アクセスログを用いた自動 Bookmark 生成システムの開発

05H082 永田和也

目次

1 はじめに 1

2 卒業研究の目的 2

3 Bookmark 生成システム 3
3.1 自動 Bookmark 生成システム 3
3.2 ソーシャルブックマーク 3

4 開発したシステム 4
4.1 システムの概要 4
4.2 システムの動作 6
4.3 スクリーンショット 9

5 考察 10
5.1 利点 10
5.2 問題点 10

6 まとめ 11

7 今後の課題 12

8 謝辞 13

付録 A ソースコード 14
A.1 cron の設定 14
A.2 Bookmark システム 14
A.3 アクセスログのリセット 18

1　はじめに

インターネットには膨大な量の情報がある。ユーザが知りたい情報や興味のある分野のキーワードを Google [1][*1]や Yahoo!JAPAN [2][*2]などの検索エンジン[*3]で検索することで、簡単に欲しい情報を手に入れることができる。しかし、検索エンジンでの検索では、入力したキーワードに関する情報しか出てこない。ユーザが興味を持っている対象は検索すべきキーワードを容易に特定できるため、情報を得ることも容易である。しかし、自分が興味を持つかも知れない対象を探すことは困難である。テレビや雑誌などの情報媒体の場合、ユーザが受動的でも様々な分野の情報を提供してくれ、ユーザの興味の幅を広げることを支援してくれる。

しかし、インターネットの検索エンジンではユーザが能動的に情報を得ようとしない限り情報を手に入れることはできない。そこで、ユーザの興味の対象となり得る Web ページを自動的に収集するシステムがあれば、ユーザが受動的でも興味の幅を広げることを支援できるのではないかと考えた。受動的にでも情報を得られる方法としては、キーワードでの検索など手動のものは向いていないと考え、自動で Web ページを登録することができる自動 Bookmark[*4]生成システムに注目した。また、ユーザに手間を掛けさせる必要なくユーザの興味の対象を調べる方法として、Web ページのアクセスログの利用を考えた。

[*1] Google とは、ソフトウェア会社 Google が提供する検索エンジンである。
[*2] Yahoo!JAPAN とは、株式会社 Yahoo! が運営するポータルサイトで、検索エンジンはその主力事業である
[*3] 検索エンジンとは、インターネット上に公開されている情報をキーワードなどを使って検索できる Web サイトのことである。
[*4] Bookmark とは、任意の Web ページの URI を登録する機能のことである。

2　卒業研究の目的

本研究では、ユーザにキーワード入力による検索や、Bookmark の手動登録などの手間を掛けさせずに、ユーザの興味の対象となり得る Web ページをインターネットから提案する方法を考案する。また、実際に proxy サーバ[*5]の squid [3][*6]のアクセスログを利用した、研究室内で動作する自動 Bookmark 生成システムを開発し、その動作結果から考案した選別方法が有効かどうかを検証する。

ユーザが受動的でいても情報の収集ができるようにし、ユーザが負担を感じること無く興味の幅を広げられるよう支援をすることが目的である。また、ユーザにとって有用な情報を収集することで、ユーザの情報収集にかける時間を減らすことも目的とする。

[*5] proxy サーバとは、ローカルネットワークとインターネットの境界などで通信を中継するサーバのことである。

[*6] squid とは、proxy サーバソフトの名称である。proxy を介してアクセスしたユーザ名、URI、日時、プロトコルなどを記録する機能がある。

3 Bookmark 生成システム

以下に、既存の Bookmark 生成システムを示す。

3.1 自動 Bookmark 生成システム

ユーザが特に操作をしなくても自動的に Bookmark を生成するシステム。

- ABM(Auto Book Mark)
  情報安全工学研究室の過去の卒研生が開発した自動 Bookmark システム。proxy サーバの squid のアクセスログを自動的に蓄積し、そのアクセスログを利用して Web 上に Bookmark を作成する。URI の観閲時期による重みづけと観閲回数を利用し点数を計算し、点数の高い URI を Bookmark する。
- EBM(Everybady's Book Mark)
  前述の ABM を改良し、さらにユーザ共有機能を追加させたシステム。アクセスログ解析時の URI の重みづけ方法が見直され、手動で Bookmark を登録する機能も追加されている。また、複数のユーザで一つの Bookmark を作ることができる。

3.2 ソーシャルブックマーク

複数ユーザ間で Bookmark を Web 上で共有できるサービス。

- del.icio.us [4]
  2003 年より提供されているソーシャルブックマーク[*7]の先駆けとなったサービス。自分の Bookmark を Web 上で管理できるほか、Bookmark を他のユーザに公開・ 共有できる他、それぞれの Bookmark にタグを付けることで全体的な分類や管理ができるなど、さまざまな先進的な機能を備えていた。
- Pathtraq [5]
  サイボウズ・ラボ株式会社が提供している Web サイトランキング化サービス。ユーザはブラウザに専用の拡張ソフトをインストールし、サーバにアクセスログを送信する。それぞれのユーザから送信されたアクセスログの統計をとり、ランキング形式で表示する。ソーシャルブックマークに似たシステムで、ユーザ間の情報の共有を主な目的としている。

[*7] ソーシャルブックマークマークとは、インターネット上に Bookmark を公開し、他のユーザと共有できるサービスのことである。

4　開発したシステム

4.1　システムの概要

本システムではアクセスログからユーザにとって重要度の高い Web ページを選出し、その Web ページと関連性の高い Web ページをインターネット上から探し出し、自動的に Bookmark を作成する。この Bookmark は Web 上に作成し、どの端末からでも、どのブラウザからでも参照できるようにする。また、複数ユーザでの使用に対応させ、それぞれのユーザ間での Bookmark の共有も行う。

システムの概略図を図 **1** に示す。

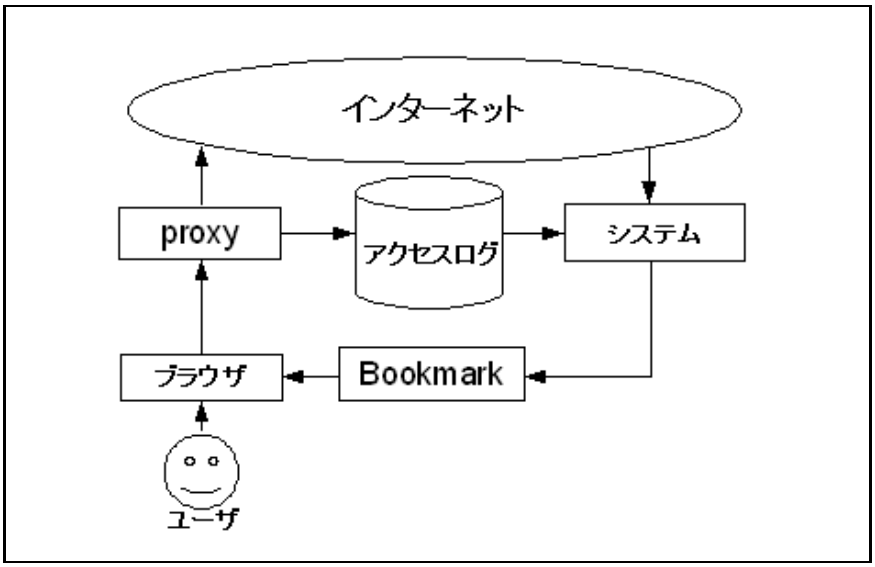

図 1　システムの概略図

4.1.1　開発環境

以下にシステムの開発環境を示す。

- ハードウェア (AT 互換機)
  - CPU:Intel(R) Celeron(R) CPU 2.80GHz
  - memory:512MB
- OS:Vine Linux 3.2 [6]
- 開発言語:シェルスクリプト (bash [7])

4.1.2 アクセスログ

本システムで利用するアクセスログは proxy サーバの squid のログを利用する。研究室内から外部へのアクセスはこの proxy サーバを介して行われる。アクセスログにはアクセスした時間、読み出しが終了するまでに経過した時間、リクエストしてきたクライアントの IP アドレス、結果コード、クライアントへ渡された総バイト数、リクエストメソッド、リクエストされた URI、ユーザ名、階層コード、コンテンツのタイプの順にスペース区切りで記録される。アクセスログのサンプルを図 **2** に示す。このアクセスログから研究室内のユーザのアクセス記録を抽出する。

```
1227660211.642    613 192.168.1.81 TCP_MISS/200 99083 GET
http://pc.watch.impress.co.jp/ nagata DIRECT/202.218.223.135 text/html
```

図 2　アクセスログのサンプル (実際は一行)

4.1.3 重要度の基準

ユーザにとって重要度の高い Web ページを選出する基準として、Web ページの観閲回数を利用する。観閲回数の多い Web ページほどユーザにとって重要度が高いと判断する。

4.1.4 関連性の基準

対象となる Web ページと関連性の高い Web ページを探し出す方法として、Web ページ中に含まれる語句を利用する。Web ページ中に含まれる語句で、使用回数の多い語句ほどその Web ページとの関連性が高いと判断する。

4.1.5 インターネットでの検索

選出した語句をインターネットで検索する方法として、検索エンジンの Yahoo!JAPAN を利用する。検索エンジンの選択肢には他に GoogleAPI[*8]の利用が挙げられたが、使用回数に制限があったため今回は回数制限のない Yahoo!JAPAN を選択した。

[*8] GoogleAPI とは、様々なアプリケーションから Google のデータベースを利用できるインターフェースである。Google が公開しており無料で利用できるが、1 日に 1000 クエリーという制限がある。

4.2　システムの動作

開発したシステムの動作の流れを図 **3** に示し、その詳細の説明を行う。

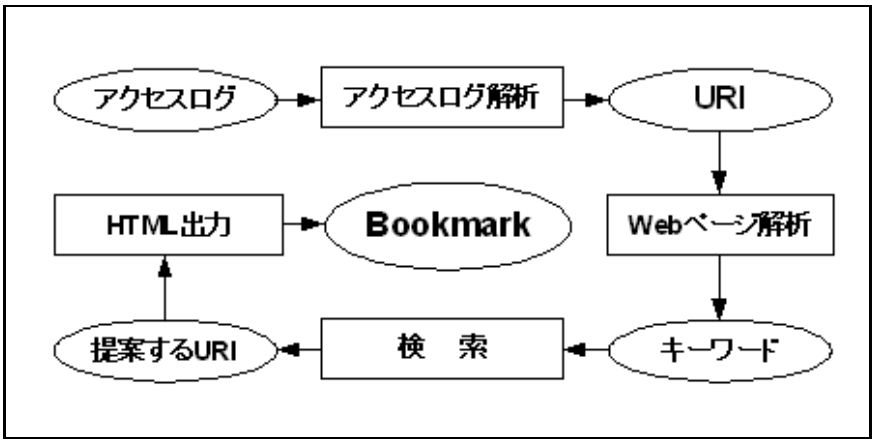

図 3　動作の流れ

4.2.1　アクセスログの保存

まず、proxy サーバのアクセスログをローカルに保存する。アクセスログは 1 日に一度 proxy サーバから取得し、ローカルに保存されたアクセスログに追加保存する。そして、ユーザの興味の移り変わりに対応するために、1 週間に一度ローカルに保存されたアクセスログの中身をリセットする。

4.2.2　アクセスログの解析

ローカルに保存したアクセスログからユーザ名毎に URI を抽出する。抽出した URI のそれぞれの観閲回数を調べ、多く観閲しているものから URI を保存する。アクセスログの解析結果を図 **4** に示す。

```
6 http://kakaku.com/
5 http://kakaku.com/pc/
3 http://www.watch.impress.co.jp/akiba/
3 http://www.k-tanaka.net/unix/cat.html
3 http://pc.watch.impress.co.jp/
3 http://gigazine.net/
2 http://dc.watch.impress.co.jp/
2 http://cgi.itmedia.co.jp/quickpoll/html/pcuser_081015.html
2 http://av.watch.impress.co.jp/
1 http://www.watch.impress.co.jp/game/
1 http://web.kanazawa-u.ac.jp/univ/center/ma5-2.html
1 http://plusd.itmedia.co.jp/pcuser/
1 http://plusd.itmedia.co.jp/lifestyle/articles/0811/26/news015.html
1 http://plusd.itmedia.co.jp/lifestyle/
1 http://plusd.itmedia.co.jp/games/articles/0811/25/news072.html
1 http://plusd.itmedia.co.jp/games/
```

図 4　アクセスログの解析結果

4.2.3 Web ページの解析

保存した URI の Web ページをテキストブラウザの w3m [8][*9]にてインターネットから取得する。Web ページ中の文章を KAKASI [9][*10]を使用してわかち書きにし、それぞれの語句の使用回数を調べ、使用回数の多い順にソートする。抽出された語句の中から記号や付属語、接続語などの不必要な語句を取り除く。そして、使用回数の多い語句を URI とセットにして保存する。保存した語句を図 **5** に示す。

```
http://kakaku.com/,グルメ,テレビ,ランキング
http://kakaku.com/pc/,掲載,価格,パーツ
http://www.watch.impress.co.jp/akiba/,PC,Watch,登場
http://www.k-tanaka.net/unix/cat.html,連続,ファイル,行番号
http://pc.watch.impress.co.jp/,PC,ネットブック,日本
```

図 5　保存した語句

[*9] w3m とは、テキストベースの Web ブラウザで、コマンドラインから利用できるほか、Web ページの内容を標準出力へ出力することもできる。

[*10] KAKASI とは、kanji kana simple inverter の略で、日本語の漢字仮名交じり文を平仮名やローマ字綴りの文に変換するプログラムである。単語ごとにスペースで区切る、わかち書きができる機能がある。

4.2.4 検索

保存した語句を検索エンジン Yahoo!JAPAN で検索する。検索にマッチし、表示された URI を検索に用いた語句とセットにして保存する。保存した URI を図 **6** に示す。また、Web ページのタイトルも URI とセットにして保存する。保存した Web ページのタイトルを図 **7** に示す。

```
グルメ,www.gnavi.co.jp,www.walkerplus.com/gourmet,g.pia.co.jp
テレビ,tv.yahoo.co.jp,www.tvguide.or.jp,tv.www.infoseek.co.jp
ランキング,ranking.goo.ne.jp,www.oricon.co.jp/rank,searchranking.yahoo
掲載,info.news2u.net/sample,jp.ibtimes.com/press/aboutservice.html,inf
価格,kakaku.com,kakaku.com/pc,bbs.kakaku.com/bbs
パーツ,www.gooparts.com,www.blog-parts.com,goobike.com/parts
PC,pc.watch.impress.co.jp,www.sotec.co.jp,plusd.itmedia.co.jp/pcuser
Watch,www.watch.impress.co.jp,www.watch.impress.co.jp/game,pc.watch.im
登場,www.apple.com/jp,ja.wikipedia.org/wiki/ドラゴンボールの登場人物,c
連続,ja.wikipedia.org/wiki/連続,ja.wikipedia.org/wiki/連続鋳造,www.the
ファイル,www.filebank.co.jp,www.tanteifile.com,external-file.com
行番号,ja.wikipedia.org/wiki/行番号,www.ed.noda.tus.ac.jp/j-takeda/api
PC,pc.watch.impress.co.jp,www.sotec.co.jp,plusd.itmedia.co.jp/pcuser
ネットブック,www.booknet.co.jp,www.netbook.jp,ja.wikipedia.org/wiki/Ne
日本,ja.wikipedia.org/wiki/日本,www.japanpost.jp,www.jfa.or.jp
```

図 6　保存した語句

```
www.gnavi.co.jp, ぐるなびホームページクイックリンク：関東版 - 関西版 -
www.walkerplus.com/gourmet, グルメウォーカー
g.pia.co.jp, グルメぴあホームページ
tv.yahoo.co.jp, Yahoo!テレビ.Gガイド [テレビ番組表]クイックリンク：ド
www.tvguide.or.jp, インターネットTVガイド
tv.www.infoseek.co.jp, 今日のテレビ番組を検索・BSアナログ番組・BSデジ
ranking.goo.ne.jp, ランキング - goo ランキングクイックリンク：キーワー
キーワードランキング - goo ランキング
www.oricon.co.jp/rank, ランキング-ORICON STYLE
searchranking.yahoo.co.jp, Yahoo!検索ランキング
info.news2u.net/sample, リリース掲載先 - ニュースリリース掲載・プレス
jp.ibtimes.com/press/aboutservice.html, プレスリリース掲載 - リリース
info.news2u.net/service, News2u リリースとは - ニュースリリース掲載・
kakaku.com, 価格.com - 「買ってよかった」をすべてのひとに。クイックリ
```

図 7　保存した Web ページのタイトル

4.2.5 html ファイルに出力

アクセスログから抽出した URI、Web ページから抽出した単語、そして検索エンジンにて検索し、マッチした URI とそのタイトルをまとめて html ファイルに出力する。この html ファイルはそれぞれのユーザ毎に作成される。また、html ファイルには他のユーザの html ファイルへのリンクが設置され、ユーザ間での bookmark の相互利用が可能となる。

4.3 スクリーンショット

システムを実際に動作させ、Web 上に作成された Bookmark を図 8 に示す
画面左上に表示されている “koyama”、“fuziki”、“murakami” はそれぞれのユーザの Bookmark へのリンクである。“http://kakaku.com/”、“http://kakaku.com/pc/” はアクセスログから抽出したユーザ の p 関越頻度の高い Web ページである。“グルメ”、“テレビ”、“ランキング” は “http://kakaku.com/” から抽出した使用回数の多い語句である。“掲載”、“価格”、“パーツ” も同様に “http://kakaku.com/pc/” から抽出した使用回数の多い語句である。それぞれの抽出された語句以下に表示されているリンクが、その語句を検索エンジンにて検索し、マッチした Web ページである。

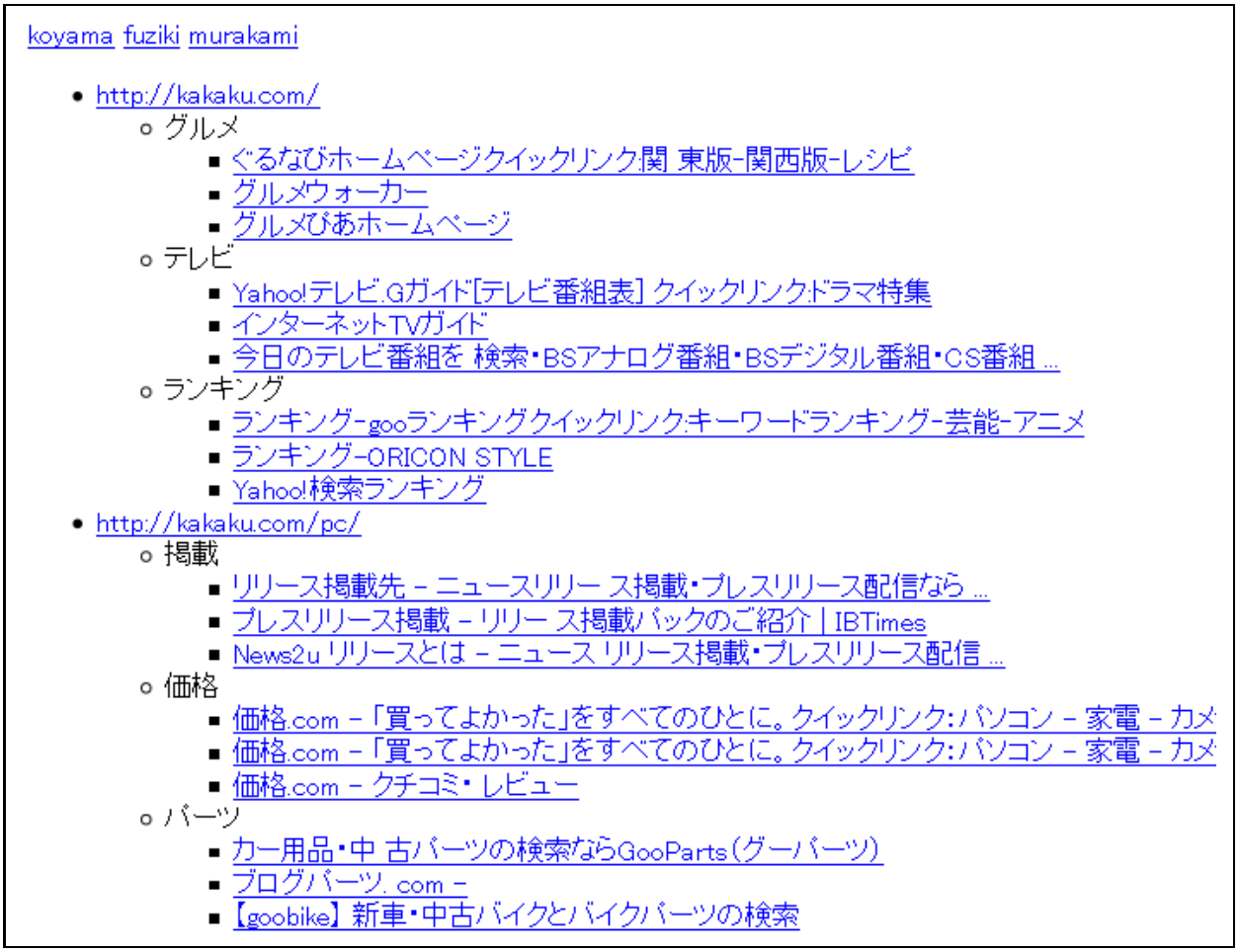

図 8　Web 上に作成された Bookmark

5 考察

システムを実際に動作させ、その結果から利点と問題点を考察する。

5.1 利点

- ユーザのアクセスログにある URI からではなく、新しくインターネットから取得してきた Web ページが Bookmark に登録されるので、思いがけない発見の可能性がある。
- ユーザが情報を収集することを意識しなくてもアクセスログから自動的に Bookmark を収集するので、ユーザにかかる負担が少ない。
- 複数ユーザ間で Bookmark の相互利用が可能なので、それぞれのユーザの興味の対象を把握することができる。

5.2 問題点

- 対象の Web ページと関連性の高い Web ページをインターネットから選別する精度が低く、アクセスログから抽出した URI と関連性の低い Web ページまで Bookmark に登録される。
- システムで利用しているテキストブラウザが UTF-8[*11]に対応していないため、UTF-8 で作成された Web ページを解析するときに文字化けが発生してしまうバグが存在する。
- 複数ユーザ間の Bookmark の相互利用は現在はリンクが相互に貼られているだけで、Bookmark 自体は独立しているため、それぞれの Bookmark 同士の関連性が低い。

[*11] UTF-8 とは、UCS、Unicode で使える符号化方式である。正式名称は UCS Transformation Format 8。
UCS とは、符号化文字集合の国際規格のひとつである。Unicode をベースにして文字空間を 31 ビットに拡張したもの。
Unicode とは、コンピュータ上で多言語の文字を単一の文字コード。

6　まとめ

開発した自動 Bookmark システムを実際に研究室内で動作させ、私個人がユーザ側の立場で使用テストを行った。他の管理者がユーザ登録をすると、Bookmark の生成や Web ページの検索などは全て自動で行われるため、一切負担を感じることはなかった。また、システムによって作成された Bookmark には、普段から見ているページと関連性の高い Web ページがいくつか登録されていた。よって、ユーザが負担感じることなくインターネットからユーザにとって興味の対象となり得る情報を収集することができたと考える。また、今回の研究で考案したユーザの興味の対象となり得る Web ページをインターネットから選別する方法は有効であったと考える。

しかし、今回の動作テストで Bookmark に興味の対象となる Web ページが登録されていたが、既に興味を持っていた分野のものばかりであり、新しい興味の対象となり得るページを見つけることはできなかった。そのため、興味の対象を広げることを支援するシステムとしては、結果を出すことができていないので未完成と言える。

7　今後の課題

現状のシステムでは、アクセスログから抽出した URI と関連性の低い Web ページまで Bookmark に登録される。そのため、対象となる Web ページと関連性の高い Web ページをインターネットから選別する精度を高めるために、語句抽出などの選別基準の見直しが必要である。見直し案として、語句抽出の際に使用回数のみで判断するのではなく、タイトルに使われている語句や見出しに使われている語句の方が文中の語句よりも Web ページと関連性が高いとし、それぞれに重みづけをして判断することで、より関連性の高い語句が抽出できると考える。Web ページを選別する精度を高めることで、関連性の低い Web ページが Bookmark に登録されることを防ぎ、ユーザ興味の対象となり得る Web ページをより多く登録できるようになると考える。

また、複数ユーザ間の Bookmark の共有についても、Bookmark に登録されている Web ページをカテゴリ分けし、カテゴリ毎に参照できるようにすることで、それぞれの Bookmark の関連性を高めることができると考える。カテゴリ分けはユーザが手動でするのではなく、それぞれの Web ページを解析し自動で行えるようにする。Bookmark の関連性を高めることで、ソーシャルブックマークのような使い方もできるようになり、情報共有によって興味の対象が似たユーザを探すことができ、そこから興味の対象を広げることもできると考える。

8 謝辞

本研究を行うにあたり、御指導及び御協力頂いた大垣 斉講師、藤井 信夫教授、情報安全工学実験室のメンバー及び卒業生の方々、その他見守ってくださった方々に深く感謝の意を表します。

参考文献

[1] Google, [http://www.google.co.jp/]
[2] Yahoo!JAPAN,[http://www.yahoo.co.jp/]
[3] squid catch[http://www.squid-cache.org/]
[4] del.icio.us[http://delicious.com/]
[5] パストラック (Pathtraq)[http://pathtraq.com/]
[6] Vine Linux Home Page[http://vinelinux.org/]
[7] BASH - GNU Project - Free Software Foundation[http://www.gnu.org/software/bash/bash.html]
[8] w3m Homepage[http://w3m.sourceforge.net/]
[9] KAKASI - 漢字→かな (ローマ字) 変換プログラム [http://kakasi.namazu.org/]

URI は 2008 年 11 月 26 日現在

付録 A　ソースコード

A.1　cron の設定

```
59 23 * * * /home/2008/nagata/sotuken/autobookmark.sh
0 0 * * 7 /home/2008/nagata/sotuken/resetlog.sh
```

A.2　Bookmark システム

A.2.1　メイン処理

```
#!/bin/bash

# アクセスログの保存
cat /var/log/squid/access.log >> /home/2008/nagata/sotuken/squid.log

# ユーザ毎に Bookmark を作成
for USER in `cat /home/2008/nagata/sotuken/user.txt`
do

# アクセスログからワードを抽出
  /home/2008/nagata/sotuken/word.sh $USER > /home/2008/nagata/sotuken/word.txt

# 保存したワード検索
  /home/2008/nagata/sotuken/search.sh

# html ファイルに出力
  /home/2008/nagata/sotuken/htmlout.sh $USER | nkf -j > \
  /home/2008/nagata/public_html/sotuken/$USER.html
done
```

A.2.2 アクセスログ解析処理

```
#!/bin/bash

# 観閲回数の多いアクセスログを抽出
for URI in ‘cat /home/2008/nagata/sotuken/squid.log | grep " $1 " | awk ’$5 > 2000{print $7}’ \
            | grep ’\(html*$\)\|\(/$\)’ | sort | uniq -c | sort -r | awk ’{print $2}’ | head -n 5‘
do
  echo -n $URI
  echo -n $URI >> test.txt

# 抽出したアクセスログの Web ページを取得し、使用回数順にソート
  w3m -dump $URI | kakasi -w | sed ’s/ /\
/g’ | sort | uniq -c | sort -r | awk ’{print $2}’ | grep -v ’[[:punct:]]’ \
| grep -v ’^[[:digit:]]*$’ > /home/2008/nagata/sotuken/wordtemp.txt

  echo -n > /home/2008/nagata/sotuken/temp2.txt

# 以下、ノイズ除去
  for WORD in ‘cat /home/2008/nagata/sotuken/wordtemp.txt‘
  do
    echo $WORD > /home/2008/nagata/sotuken/temp1.txt

# 記号などの除去
    for NOISE in ‘cat /home/2008/nagata/sotuken/noise1.txt‘
    do
       cat /home/2008/nagata/sotuken/temp1.txt | grep -v $NOISE > \
   /home/2008/nagata/sotuken/temp1.txt
    done

# 接続語、助詞などの除去
    for NOISE in ‘cat /home/2008/nagata/sotuken/noise2.txt‘
    do
      cat /home/2008/nagata/sotuken/temp1.txt | grep -v ^$NOISE$ > \
  /home/2008/nagata/sotuken/temp1.txt
    done

    cat /home/2008/nagata/sotuken/temp1.txt >> /home/2008/nagata/sotuken/temp2.txt
    if [ ‘cat /home/2008/nagata/sotuken/temp2.txt | wc -l‘ -eq 3 ]
    then
      break;
    fi
  done
```

```
  for WORD in ‘cat /home/2008/nagata/sotuken/temp2.txt‘
  do
      echo -n ,$WORD
  done

  echo -e -n "\n"
done
```

A.2.3 インターネット検索処理

```
#!/bin/bash

# URI ファイル、タイトルファイルの初期化
echo -n > /home/2008/nagata/sotuken/URI.txt
echo -n > /home/2008/nagata/sotuken/title.txt

# 抽出したワード毎に処理を行う
for WORD in ‘cat /home/2008/nagata/sotuken/word.txt | sed ’s/,/ /g’ | awk ’{print $2,$3,$4}’‘
do
  echo -n $WORD >> /home/2008/nagata/sotuken/URI.txt

# Yahoo!JAPAN にてワードを検索
  w3m -dump_source http://search.yahoo.co.jp/search?p=$WORD > /home/2008/nagata/sotuken/ysearch.txt

# URI の抽出
  for URI in ‘cat /home/2008/nagata/sotuken/ysearch.txt | grep ’div class="abs"’ \
             | sed ’s/.*"yschurl">//g’ | sed ’s/<[^>]*>//g’ | head -n 3‘
  do
    echo -n ,$URI >> /home/2008/nagata/sotuken/URI.txt

# タイトルの抽出
    echo "$URI, ‘cat /home/2008/nagata/sotuken/ysearch.txt | grep "$URI" | grep "<h3>" \
         | sed ’s/^.*<h3>//g’ | sed ’s/<[^>]*>//g’‘" >> /home/2008/nagata/sotuken/title.txt
  done

  cat /home/2008/nagata/sotuken/title.txt | sort | uniq > /home/2008/nagata/sotuken/title.txt

  echo -e -n "\n" >> /home/2008/nagata/sotuken/URI.txt
done
```

A.2.4 html 出力処理

```
#! /bin/bash

# ヘッダ出力
cat <<EOF
<?xml version="1.0" encoding="ISO-2022-JP"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml" xml:lang="ja" lang ="ja">
<head>
<meta http-equiv="Content-Type" content="text/html;charset=ISO-2022-JP" />
<meta http-equiv="Content-Style-Type" content="text/css" />
<title>bookmark test</title>
</head>
<body>
EOF

# 他ユーザの Bookmark へのリンク
for USERNAME in ‘cat /home/2008/nagata/sotuken/user.txt | grep -v $1‘
do
  echo "<a href = $USERNAME.html>$USERNAME</a>"
done

echo "<ul>"

for LINE in ‘cat /home/2008/nagata/sotuken/word.txt‘
do

# アクセスログから抽出した URI、ワード、提案する URI とそのタイトルの出力
  echo -n "<li>"
  MAINURI=‘echo $LINE | awk -F, ’{print $1}’‘
  echo  "<a href = $MAINURI>$MAINURI</a>"
  WORD1=‘echo $LINE | awk -F, ’{print $2}’‘
  WORD2=‘echo $LINE | awk -F, ’{print $3}’‘
  WORD3=‘echo $LINE | awk -F, ’{print $4}’‘

  echo "<ul>"

  for WORD in $WORD1 $WORD2 $WORD3
  do
    echo "<li>$WORD"
    echo "<ul>"
```

```
    URI=‘cat /home/2008/nagata/sotuken/URI.txt | grep "^$WORD," | uniq | awk -F, ’{print $2}’‘
    echo "<li><a href =\"http://$URI\">‘cat /home/2008/nagata/sotuken/title.txt | grep "$URI" \
         | awk -F, ’{print $2}’‘</a></li>"
    URI=‘cat /home/2008/nagata/sotuken/URI.txt | grep "^$WORD," | uniq | awk -F, ’{print $3}’‘
    echo "<li><a href =\"http://$URI\">‘cat /home/2008/nagata/sotuken/title.txt | grep "$URI" \
         | awk -F, ’{print $2}’‘</a></li>"
    URI=‘cat /home/2008/nagata/sotuken/URI.txt | grep "^$WORD," | uniq | awk -F, ’{print $4}’‘
    echo "<li><a href =\"http://$URI\">‘cat /home/2008/nagata/sotuken/title.txt | grep "$URI" \
         | awk -F, ’{print $2}’‘</a></li>"
    echo "</ul>"
  done
  echo "</ul>"
  echo "</li>"
done

# フッタ出力
cat <<EOF
</ul>
</body>
</html>
EOF
```

A.3 アクセスログのリセット

```
#! /bin/bash
#
echo -n > /home/2008/nagata/sotuken/squid.log
```