英幸<sup>1,2</sup>

<sup>1</sup> 慶應義塾大学環境情報学部 <sup>2</sup> 慶應義塾大学大学院政策・メディア研究科

近年、コンピュータ技術の進歩に伴い、様々な分野で情報の電子化が行われてきた。情報とは文書や画像、映像、音楽のデータなどが該当し、ユーザは PC 上でそれらのデータを参照可能になった。その過程において、ユーザは一度閲覧したデータを再度閲覧したいと考える状況が想定される。しかし、そのような状況でユーザが目的のデータを閲覧履歴から探すのは困難である。そこでユーザがデータを閲覧した際の動作を基にデータ間関連度とデータ着目度を算出し、データ間関連度を用いた関連検索機能とデータ着目度によるソート機能を閲覧履歴の検索に付加しユーザの閲覧履歴検索を支援する手法を提案する。本研究では OS およびアプリケーションからイベントを取得し、データ間関連度とデータ着目度算出を動的に行うミドルウェア DMemFinder を構築し、有効に働くかを検証した。

## Data History Search Framework that use Relation Value between Data and Attention Value of Data based on User Behavior

Ryo Ohsawa<sup>2</sup> Masao Ideuchi<sup>2</sup> Kazunori Takashio<sup>2</sup> Hide Tokuda<sup>1,2</sup>

<sup>1</sup> Faculty of Environmental Information, Keio University

<sup>2</sup> Graduate School of Media and Governance, Keio University

Recently, the computerization of information that are document, movie, music, etc is progressing in various fields. Then user read many these data on the PC. In this process the situation that a user search his history data to reread a data exists. However, when the number of the history data increases, finding a data from history data becomes difficult. The purpose of this research is developing the technique for a user to find out a data from history data efficiently without the manual operation. This research proposes the technique that the relation value between data and the data attention value are calculated based on a user behavior. I implemented DMemFinder (Deep Memory Finder) middleware that dynamically calculate these values. Moreover I evaluated DMemFinder whether to work effectively.

### 1 はじめに

近年、コンピュータ技術の進歩に伴い、様々な分野で情報の電子化が行われてきた。情報とは文書や画像、映像、音楽のデータなどが該当し、ユーザは PC 上でそれらのデータを参照可能になった。その過程において、ユーザは一度閲覧したデータを再度閲覧したいと考える状況が想定される。しかし、そのような状況でユーザが目的のデータを閲覧履歴から探すのは困難である。データの閲覧履歴を時系列順に一覧表示したとしてもデータ数が膨大な場合、見つけるのは困難である。

そこでユーザがデータを閲覧した際に行った動作を基にデータ間関連度とデータ着目度を算出し、関連度を用いた関連検索機能と着目度によるソート機能をデータ閲覧履歴の検索に付加し、ユーザを支援する手法を提案する。関連検索とは、Google の関連ページ機能のようにあるデータに関連したデータを提供する手法である。既存システムの多くはデータの内容から関連度を算出しているが、本提案ではユーザがデータを閲覧した際に行った動作の履歴を基に関連度を算出

する。人間の記憶は連想からなる [1] ため、状況次第ではユーザのデータ閲覧履歴の検索を有効に支援できる。例えば、ユーザは検索システムに対して「Word ファイル A を作成していたときに主に参照していた Web ページおよび Excel データを検索」といった要求が可能になる。データ着目度とは、Google における Page Rank のようにデータの相対的な着目指標である。データ着目度はデータの閲覧時間や閲覧回数、選択文字列のコピーなどの様々な要素から算出される。データ着目度を用いることでユーザは過去に着目したデータを閲覧履歴の検索結果の上位にある程度表示できる。

本研究の目的は、普段のユーザに手動操作を要求しない手法で、ユーザが閲覧した履歴データの検索をより効率化することである。本研究では、ユーザがデータを閲覧した際に行った動作履歴を基にしたデータ間関連度とデータ着目度算出機構 DMemFinder (Deep Memory Finder) を構築する。DMemFinder は検索アプリケーションに対して、関連検索用 API および着目度算出用 API を提供する。DMemFinder は様々なア

アプリケーションから汎用的に利用でき、DMemFinder を利用したプラグインをアプリケーションごとに作成できる。本論文ではユーザがデータ閲覧履歴から過去に閲覧したデータを探す検索を閲覧履歴検索、ユーザがデータを閲覧していた際に PC 上で行った動作をユーザ動作と呼ぶ。

## 2 データ間関連度とデータ着目度

本研究では閲覧履歴検索を効率化するためにデータ間関連度を利用した関連検索とデータ着目度を利用したソート機能を利用する。本節ではまず一般的なデータ間関連度とデータ着目度算出手法と本研究の関連度を述べる。次にユーザ動作を基にしたデータ間関連度とデータ着目度算出手法の概要と本手法を提案する理由について述べる。

### 2.1 一般的なデータ間関連度とデータ着目度算出手法

#### 2.1.1 データ間関連度

既存手法の多くはデータ内容を解析し、データ間の関連性を算出する。データ内容を基にしたデータの関連性算出手法はそのデータ形式によって異なる。テキストデータの場合はデータに含まれる用語の使われる頻度や文脈からデータ間の関連度を算出する。画像や音声、動画といったバイナリデータの場合、そのフォーマット共通の属性情報を利用し関連度を算出する。画像を例として挙げると、物体の配置、形状、色特徴から属性情報が抽出される。画像、映像、音声などのマルチメディア情報においては、属性情報共通規格として MPEG-7[2] が定められている。本研究ではユーザの動作履歴を基にした検索を対象としており、本節で述べた手法と並列してユーザの検索を支援できる。

#### 2.1.2 データ着目度

データ着目度とはユーザが過去に見たデータの主観的な着目度合いである。着目度を保持するシステムには UJIKO[3] や My Yahoo! Search[4] などが挙げられる。これらのシステムを用いることでユーザは検索結果のデータへのレーティングができ、次回以降の検索に役立てられる。例えば、検索結果をレート順にソートしたり必要のないデータを次回以降の検索対象から除外できる。しかし、この手法を用いるにはユーザが手動でデータのレーティングをしなければならず、参照量が膨大になればなるほどユーザの負担が大きくなる。本研究では普通のユーザに手動操作を要求しない手法で、閲覧履歴検索をより効率化することを目的とする。

### 2.2 ユーザ動作を基にしたデータ間関連度とデータ着目度算出手法

#### 2.2.1 データ間関連度

ユーザ動作を基にした関連度を用いることで、検索システムはユーザに対して個々人の参照記憶を基にした関連検索を提供できる。例えば、ユーザは検索システムに対して「Word ファイル A を作成していたとき

に参照していた Web ページ」といった要求ができる。人間の記憶は連想からなるため、閲覧履歴検索においてはユーザの動作履歴を基にした手法は有効である。

#### 2.2.2 データ着目度

本研究ではユーザ動作を基に着目度を算出する手法を用いる。ユーザ動作を基に着目度を算出する手法は、文章内容ではなくユーザの主観に基づいた着目度算出ができる。文章内容から算出される着目度とユーザが感じた着目度は異なるため、閲覧履歴データの着目度算出においてこの手法は有効である。

また、付加的機能として、ユーザのフィードバックを基に要素と着目度の相関係数を求め動的に重みを調整する機構を持っている。例えば、ユーザが高レーティングに設定したページとユーザの閲覧時間の相関係数が高かった場合、データ閲覧時間とデータ着目度の相関が高いとする。

#### 2.2.3 関連研究

データ間の関連性算出を自動で行う先行研究として、大阪市立大学の前田氏による興味空間ブラウザ [5] が挙げられる。興味空間ブラウザはユーザが閲覧した Web ページ中の語句を数量化 3 類を用いて解析し、Web ページ間の関連性を算出する。本研究では、ユーザ動作を基に関連性を算出しデータ中の語句解析を行っていないため、語句解析手法と並列拡張できる。

データの着目度算出を自動で行う先行研究として、関西大学の原田氏による Web 検索履歴を用いたブラウジング支援ツールが挙げられる [6]。このツールは、ユーザが過去に閲覧した Web ページの着目度を閲覧回数、閲覧日時、参照時間から算出する。しかし、全てのユーザに対して必ずしもこの要素が着目度と相関があるとは限らない。本研究では、ユーザのフィードバックを基に要素と着目度の相関係数を求め動的に重みを調整する機構を持っており、その点でこの手法とは異なる。

## 3 DMemFinder

本節では、前節で述べたユーザ動作を基にしたデータ間関連性とデータ着目度算出手法を具体的に実現したミドルウェア DMemFinder について述べる。

DMemFinder の概要を図 1 に示す。DMemFinder は OS とアプリケーションのイベントを取得し、データのメタ情報として保存する。そして、アプリケーションからの要求に応じて、メタ情報データベースからデータ間関連度とデータ着目度を算出し、アプリケーションに提供する。データ間関連度とデータ着目度を用いることでデータ検索アプリケーションは、関連検索と着目度によるソート機能をもった検索をユーザに提供できる。

### 3.1 イベントの取得

DMemFinder は OS の起動と同時にデーモンとして立ち上がり、アプリケーションおよび OS のイベン

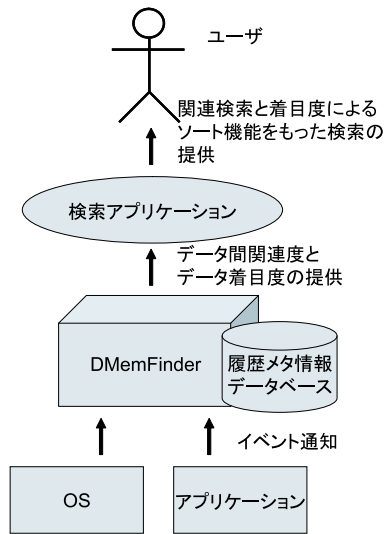


図 1: DMemFinder

トをフックする。DMemFinder はこれらのイベント情報を現在ユーザが閲覧しているデータのメタ情報として保存する。

### 3.1.1 OS のイベント

OS のイベントとはユーザがデータを参照した動作を基にした OS のイベントをフックして得られるイベントである。具体的にはデータへのロード・アンロード、クリップボード利用、選択文字列反転、テキスト内検索などが該当する。ユーザが席を離れているかどうかは、マウスとキーのイベントを監視し一定期間以上入力がなかった場合、席を離れているとし、参照時間から除外する。これらのメタ情報を用いるによりユーザは過去のデータアクセスイベントを基にした検索ができる。例えば「文書 A を作成中に、クリップボードのコピー先として利用していた Web サイトを検索」といったことが可能になる。

### 3.1.2 アプリケーションイベント

アプリケーションイベントとは「メッセージャーで同僚 A に話かけられた」や「音楽 B を再生していた」などアプリケーション固有のイベントが該当する。これらのイベントを取得するにはアプリケーションごとにプラグインを作成しなければならない。アプリケーションイベントを用いることで、「同僚 A と話していたときに参照していたファイルを検索」といったことが可能になる。

## 3.2 データ間関連度の算出

DMemFinder を利用した検索アプリケーションがユーザに関連検索を提供する様子を図 2 に示す。

DMemFinder は参照時刻、テキスト内検索、クリップボードの利用履歴など複数の要素を基に関連度を算出し、検索アプリケーションに提供する。参照時刻の場合、近い時刻に参照していたデータは関連が深いと

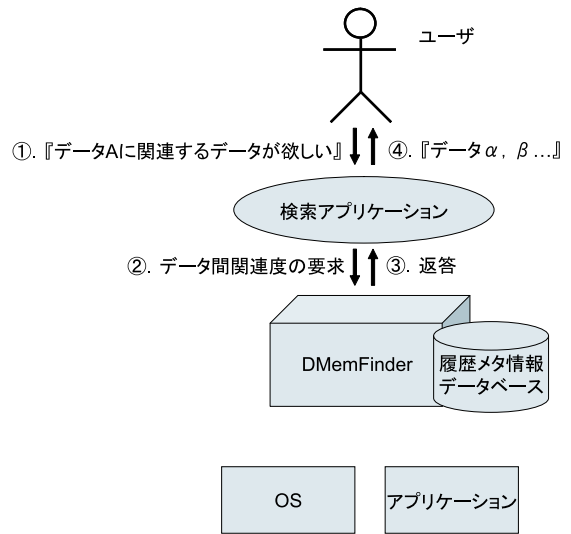


図 2: 関連検索

する。具体的には、ユーザがデータ A にアクセスした時刻が

$$(a_1, a_2, \dots, a_n) \quad (n \text{ は正の整数})$$

であり、データ B にアクセスした時刻が

$$(b_1, b_2, \dots, b_m) \quad (m \text{ は正の整数})$$

のとき、データ A, B 間の関連度  $R_{ab}$  は

$$R_{ab} = \frac{\sum_{k=1}^n \min(|a_k - b_1|, |a_k - b_2|, \dots, |a_k - b_m|)}{n}$$

で算出する。また、同語句を検索したデータ同士は関連が深いとする。さらに、データ A でクリップボードにコピーし、データ B にペーストした場合、データ A と B は関連が深いとする。これら複数の要素を組み合わせ、最終的な関連度を算出する。

## 3.3 データ着目度の算出

DMemFinder を利用した検索アプリケーションがユーザに着目度を利用した検索を提供する様子を図 3 に示す。

データ着目度は参照時間、参照回数、クリップボードの利用、選択文字列反回数などを代表とする様々なユーザ動作から算出する。しかし、これらの値が常にユーザにとっての着目度と関連があるとは限らない。例えば、あるユーザは重要な資料に対して選択文字列反転に相関があったとしても、別のユーザも相関があるとは限らない。そこで本機構では付加的機能として、ユーザのフィードバックを基に要素と着目度の相関係数を求め動的に重みを調整する機能を加える。ユーザはデータに対して着目度を入力できるようにし、入力された着目度と自動に取得したユーザ動作との相関を

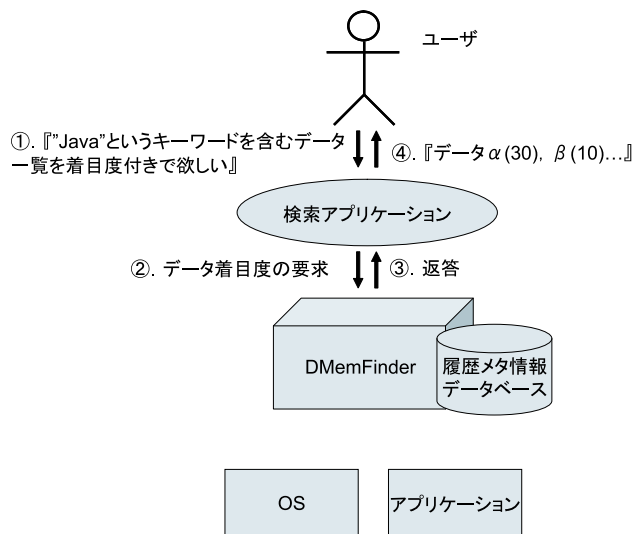


図 3: 着目度を利用した検索

求める。例えば、ユーザ A と B において入力したデータ着目度と選択文字列反転回数の分布が図 4 になったとする。

この時、ユーザ A と B の着目度と選択文字列反転回数の相関係数を算出すると

ユーザ B の相関係数 > ユーザ A の相関係数

となり、ユーザ B の方が選択文字列反転回数と着目度の相関が強い。この場合、DMemFinder はユーザ B の選択文字列反転回数からデータ着目度を求める際の重みをユーザ A より上げる。

## 4 実装

本節では DMemFinder のプロトタイプ実装について述べる。

### 4.1 実装環境

プロトタイプ実装では DMemFinder が監視するアプリケーションとして、Mozilla Firefox[7] と OpenOffice.org[8] を用いた。また、データ検索アプリケーションとして DMemSearch を実装した。DMemSearch はユーザに対して関連検索や着目度順検索を提供する。DMemFinder とそれぞれのアプリケーションの関係を図 5 に示す。

また、開発は Windows 上で主に Java を用いて行い、一部 Windows 特有の機能を利用するために VisualC++ を利用した。さらに、DMemSearch のユーザインターフェースは軽量化のため、Eclipse.org[9] の SWT(Standard Widget Toolkit)[10] を用いた。実装環境を表 1 にまとめる。

### 4.2 DMemSearch

本節では作成したデータ検索アプリケーション DMemSearch のユーザインターフェースについて述べる。

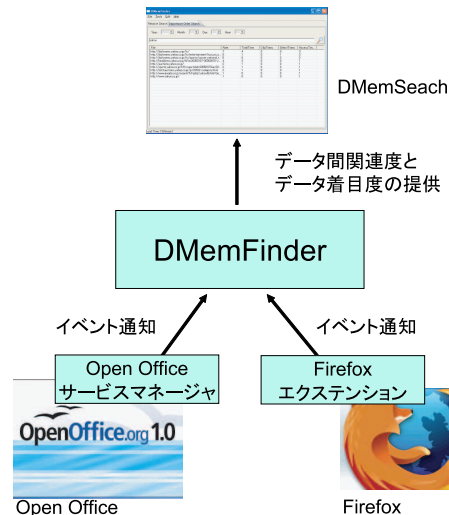


図 5: DMemFinder とアプリケーションの関係

表 1: 実装環境

対応 OS	Windows 2000/XP
実装言語	Java, VisualC++
対応アプリケーション	Firefox, OpenOffice
フックする OS のイベント	ウィンドウフォーカス クリップボード マウス, キーボード ファイルアクセス
使用ライブラリ	Eclipse SWT

#### 4.2.1 関連検索機能

DMemSearch が関連検索を提供する際のユーザインターフェースを図 6 に示す。ユーザはファイル名または Web サイトの URL 名を入力することで、そのファイルまたは Web サイトに関連のあるデータ一覧を得られる。データは関連度が高く着目度の高いページが上位に表示される。

#### 4.2.2 着目度順検索機能

DMemSearch が着目度順検索を提供する際のユーザインターフェースを図 7 に示す。ユーザは探したいデータに含まれるキーワードを入力することで、そのキーワードを含む Web サイトおよびファイルを含むデータを着目度順に表示できる。また、検索の際、日付・時刻を指定するとその時刻に近い時間にアクセスがあったデータの着目度が上昇する。

## 5 検証と評価

本節では、DMemFinder の有用性検証とパフォーマンス評価を行う。

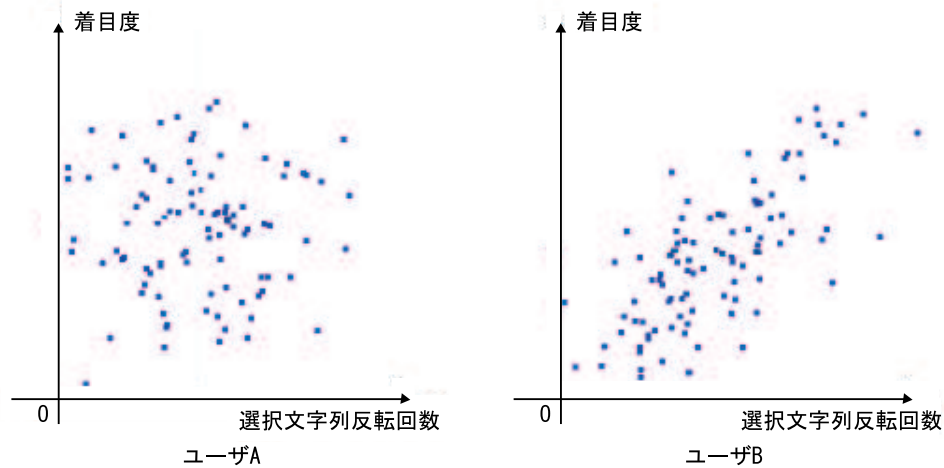


図 4: ユーザ動作を基にしたデータ間関連性とデータ着目度算出機構

File	Rate	TotalTime	ClipTimes	SelectTimes	AccessTime
1 http://www.google.co.jp/	2730	545	1	2	32
2 http://www.yahoo.co.jp/	2000	0	0	0	2
3 http://dailynews.yahoo.co.jp/	322	33	0	78	19
4 http://www.ht.stc.keio.ac.jp/tyo/dmem/wiki/index.php...	33	787	1	108	9
5 http://www.ht.stc.keio.ac.jp/tyo/dmem/wiki/index.php...	29	2	0	0	2
6 http://global.yesasia.com/ProdDept.asp?pid=100388...	28	741	0	9	1
7 http://xri.jblob.org/archives/0012031mifre...	21	179	0	445	3
8 http://pc52ch.net/test/read.cgi/pcg/109377042/...	7	52	0	105	4
9 http://xri.jblob.org/archives/001322.html	6	20	0	154	1
10 http://xri.jblob.org/archives/2006_01.html	5	29	1	125	1
11 http://f31.aaa.livedoor.jp/coffee/cdp3/italy.htm	5	24	0	146	2
12 http://www.thissys.net/report/024/	4	10	0	95	8
13 http://s.wiki.jpedia.org/wiki/%E7%BB%9C%E8%96%BE%E...	4	12	0	95	1
14 http://www.w3.org/TR/1999/WAI-WEBCONTENT-1999...	3	14	0	81	1
15 http://xri.jblob.org/	3	19	0	64	30
16 http://www.sak-stake-adb.co.jp/kyo/enterbu/abuse/la...	2	7	0	56	1
17 http://www.stc.wide.ad.jp/re/2004/attendance/check.p...	2	62	0	0	1
18 http://sashd01.jp/poll/00hp/gips-208&as=3	2	71	0	0	1
19 http://xri.jblob.org/archives/0013261mifre...	2	13	0	51	1
20 http://www.4gamer.net/review/live/dp.html	2	14	0	51	1
21 http://www.sfc.wide.ad.jp/re/2004/attendance/check.p...	1	71	0	0	1
22 http://headlines.yahoo.co.jp/hl?c=20050111-00000111-...	1	5	1	19	3
23 http://sports.yahoo.co.jp/hl?c=sports&d=20050111&a=2...	1	9	0	36	2
24 file:///C:/Documents%20and%20Settings/yo/Application...	1	19	0	0	12
25 http://www.google.co.jp/search?q=MC764H4G0C8067&l...	1	12	0	16	5
26 http://www.google.co.jp/search?q=WC+KEE682NA&NE&...	1	31	0	9	1
27 http://www.google.co.jp/search?q=WC+KEE682NA&NE&...	1	2	0	99	1

図 6: 関連検索

File	Rate	TotalTime	ClipTimes	SelectTimes	AccessTime
1 http://dailynews.yahoo.co.jp/	290	33	0	78	19
2 http://sports.yahoo.co.jp/hl?c=sports&d=20050111&a=2...	167	3	0	36	2
3 http://dailynews.yahoo.co.jp/	156	3	0	42	2
4 http://headlines.yahoo.co.jp/hl?c=20050112-00000113-...	156	6	0	29	1
5 http://headlines.yahoo.co.jp/hl?c=20050111-00000111-...	148	6	1	19	3
6 http://dailynews.yahoo.co.jp/	145	1	0	22	2
7 http://dailynews.yahoo.co.jp/	144	2	0	20	2
8 http://dailynews.yahoo.co.jp/	143	2	0	16	5
9 http://headlines.yahoo.co.jp/hl?c=20050111-0000012-...	141	2	0	18	1
10 http://headlines.yahoo.co.jp/hl?c=20050112-0000019-...	140	1	0	18	1
11 http://dailynews.yahoo.co.jp/	137	1	0	11	5
12 http://dailynews.yahoo.co.jp/	135	1	0	13	1
13 http://page8.auctions.yahoo.co.jp/auction/129793439	131	2	1	7	1
14 http://dailynews.yahoo.co.jp/	129	3	0	2	4
15 http://headlines.yahoo.co.jp/hl?c=20050111-00000001-...	126	4	0	0	2
16 http://page8.auctions.yahoo.co.jp/auction/h22124139	126	4	0	1	1
17 http://auctions.yahoo.co.jp/	125	1	0	0	1
18 http://sports.yahoo.co.jp/hl?c=sports&d=20050107&a=2...	125	3	0	0	2
19 http://dailynews.yahoo.co.jp/	125	0	0	1	3
20 http://dailynews.yahoo.co.jp/	125	0	0	3	2
21 http://headlines.yahoo.co.jp/hl?c=20050112-0000008-...	125	1	0	3	1
22 http://headlines.yahoo.co.jp/hl?c=20050107-00000015-...	124	2	0	0	1
23 http://search.auctions.yahoo.co.jp/search/acc?ip=3...	124	2	0	0	2

図 7: 着目度順検索

## 5.1 有用性検証

### 5.1.1 被験者情報

被験者情報を表 2 に示す。被験者が通常利用している PC に DMemFinder をインストールし、数日間利用させた。この際、DMemFinder は、ユーザが Firefox で閲覧した Web 履歴および OpenOffice で作成した文書履歴を蓄積する。

表 2: 対象ユーザ情報

被験者数	15 人
年齢平均	22.0 歳
男女比	2:1
平均 PC 歴	5.1 年
一日平均 PC 作業時間	6.6 時間

### 5.1.2 検証 1. 関連検索は有効に働くか

DMemFinder はユーザの動作を基にデータ間関連度を算出する。関連検索を行う例として、ユーザがあ

るレポートを作成し、後にそのレポートを書いていた時に、参照していた Web ページを検索するといった状況を想定する。検証 1 では DMemFinder を用いて上記シナリオが適切に行えるかを検証する。具体的に行った実験を以下に示す。

1. 被験者に OpenOffice で「世界の携帯電話」という題目で、制限時間 15 分でレポートを作成してもらった。この題目に設定した理由は、一般の人々は携帯電話に関して、ある程度の興味を持っていると想定したからと「世界」と枠を限定したときに専門的知識をもっている人は少ないと想定したからである。
2. レポートの素材は Web から持ってくることにし、役に立った参考文献の URL 5 つをレポートにコピーしてもらった。
3. レポート作成後、レポート名をキーにして関連検索を実行する。被験者がレポート内に記載した URL が関連検索結果の何位に表示されるかを調べた。実験結果を表 3 に示す。また、今回各被験者が事前に蓄積していた履歴の平均数は 312.5 であった。



表 3: 参考文献 URL 表示順位 (位)

平均値	最小値	最大値	中央値
12.3	6.3	43.5	10.4

平均母集団 312.5

被験者がレポートに記載した参考文献の URL は関連検索の順位で平均 12.3 位であった。ある程度目的のページが上位に表示されることを確認した。今回、関連検索では関連度が高く着目度も高いページを上位に表示させるようにしたため、上位数位以内が Google や Yahoo! といった検索エンジンが占めていた。それを差し引けば、それなりに有効に働いたと言える。関連度と着目度の適切な組み合わせ手法は、今後調整していく予定である。

### 5.1.3 検証 2. 着目度算出に利用したユーザ動作は妥当であったか

今回、着目度算出を行う上でのユーザ動作として、データアクセス回数、データアクセス時間、クリップボード利用回数、選択文字列反転回数を用いたが、検証 2 ではそれらが妥当であったかを検証する。DMemFinder では初期段階でユーザの明示的なレーティングと各ユーザ動作との相関を求めるが、被験者全体でどのような傾向がでるか調べた。具体的に行った実験を以下に示す。

1. 被験者に一定期間ニュースサイトを巡回してもらい、各ページをレーティングしてもらう。ニュースサイトにした理由は、1 ページあたりの情報量に偏りが少ないと考えたからである。
2. 被験者が明示的につけたレートと各ユーザ動作の相関係数を算出する。  
実験結果を表 5 に示す。

表 4: 各ユーザ動作と着目度の相関係数 (-1 ~ 1)

項目	平均値	最小値	最大値
データアクセス回数	0.36	0.05	0.87
データアクセス時間	0.25	0.03	0.49
クリップボード利用回数	-	-	-
選択文字列反転回数	0.04	-0.23	0.41

被験者 15 人で対象となった Web ページは 356 であった、有意水準を 1 % としたときの限界値は 0.14 である。データアクセス回数とデータアクセス時間に関しては、平均で限界値以上、正の相関が出てたため、大半のユーザには着目度として利用可能と判断できる。またそれぞれ最小値と最大値に幅があり、個々人の癖を学習しフィルタを作成するのは意義があると言える。クリップボード利用回数に関しては、今回参照した Web サイトがニュースサイトであったため、ほと

んど相関は見られず検証は行えなかった。検証 1 のようにレポートを作成している際の相関を計測すると、ある程度の相関が見られたので、今後いろいろな状況で検討していく必要がある。

### 5.1.4 検証 3. 着目度を用いたフィルタは有効に働いたか

DMemFinder ではユーザの明示的データレーティングを基に着目度用のフィルタを作成した。検証 3 では、フィルタ利用の有無で検索の順位がどのように変動するか調べ、フィルタが有効に働くかどうかを検証する。具体的に行った実験を以下に示す。

1. 検証 2 で算出した相関係数を基にフィルタを作成する。
2. フィルタを用いた上で検証 1 で作成したレポートをキーに関連検索を行い、
3. 検証 1 のフィルタ未使用時の検索結果と比較する。  
実験結果を表 5 に示す。

表 5: フィルタ使用時の参考文献 URL 表示順位 (位)

	平均値	最小値	最大値	中央値
通常時	12.3	6.3	43.5	10.4
フィルタ使用時	10.8	6.2	33.5	7.4

平均母集団 312.5

参考文献の順位平均と最大値が下がり、着目度を用いたフィルタは今回の実験に関してはある程度、検索結果の精度を上げる効果があったと言える。「世界の携帯電話」というテーマではニュースサイトを調べることが多かったため、今回は効果があったと推測され、状況に応じてフィルタを使い分ける手法を考えていく必要がある。

## 5.2 パフォーマンス評価

DMemFinder は検索の度にユーザの全閲覧履歴を参照し直しているため、閲覧履歴が増えるにつれて検索にかかる時間が増加する。そこで本節では、保存したユーザの閲覧履歴数と検索にかかった時間の関係を調べ、パフォーマンスを評価する。実験環境を表 6 に、測定結果を図 8 に示す。

表 6: 実験環境

項目	環境
CPU	Pentium 4 2.8EGHz
メモリ	DDR512MB
HD 回転速度	7600rpm (SATA)
OS	WindowsXP SP2

測定の結果ほぼ履歴数と検索にかかった時間は正比例の関係があると言える。次にプログラムを故意的に

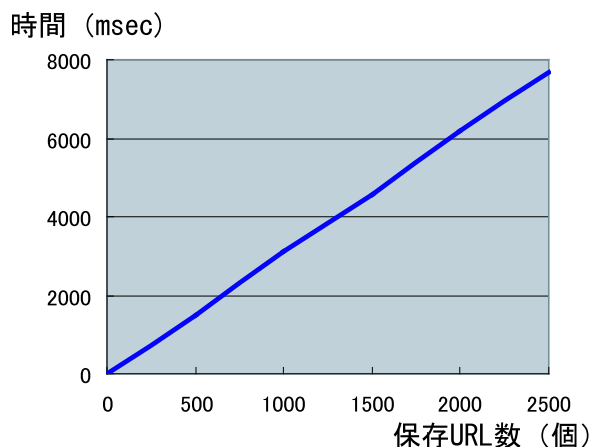


図 8: 検索時間測定結果

遅らせ、検索にどの程度時間がかかるとストレスになるかを被験者にアンケートを行った。アンケートの結果 2.8sec 以上検索に時間がかかると半分以上の人がストレスを感じると答えた。検索時間を 2.8sec 以内にするには、測定結果を基にすると検索対象 URL 数を 1000 個以内にしなければならない。今回の調査では、被験者が一日に参照する Web ページ数平均は 47.6 ページであった。結果、平均して約 20 日分のデータをひとまとめで検索するのがストレスを与えない境界であると言える。今回はプロトタイプ実装であり、パフォーマンスに重点を置かなかったが、今後保存方法、検索方法を最適化していく必要がある。

## 6 今後の展望

本節では、本研究のこれからの展望について述べる。

### 6.1 利用するユーザ動作の追加

今回はユーザのデータアクセス時間、アクセス回数、文字列選択回数、クリップボード使用回数でデータ間関連度とデータ着目度を算出した。今後は、さらに多くの要素を検証していく予定である。また、近年、ユビキタスコンピューティング環境 [11] が実現されつつあり、その環境に対応したアプリケーションの開発も進んでいる。DMemFinder はそれらのアプリケーションに組み込むことが可能である。具体的には、センサを付け部屋の入退出管理アプリケーションに DMemFinder を組み込み、「3 日前、友人 A と一緒に見ていた Web サイトを検索」といったことを可能にできる。

### 6.2 データの動的分類

履歴データ検索を効率化する手法に、データのカテゴリ分けがある。今後は本機構を用いてユーザ動作を基にした自動分類を実現する予定である。例えば、ユーザが頻繁に閲覧し、一閲覧あたりの閲覧時間が短い Web ページを「確認用ページ」と定義すれば、「確認用ページ」に分類されたページは常に監視し、変更

があったらユーザに知らせるといったアプリケーションが作成できる。現在、分類作業を自動化する様々な先行研究 [12] は存在するため、それらとの差分を明確にし研究を進めていく予定である。

## 6.3 メタ情報共有化

DMemFinder を使い、他人との情報の共有化を計り、データ検索をグループ内で最適化するアプリケーションが考えられる。共有化を行うことで友人が高いレートで付けたデータ一覧を検索できる。しかし、他人と共有する場合は、ユーザプライバシーを保護するため、適切なアクセス制御が必要になる。先行研究として、友人同士でブックマークを共有し合う機構がある [13]。

## 7 まとめ

本研究の目的は、普通のユーザに手動操作を要求しない手法で、履歴データ検索を効率化することであり、ユーザ動作を基にデータ間関連度とデータ着目度を算出し、ユーザのデータ検索を支援する手法を提案した。本研究では、ユーザ動作を基に、データ間関連度とデータ着目度算出を動的に行うミドルウェア DMemFinder を構築し、その有用性を確認した。

本研究は総務省委託研究「ユビキタスネットワーク制御・管理技術の研究開発 (ubila プロジェクト)」の一部として行われた。

## 参考文献

- [1] Marvin Minsky. *The society of mind*. Simon & Schuster, Inc., 1986.
- [2] Jose M. Martinez. Mpeg-7 overview. <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>, 3 2003.
- [3] Ujiko. <http://www.ujiko.com/>.
- [4] Yahoo! Inc. My yahoo! search. <http://mysearch.yahoo.com/>.
- [5] MURAKAMI Harumi and HIRATA Takashi. A system for generating user's chronological interest space from web browsing history. *International Journal of Knowledge-Based Intelligent Engineering Systems* Vol.8 No.3, 2004.
- [6] Takehiko Ohno. Impact: Eye mark reusing technique to support information browsing task. *Proceedings of The 8th Workshop on Interactive Systems and Software (WISS'2000)*, 2000.
- [7] The Mozilla Organization. Mozilla firefox. <http://www.mozilla.org/products/firefox/>.
- [8] Inc Sun Microsystems. Openoffice.org. <http://www.openoffice.org/>.
- [9] eclipse.org. Eclipse. <http://www.eclipse.org/>.
- [10] eclipse.org. Swt: The standard widget toolkit. <http://www.eclipse.org/>.

- [11] Mark Weiser. The computer for the 21st century. *Scientific American* 256(3), September 1991.
- [12] Yoelle S. Maarek and Israel Z. Ben Shaul. Automatically organizing bookmarks per contents. In *Proceedings of the fifth international World Wide Web conference on Computer networks and ISDN systems*, pp. 1321–1333. Elsevier Science Publishers B. V., 1996.
- [13] Alessandra Alaniz Macedo, Khai N. Truong, Jos#233; Antonio Camacho-Guerrero, and Maria da Gra#199;a Pimentel. Automatically sharing web experiences through a hyperdocument recommender system. In *HYPertext '03: Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, pp. 48–56. ACM Press, 2003.