マルチエージェントシステムによる更新検知を用いたWebサーチエンジン

2001MT026 廣瀬 孝志
指導教員　河野 浩之

1　はじめに

2004 年現在の日本のインターネット人口は，およそ 6284 万 4000 人である [1]．その多くの人々が利用しているだろうサーチエンジンには，さまざまな問題点が存在する．本研究ではそれら問題点の解消を目的としたサーチエンジンの実装を試みる．その手段の一つとして，各サイトの更新状況を把握し，より「更新が多く」，「アクセスが多い」ページが検索結果の上位に表示されるようなシステムを提案する．そしてその手段として，マルチエージェント技術を利用する．

2　現在の更新検知技術とサーチエンジンの技術背景

サーチエンジンの課題として，検索結果の表示順序が Google では他の有用なページにリンクされているページを上位に表示，Yahoo では階層構造をあらかじめ登録しておくなど，必ずしも有用なページが上位に表示されないような順序付けの不正確さや，よく使用される単語による検索を行った際に，10000 件以上の過剰な検索結果を示してしまうなどが挙げられる [2]．本研究ではその他様々な問題点の中から，先に述べた「順序付けの不正確さ」という点に着目し，研究を進める．このとき，順序付けをどのように行うかの手段として，更新検知の技術を提案する [3]．

現在の更新検知技術として，RSS という手段が存在する [4]．これは，興味あるサイトに RSS が提供されている場合に，そのサイトの URL を登録しておくことによって，登録されたサイトを要約し更新日時などを記録する技術である．その特性を利用して，RSS リーダーというソフトで更新を管理することができる．

3　更新検知型サーチエンジンの実装

本研究での実装に求めるべき条件を以下の三点に大まかに分類した．

1. 有用なページを優先的に取得できること
2. 更新されたページを更新後早期の段階で取得できること
3. サーチエンジンという形で提供できること

本研究では他の研究生の協力のもと，アクセスログを蓄積する．プロキシによって access.log というファイルで保存され，それを解析することで，上記の要求を順に満たす．

有用なページを優先的に取得するためには，ページごとのアクセス数をカウントすることで，どのページがより多くの人に見られているかを調査する．
access.log の中の複数の情報から URL 情報のみを抽出し，.html，.htm，/のいずれかで終わる URL だけを取り出す．それぞれの URL に何回アクセスされているか調査し，アクセス回数が多い順に URL をソートする．

次に，更新が行われたかどうかを調査し，それぞれ更新の行われたページに重みを与える処理をする．更新検知には，wget を用いて一定時間間隔でページごとに HTML ファイルのダウンロードを行い，以前取得したものと比較して変更があれば，更新としてみなす．wget はアクセス数を元に実行間隔を決定する．
まず，wget が行わると，そのサイトが以前にも収集されたかどうかを確認する．

初めてのダウンロードの場合には重み 1 を与える．そうでない場合は以前取得した際のファイルと比較し，変更点があれば 1 以上の重みを加える．変更がなければ重みを 0 として加える．

これらから得られたページごとのアクセス数や重みを利用して，サーチエンジンを図 1 のように実装する．

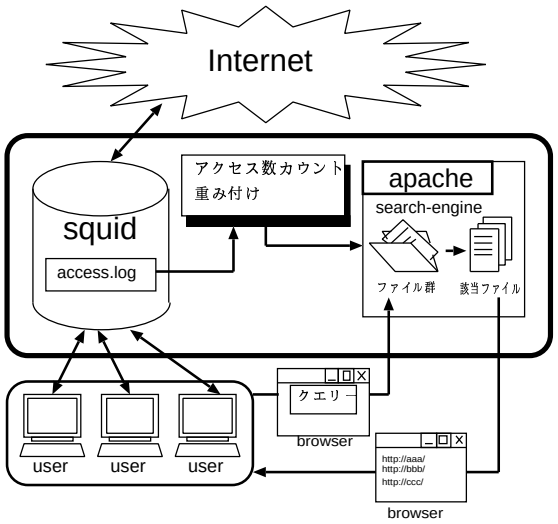

図 1　更新検知型サーチエンジンの実装

まず，ユーザがクエリーを入力するページを HTML 文書で記述し，ブラウザで表示する．ユーザからのクエリーを全文検索のキーにする．サーチエンジンは重みの大きい順に検索結果をソートして返すため，重みの大きい順に並べ換え，その上位のページから全文検索を行う．

キーとなる語が見つかったページ名から順に，そのページの URL を検索結果としてユーザに返す．

これらのシステムをエージェントとして自動的に動作させる．

4 更新検知型サーチエンジンの検証

ページごとに重みを与えるプログラムにおいて，重みを与える際の与え方について最適な手段を考察する．本研究では，重みを与える際に，前回ダウンロード時との比較を行い更新が行われたかどうかを検知する手段として，diff コマンドを使用する．

ページごとの重みがバラバラに別れ，順位付けする際には非常に有用であるが，その反面，値が収束せず順位の変動が難しくなる．

ここで diff カウントによる値を一定間隔ごとに区切り，その間隔ごとに重みを与える方法を考える．このことにより，値がある程度の範囲に収まる．一度の変更による格差はなくなり，順位の変動も容易となる．ただし，細かいところの順位付けができないという短所が挙げられる．

さらに diff カウントによる値を異なる間隔ごとに区切り，その間隔ごとに重みを与える方法を考える．この方法で最適な値の指定を考えれば，前述の二つの方法の長所を取り出し短所を取り去ることが可能である．このとき，次の条件の下で最適な式を提案する．

●更新検知判定に必要な条件

1. ページ自体が入れ替わった場合などに極端に大きな重みを与えない
2. 更新量の小さいものに，大きな重みを与えない
3. ページごとに重みの順位付けができるように与える重みの範囲を決定する

上記の条件により，更新量が小さい場合または大きすぎる場合には更新量によって重みにほとんど差が生じず，適度な量の更新が行われているページに対して重みに差をつけて与える，式 (1) を提案する．

$$y = \begin{cases} \frac{D}{1+E^{-(x-C)}} & (x > c) \\ \frac{D}{1+E^{x-C}} + D & (x \leq c) \end{cases} \quad \cdots(1)$$

C，D，E の値は，検証の結果，条件を最も満たすと思われる値を与える (C(適度な更新量)=20，D(与える重みの限界値)=20，E=1.17，x=diff カウント，y=重み)．

検証では，一日当たりの各ページの更新の量の平均値を求め，それを適度な更新量として C に与えている．D と E は C に依存して決定する．研究時の平均値はおおよそ 20 であったが，この数値は日毎変動する可能性があるため，更なる評価を要求される．この式を元にそれぞれ重みを与えると，通常の diff カウントをそのまま重みとするのに比較して，図 2 のような，条件に見合う重みの与え方が可能となった．図 2 の F〜T は，重みを与えたページを更新量ごとに取り出したものである．

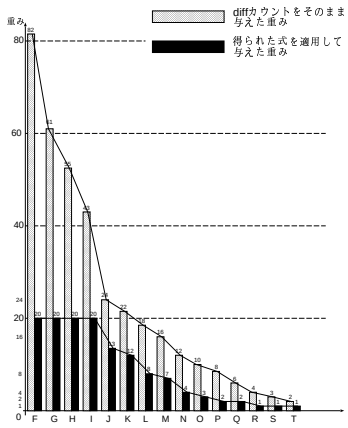

図 2 diff カウントと式 (1) を利用した重み付けとを比較したグラフ

5 おわりに

本研究ではアクセス数の多いページに積極的に更新検知を行い，更新が検知されたページに重みを与える．その重みが大きい順にサーチエンジンの検索結果として表示することで，更新の多いページを優先的に表示できるサーチエンジンを実装し，重みの付け方を考察することでその性能向上を図った．重みの与え方として，更新量を不定な異なる間隔で区切り判定する手法を最適とし，関数を提案した．だが，本研究にはさまざまな問題点が挙げられる．膨大なページを扱う場合の処理時間の遅延や，一度与えた重みを保持し続けるという問題がある．検索段階で AND 検索や日本語検索が出来ないなどの，サーチエンジン本体の性能についても挙げられる．今後の研究では，大規模長期利用の際に生じる問題について考察していくことが求められる．

謝辞

最後になりましたが，本研究を御指導頂いた河野浩之教授，アクセスログの蓄積に協力頂いた河野研究室の学生のみなさんに感謝致します．

参考文献

[1] インターネット白書 2004，財団法人インターネット総会 (2004.7)

[2] 原田昌紀：サーチエンジン徹底活用術， オーム社 (1997.12)

[3] 山田誠二：Web 更新モニタリング，情報処理 44 巻 7 号 5 章 (2003.7)

[4] Dan Libby：RSS0.91 Spec revision3, `http://my.netscape.com/publish/formats/rss-spec-0.91.html(accessed 2004.6)`