論文番号 A1-2

利用履歴に基づく PageRank アルゴリズムの改良

向 亨　成 凱　上林 彌彦

現在の WWW 検索エンジンにおいて PageRank は最も有名なページスコアリングアルゴリズムの一つである。この手法は Web ページが持つリンク構造に基づいてページのスコアリングが行われており、その基本的な考え方は重要なページは重要なページにリンクされているという概念から成り立っている。しかし、ページの作者は必ずしも利用者のアクセス傾向を反映しながらリンクを作っているわけではないため、PageRank の結果は常に利用者が望む結果が出るわけではない。本稿では、プロキシサーバから得られる利用者のアクセスログからリンクを用いないアクセスの履歴(page jump history)及びリンクのアクセス履歴(link navigation history)をページスコアリングに利用する。特に page jump history はブックマークに代表される重要な情報を提供しており、リンクを辿るアクセスよりも重要となると考えられる。これらのデータを用いて、本稿ではより有効な検索結果を導き出すことが可能となる PageRank の拡張を提案する。

A PageRank Algorithm Based on History of Page Jump and Link Navigation

Tohru MUKAI, Kai CHENG, Yahiko KAMBAYASHI.

1. はじめに

近年における WWW の急速な拡大により、それらのデータの中から利用者が必要とする情報を見つけ出す事はますます困難になってきている。そのため、Web 空間における情報検索を実現する手段として検索エンジンが開発され、実際に Web 上で幅広く使用されている。検索エンジンは主にサイトの持つジャンルを手作業で分類、登録するディレクトリ型と、自動的にページを収集し、それらのページに対して索引付けをするロボット型に分類される。本稿ではロボット型の検索エンジンに関して研究を行っている。

ロボット型の検索エンジンは主に以下の手順で検索が実行される。(1) クローラーが Web 空間からホームページを収集する。(2) 検索に用いるインデックスを作成する。インデックスとはホームページの内容（キーワード、URL 等の属性）をファイルにしたものである。(3) それぞれのページがどれほどクエリーに近いかスコアリングアルゴリズムを用いて計算する。(4) クエリーを受け取り、インデックスを参照して結果を返す。この中で最も検索結果の質を左右する意味で重要な役割を果たすのが、スコアリングアルゴリズムである。

スコアリングアルゴリズムは旧来から情報検索の分野で用いられており、その代表的な手法はキーワードの頻度をベクトル化し、クエリーのベクトルとのコサイン値で近さを導き出す方法で、その発展として TF-IDF 法[1]等が提案されて来た。その後 WWW の発展に伴い、それらの持つリンク構造や各種メタデータの利用が着目されるようになってきた。特にリンク構造はこれまでに非常に多くのアルゴリズムにおいて利用されている。その中でも Google[1] で使用されている PageRank [2]は非常に有名なアルゴリズムである。このアルゴリズムは「重要なページにリンクされているページは重要なページである。」という考えに基づき、

京都大学情報学研究科社会情報学専攻
Department of Social Informatics, Graduate School of Information, Kyoto University

[1] http://www.google.com/

リンク元のページの重要度とその数を用いて計算を行っている。しかし、PageRank には幾つか問題点が存在する。その最も重要な問題の１つに、利用者の利用状況とは関係なく全てのリンクを等価に扱っているという問題がある。

また、本稿では現在の多くのスコアリングアルゴリズムが考慮に入れていないブックマークなどに代表されるリンクを用いないアクセスも考える。これらのアクセスは、利用者の意図をより強く反映していると考えられ、これを用いることによってより高い検索結果を得られることが期待される。

以上に基づき、本研究ではアクセスログから得られるリンクアクセス状況及びリンクを用いないアクセス情報を用いて PageRank の改良を提案する。

本稿は以下の内容から構成される。まず２章で実際に本研究で考えているデータモデル及び、アクセスログから得ることができる履歴情報について述べる。次に３章で PageRank のアルゴリズムについて詳細に述べ、４章でこのアルゴリズムに２章で得た履歴情報を適用し、５章でプロトタイプを実装してそのアルゴリズムの評価を行う。６章では実際に検索エンジンを開発する上での履歴情報の更なる利用手段の考察や、利用することによって得られる利点について考え、７章では関連研究の紹介及び、まとめを行っている。

2. ページスコアリングのための履歴情報

本研究はプロキシベースの Web アクセスに基づくシステムを想定している。プロキシサーバはクライアントと HTTP サーバの中間に存在して動作する特別なサーバで、インターネットを利用する際にファイアウォールを通してしか利用できない利用者のアクセスを提供するサーバである。プロキシサーバはユーザが Web にアクセスしている間の利用状況の記録を保持しており、これらのデータは利用の解析や監視において重要な役割を果たしている。

2.1 データモデル

本研究で提案するアルゴリズムはプロキシサーバで記録されているアクセスログを使用している。その他システムの構築のために以下に示す３つの種類のデータを収集している。

1) access log data：いつ、誰が、どのページを閲覧したかを記録しているテキストファイル
2) web data：クローラーや利用者によって集められた Web ページの実データ
3) link structure：Web ページ間のリンク構造

これらのデータはオブジェクト指向データベースに格納され管理される。データベースのデータモデルは図１に示す。

web object は Web 上の web image（画像）や web audio（音声）、web video（ビデオ）、web textual（文書)などのデータを表し、Web HTML は web textual のサブクラスとして存在し、web link を含んでいる。クライアントが新しいページにアクセスしている間、時間、クライアントの ID(または IP)、URL 等のオブジェクトが web access log に格納される。また、リンクやアクセス情報から我々はリンクナビゲーション等に代表される web actemes を得ることが出来る。これらの情報から次節以降に定義される link navigation history と page jump history が容易に計算可能となる。

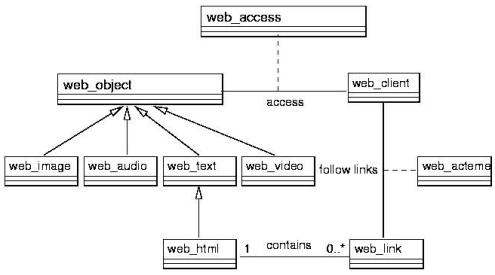

図１：履歴情報のデータモデル

2.1.1 本研究で利用するデータ

本研究では実際にシステムを開発するにあたって、（財）京都高度技術研究所（ASTEM）の協力で、同研究所が運営しているインターネットプロバイダのプロキシサーバで記録されたアクセスログデータを利用している。このデータを用いて次節以降で紹介する履歴情報を取得することになる。

また、Web ページの実データに関しては、このログデータにおいてアクセスが確認されたページを自動的に収集する。そしてそれらのページを解析して、これらのページにおけるリンク構造を抽出することになる。

2.2 Link Navigation History (h_l)

link navigation history はリンクを用いてあるページから別のページへ何度アクセスしたかを示すものである。図５にその例が示されており、この図ではそれぞれの矢印の添え字がそれを表現している。

$h_l(u,s)$を任意のページ u から s への直接のリンクをアクセスした回数とする。O_u を u から出ているリンクの集合とした場合に、４章で行われる行列計算のためにこのデータを以下の式で $rl_{s,u}$ として一般化する。

$$rl_{s,u} = h_l(u,s) / \Sigma h_l(u,t) \ (t \in O_u)$$

2.3 Page Jump History (h_p)

Page jump とはリンクを伴わないアクセスのことであり、page jump history はその頻度を表している。このアクセスには幾つかの種類があると思われるが、その中でも大部分を占めると考えられるのがブックマークによるアクセスである。このアクセスは利用者の趣向を色濃く反映しているものであるため、非常に高い重要性を持つ要素となりうることがある。ブックマークのアクセスは同一人物がリンクを用いず複数回同一のページにアクセスした時であると仮定して、こういったアクセスには重みを加えて値を加算することにしてそれぞれのページにおける page jump history を集計する。

こうした重み付けを行って集計されたページ u に対する page jump history を $h_p(u)$ とした場合に、こちらも４章で行う行列計算のために以下の式によって一般化を行う。

$$rp_u = h_p(u) / \Sigma h_p(w) \ (w \in \text{all web pages}).$$

3. PageRank の概要

L. Page によって提案された PageRank[2] はリンク構造に基づいて Web ページのランキングを計算するアルゴリズムである。本章では次章でこのアルゴリズムの改良を提案するにあたって、このアルゴリズムの内容を述べていき、PageRank が持つ改良すべき問題点を挙げていく。

PageRank の基本的な考え方は、高いランクのページとはリンクされているページのランクの総和が高いページと考えることが出来る。つまり、多くのページからリンクされているページやランクの高いページからリンクされているページは非常に重要なページとなり、スコアも高くなるということである。

3.1 PageRank アルゴリズム

最もシンプルな PageRank を図２に示す。この図を例に基本的な PageRank の計算方法を説明すると、まず全ての Web ページはそれぞれ pagerank の値を持っている。そしてこの値はそのページがリンクしている先のページへ均等に分配されることになる。

図２を例に取ると、図中にある 100 の値を持ったページを考える。このページは２つのページへリンクされているので、このページの持つ 100 の値は２つに分割されてリンク先へ与えられる。つまり、リンク先のページはそれぞれ 50 ずつの値を得ることになる。この考え方を基に以下のようなアルゴリズムで実際に計算を行うことになる。

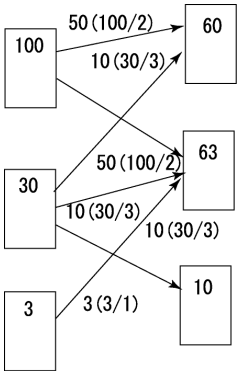

図２：PageRank の基本概念

3.1.1 基本的な計算式

基本的な PageRank の概念は前述した通りだが、この節ではその内容を具体的な計算式として示すことにする。

もし u が Web ページとした場合に、F_u は u にリンクされているページの集合である。また、N_u を u から出ているリンクの数（$N_u = |F_u|$）とし、c を一般化のための定数、そして u からリンクされているページ集合を B_u とする。この時ページ u における pagerank の値 $R_{(u)}$は以下の計算式によって計算される。

$$R_{(u)} = c \Sigma R_{(v)} / N_v (v \in B_u)$$

3.1.2 行列計算による方法

前節で述べた計算式を実際に全てのページに対して適用するために、行列を用いて計算を行う。計算は以下の方法によって計算される。

まず、前の節同様に u を Web ページ、u からリンクされているページの集合を F_u、N_u を u から出ているリンクの数、c を一般化のための定数とする。この時、ある行列 A を考える。この行列は要素 $A_{u,v}$ を次のように定義した行列である。もし u から v へとリンクが貼られている場合には $A_{v,u} = 1/ N_u$、もしリンクが無い場合には $A_{v,u} = 0$ となる。そして、R をそれぞれの Web ページの持つ pagerank の値を要素とするベクトルとした場合に

$$R = cAR.$$

という式を得る。この R は行列 A の固有ベクトルとなることがわかっている。実際には R に対して A を繰り返し適用することによって値を計算することになる。

図３のようにリンクがつながっている Web 空間の場

合を例にとって実際に行列を求めてみる。このようにリンクの関係が成り立っている場合、求める行列式 A は図４のようになる。

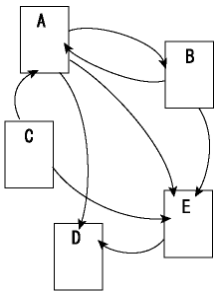

図３：リンク例

$$\begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 1 \\ 1/3 & 1/2 & 1/2 & 0 & 0 \end{pmatrix}$$
図４：行列 A

3.1.3　リンクを用いないアクセスの考慮

しかし、このアルゴリズムには小さな問題が存在する。幾つかのページがリンクによってループしていたり、その数ページの間だけでリンクの関係が成り立っていたりする場合に、これらのページの値が無限大に発散してしまう点である。この問題を解決するために PageRank アルゴリズムは以下のように拡張されている。

$E(u)$を page jump を示すベクトルと考えた場合、３．１．１節の計算式とこのベクトルを用いて pagerank の値 $R'(u)$は以下の計算式で計算が可能である。

$$R'(u) = c\Sigma R'(v) / N_v + cE(u) \quad (v \in B_u)$$

（c は最大の値をとり、 $||R'||_1 = 1$ を満たす）

この計算も同様に行列で計算が可能となる。行列 E を全ての行列要素が等しく（全ての要素が正である）、かつ $||E|| = 0.15$ となる行列とする。勿論、この行列はリンクを用いないアクセスに対応する行列である。この時、これらの行列は次の式を満たすことになる。

$$R' = c(AR' + E) \quad (||R'||_1 = 1)$$

ここで、$||R'||_1 = 1$ であるため、$\mathbf{1}$ を全ての要素が１である行列として、上の式は以下の式に置き換えることが出来る。

$$R' = c(A + E \times \mathbf{1})R'$$

この計算式も３．１．２節の計算と同様に$(A + E \times \mathbf{1})$を繰り返し適用することによってベクトルを導き出すことが可能となる。

3.2　PageRank の問題点

PageRank はリンク構造の解析を用いた非常に優れたアルゴリズムである。しかし、PageRank には幾つかの問題点が存在している。最も大きな問題として存在するのが各種アクセスが全て等価であるという仮定の基にこのアルゴリズムが成り立っているという点である。

この点は実際の計算においては、行列 A を作成する際に、ある列における要素の値がすべて同じ値になっているという問題となる。しかし、実際の利用者がアクセスする際には、全てのリンクに対して均等にアクセスするということは考え辛く、必ずリンクによってアクセスに偏りがあるはずと考えられる。これは行列 E に関しても同様で、全ての要素が等しい行列、つまり全てのページに利用者は同じ確率でアクセスすると考えられているが、実際にはアクセス状況には大きな偏りが存在する。

また、PageRank は定常確率を考慮に入れていない。定常確率とは他のページにアクセスすることなく利用者がとどまる確率を表すもので、全てのページはそれぞれこの確率を一定の割合で保持していると考えられている。

これらの問題点を解決するために、２章で述べた link navigation history 及び page jump history を用いた PageRank の改良、つまり履歴情報による PageRank の拡張を次章において提案する。

4.　履歴情報を用いた PageRank の拡張

本章では前章で述べた PageRank のアルゴリズムに対して、２章で述べたデータを適用して拡張を行う。この拡張は以下に示す３つの改良から成り立っている。

1)　link navigation history の適用
2)　定常確率の導入
3)　page jump history の適用

これらについて順に詳細に述べていく。

4.1　Link Navigation History の適用

link navigation history は全てのリンクに存在している。図５は link navigation history を付加したリンク構造の図である。このデータを３．１．１節で述べ

た計算式及び３．１．２節で述べた行列計算に対して適用することで改良を行う。

具体的な手法を次に述べる。従来までの手法では、pagerankの値はリンクに対して均等に分配されていた。しかし、本稿ではその分配を link navigation history の割合に応じて分配しようと考える。図５のページ A を例にとって説明すると、これまでの手法であると A の pagerank の値は A がリンクしているページである B、D、E に均等に 1/3 ずつ与えられるが、本稿の手法では B に 10/17、D に 2/17、E に 5/17 の割合で分配される。つまり、計算式は以下のようになるのである。

$$R_{(u)} = c\Sigma R_{(v)} \cdot rl_{s,u} \quad (s : u \rightarrow v, \quad v \in B_u)$$

次にこの改良を行列式の計算にも適用する。行列 A はもし u から v へとリンクが貼られている場合には $A_{v,u} = rl_{s,u}$ 、もしリンクが無い場合には $A_{v,u} = 0$ となる。そして同様に各ページの pagerank の値が要素となるベクトルを R とした場合、

$$R = cAR.$$

の式が成り立つ。

図５の場合、行列 A は図４の行列から図６の行列へと変化することになる。

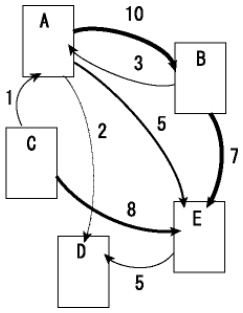

図５：link navigation history の適用

$$\begin{pmatrix} 0 & 3/10 & 1/9 & 0 & 0 \\ 10/17 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 2/17 & 0 & 0 & 0 & 1 \\ 5/17 & 7/10 & 8/9 & 0 & 0 \end{pmatrix}$$

図６：link navigation history を適用した行列 A

4.2　定常確率の導入

次に、定常確率の導入を行う。この改良は３．１．３節で述べた計算式に対して行う。

まず、定数 con を定常確率の値として定める。この時、ページ u における pagerank の値 $R'_{(u)}$は以下の計算式で計算する。

$$R'_{(u)} = c\Sigma R'_{(v)} \cdot rl_{s,u} + cE_{(u)} + con$$

$$(s : u \rightarrow v, v \in B_u, t \in O_v)$$

（c は最大の値をとり、 $||R'||_1 = 1$ を満たす）

これを行列式に当てはめると次のようになる。定数 con を定常確率とした場合に、C は要素が $u = v$ の場合 $C_{u,v} = con$ で、それ以外の場合は $C_{u,v} = 0$ となる正方行列とする。この時、先述の計算式は以下の行列式で計算できる。

$$R' = c(AR' + E) + C$$

この式は以下の式に置き換えが可能である。

$$R' = c\{AR' + E + (con / c) \cdot C\}$$

この時、Cはすべての要素が正の数の行列あり、等式 $||R'||_1 = 1$ が成り立つので $E' = E + (con / c) \cdot C$ とすると、上記の式はこのように書き換えることが可能となりこれまでと同様の手段で計算することができる。

$$R' = c(A + E' \times \mathbf{1})R'$$

4.3　Page Jump History の適用

すべてのページはこれまで得た利用履歴のデータから page jump history を保持している。このページの様子を図７に示す。これらのデータから、改良のために必要な全てのページの持つ page jump history の総和に対するそれぞれのページの page jump history の割合が計算できる。これを用いてこれまでリンクを用いないアクセスについての扱いは全てのページに等確率にアクセスすると仮定していたが、page jump history の割合にしたがってアクセスすると解釈する。

実際に図７を例にとってみると、これまでの手法ではページ C へのリンクを用いないアクセスは 1/5 の割合で行われるとして計算していたが、本研究では 7/100 の割合で行われると判断する。

これを行列式に適用すると、特に式に変更はないが、行列 E がそれぞれの要素が $E_{u,v} = rp_u$ の正方行列へと置き換わることで計算できることになる。

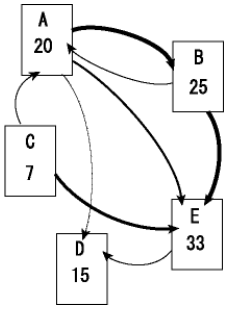

図７：page jump history の適用

5. 実装及び実験結果

これまで提案してきたアルゴリズムの効果を調べるために実際にプロトタイプを開発し、それを用いて実証実験を実施した。この章ではその概要と結果について述べる。

5.1 プロトタイプの概要

検索語の入力画面を図８に示す。この画面で利用者は検索語の入力と使用するアルゴリズムの選択を行い、検索ボタンを押すと選択されたアルゴリズムのスコアが上位１０ページの URL のリストが表示される（図９）。利用者はこのページからその検索されたページへアクセスすることが可能となる。

このプロトタイプの実装に際して、2.1.1 節でも述べたように実際に商用のプロバイダ（Kyoyo-Inet）のプロキシサーバで昨年７月に記録されたアクセスログデータをログ解析用のデータとして利用している。ただし、このデータはプライバシー保護のためユーザに関する情報はマスクされている。ページ数は約１２万ページ、html のアクセスログは約１３０万アクセスのデータを用いている。

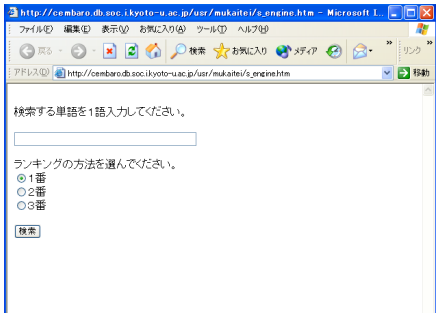

図８：クエリー入力画面

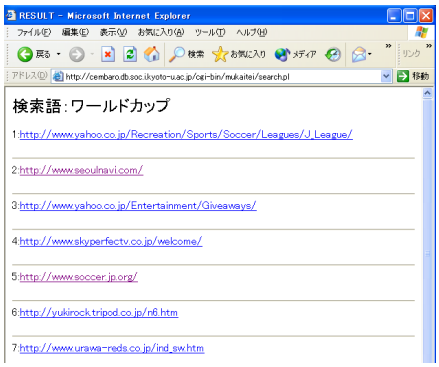

図９：結果出力画面

5.2 実証実験概要

前節で述べてきたプロトタイプを用いて本研究で提案したアルゴリズムの効果の検証を行う。実験は以下の手順で行う。

被験者は実験側が２５の単語から任意に５つの単語を選択する。

それぞれの単語に対してプロトタイプシステムを用いて、３つのアルゴリズムについて検索を行う

出力されたスコア上位１０位までのホームページのリストを閲覧し、その内容を０～５の６段階で評価する。

この実験を７人の被験者に対して行い、得られたデータに関して以下の３つの基準で評価を行った。

適合率：１以上の評価を得たページの割合

平均評価：評価された１０ページの評価値の平均

重み付評価：評価値に対して（１１－ページの順位）倍した値の合計

5.3 実験結果

それぞれの評価基準についての結果を以下の表に示す。表１は適合率、表２は平均評価、表３は重み付評価の結果を示しているが、全ての基準において若干ではあるが PageRank を下回る結果となった。

ユーザ番号	PageRank	改良後
1	0.36	0.32
2	0.54	0.48
3	0.72	0.66
4	0.56	0.56
5	0.58	0.48
6	0.48	0.50
7	0.88	0.78
平均	0.59	0.54

表１：適合率

ユーザ番号	PageRank	改良後
1	1.12	0.82
2	1.86	1.60
3	2.04	1.70
4	1.38	1.06
5	1.70	1.46
6	1.62	1.54
7	1.94	1.76
平均	1.67	1.43

表２：平均評価

ユーザ番号	PageRank	改良後
1	304	238
2	552	473
3	535	514
4	383	340
5	452	346
6	458	447
7	528	468
平均	459	404

表 3：重み付評価

5.4 考察

前節で述べた結果は期待されたものとは異なる結果となった。この節ではその原因を考えることで、本アルゴリズムが有効に機能する状況を考察することにする。原因考察のためにここで改良後のアルゴリズムのみに出てきたページの性質について調べることにする。図 10 は検索語をプロ野球選手である「イチロー」とした時に改良後のアルゴリズムにのみ出てきたページである。このページはログデータを取得した時期におけるイチローに関するニュースのページである。このようなニュースページはこの場合のようにページが残っている場合には高い評価を得ることができるが、その殆どは内容が変わっているか削除されてしまっており、それが評価を下げる原因となっている。

また、この他に見られたページにイベント関連の情報が書かれたページがある。図 11 は検索語を歌手である「ゆず」とした時に改良後のアルゴリズムにのみ出てきたページである。このページには全く「ゆず」についての記述は無い。しかし、データを取った 7 月時のページ（図 12）では「ゆず」がその時に行ったイベントに関するトピックが記載されている。

こういったページに共通するのはいずれも時間に対する依存度が強いことである。今回は半年ほど前に取得したデータであったため、時間依存の強い改良後のアルゴリズムに悪い結果が出たのではないかということが推測される。しかし、このことから常に新鮮なログデータと定期的なデータ取得を実行できれば本研究で提案したアルゴリズムは有効に機能するということが期待されると考えられる。

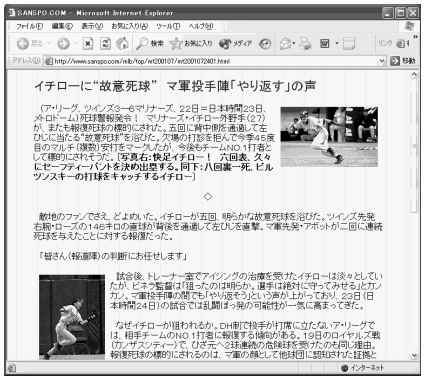

図 10：ニュースページの例

図 11：時間限定イベントに関するページ

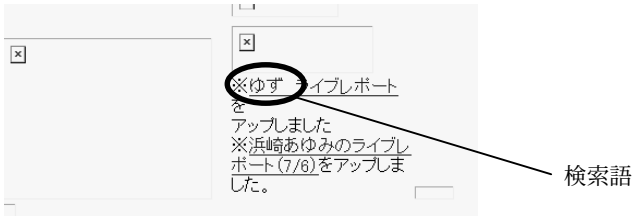

図 12：7 月時のページ

6. 履歴情報に基づく情報検索

この章ではこれまで述べてきた履歴情報を用いた検索エンジンの開発を考える。もちろんこのシステムのスコアリングアルゴリズムはこれまで述べてきたアルゴリズムを採用する。それに加えて、その他アクセスログデータから取得できるデータを用いることで別の角度から利用者の要求に合った検索結果を提供できると考える。

履歴情報を利用した情報検索手法にはいくつかの利点があると考えられる。1 つは、伝統的な情報検索の手法では実現不可能であった新しい種類の検索が可能となることである。たとえば、以下に示すような要求

を満たすことも可能となるということである。

1) あるクエリーに関して「ここ１週間でもっとも重要な」ページを検索する
2) あるクエリーにおいて「ある曜日に最も重要な」ページを検索する

これらは履歴データを常に得ることで動的な情報として扱うことを可能とし、特定の時間帯や期間における重要度が計算可能となる。

次に重要なこととしては興味の共有ということが考えられる。履歴データは個人のものではなく、不特定多数の利用に関するデータである。つまり、集団の持つ利用傾向を検索結果という情報を通して知ることができるようになるわけである。この利点は検索のみならずページの推薦といった分野でも非常に効果的であると考えられる。

その上、利用履歴を利用することで我々はまったく閲覧されていないページなどといった、利用者も製作者も重要と感じることのないページを、結果を返す前にフィルタリングしてしまえるという利点も存在する。こういった要因からこれまでのアルゴリズムの改良も含め利用履歴を用いた情報検索はより高い質を持った検索結果を提供することが可能であると考えられる。

7. 関連研究及びまとめ

7.1 関連研究

検索にリンク構造を利用した例は他にも幾つか存在している。PageRank と並んで有名なアルゴリズムに Kleinberg の HITS[3]がある。これは PageRank 同様、重要なページにリンクされているページは重要であるという考え方に基づいたアルゴリズムで、ページを Hub、Authority という２つの値で評価するアルゴリズムである。

そして、これらのアルゴリズムに関しては HITS の方が数多く改良案が提案されている。そして本研究のようにリンクの状態に応じて重み付けを行うアルゴリズムが存在している。これらの研究は、リンク周辺の文章の内容によって Hub 及び Authority の値を変化させるもの[4]や、リンクの部分集合の構造によって変化させていくもの[5]などが存在している。

また、ブックマークに関する研究として NEC America の PowerBookmarks[6]がある。これはデータベースを用いることによって複数人のブックマークを管理し、それによってユーザのアクセス傾向などを判断し最適なサービスを提供するシステムである。

7.2 おわりに

本稿ではリンク構造を利用したスコアリングアルゴリズムである PageRank に対して、利用履歴に基づいた拡張を提案した。本研究ではこの拡張のためにプロキシサーバから得られるアクセスログデータから page jump history と link navigation history という２つのデータを抽出し利用している。そして実際にプロトタイプを実装し評価を行った結果、期待した結果を得ることは出来なかった。しかし、原因を考察した結果、時間に対する依存性が強いことがわかった。このことから、より新しいログデータと定期的なデータ収集によってより良い結果が得られることが期待できる。

今後の目標としては、アルゴリズムの改良や更なる利用履歴のデータ活用によって、これまで以上に利用者にとって有用な検索エンジンの開発を目指すことが挙げられる。

参 考 文 献

1) G. Salton and C. S. Yang: On the Specification of Term Values in Automatic Indexing, *Journal of Documentation*, 29(4): pp 351-372. December 1973
2) L. Page and S. Brin and R. Motwani and T. Winograd: The PageRank Citation Ranking: Bringing Order to the Web, *1998*
3) J. Kleinberg: Authoritative Sources in a Hyperlinked Environment, *Research Report RJ 10076(91892)*, IBM, 1997
4) K. Bharat and M. Henzinger: Improved Algorithms for Topic Distillation in a Hyperlinked Environment, In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp104-111, 1998
5) S. Chakrabarti, M. Joshi, V. Tawde: Enhanced Topic Distillation using Text, Markup Tags, and Hyoerlinks, *Proceedings of the 24th annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp 208-216,2001
6) W. Li, Q. Vu, D. Agrawal, Y. Hata, H. Takano: Powerbookmarks: A system for personalizable web information organization, sharing, and management, *Proceedings of the 8th WWW Conference*, 1999
7) K. Cheng and Y. Kambayashi: Enhanced Proxy Caching with Content Management**,** Knowledge and Information Systems: *An International Journal, Springer-Verlag*, UK 2001 (to appear)