検索支援システムに関する研究

01H078 チン リョウウン

目次

1　はじめに

近年 Web は一般的なものとして普及し、我々がインターネットを通じて、利用できる情報量が飛躍的に増大している。その中から自分の目的にあった情報を効率よく取り出すために、多くの人が検索エンジンを利用している。

インターネット上の検索エンジンは、インターネットで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 サイトのことを意味する。現在検索エンジンは世界中では多く存在し、よく知られ、よく使われている検索エンジン Google、Yahoo など、代表的なものとしてあげられる。しかし、このよな検索エンジンは持つデータの量が非常に多く、利用者は目的情報を得るまでに検索結果を閲覧し、検索条件を調整して再検索し、といった試行を何度か繰り返すのが普通である。そのため、効率的な検索手法に対するニーズはますます強まっている。

2　研究内容

2.1　現状分析と問題点

検索エンジンを利用し、欲しい情報を捜し出すため、キーワードによる検索方法はもっとも広く使われている。検索エンジンの利用者は主題に関するキーワードを自分で考えて検索フォームに入力し、そのキーワードを基に検索を行って結果を得る。時には思いかけず大量の検索結果を得てしまい、網羅性が高いという利点をもつ一方で、利用者は自分がどのような情報を欲しているのか、目的情報を絞り込むには時間が掛かる場合もある。また欲しい情報をはっきり意識してない場合、意味の曖昧なキーワードをいれて検索した場合、関係ないサイトを含めた検索結果も出てしまうことが多く。そのため、適切なキーワードを選ばないと、探したい情報を含む web ページを見つけることができない可能性がある。また、検索で得たほしい情報は時間を経ってから、もう一度見たい時、ウェブブラウザの bookmark や履歴に記録されない場合、再び検索しなければならないのが普通である。

ここで、本研究は前述した現状の改善を含め、グループにおける検索支援を目的として、グループのメンバが検索を行う際に、以下の問題の解決手法を提案する。

- 求める情報に関する適切のキーワード或いはヒントがほしい
- 過去に検索した情報をもう一度見たい時、再び最初から検索し直す
- 同じキーワードに対して、グループのメンバがどんなページを見たか、またアクセスしたページを再利用する

以上述べたグループとは、同じ研究室や、共通の目的をもつ集団である。このようなグループのメンバは同じ目的の情報を検索することがあると考えられる。そのため、検索キーワードを共有し、お互いに参考することが可能と思われる。

2.2 解決手法の提案

本システムでは、グループのメンバが HTTP プロキシサーバ[*1]を利用して、外部ネットワークに接続することを前提とする。

そのため、グループのメンバのアクセス情報がすべてプロキシサーバのログに記載される。その中にグループのメンバが、検索を行う際に使われていた検索キーワードもプロキシサーバのログに含まれている。こういった検索キーワードを抽出し、グループ内検索キーワードの共有をできる。

そして、プロキシサーバはグループのメンバがアクセスしたページをローカルに蓄えておくキャッシュ機能がある。プロキシサーバのキャッシュ機能とは一度アクセスしたコンテンツを、一定期間保持しておき、次に同様の接続要求があった場合、キャッシュしたコンテンツを提供することによって、ネットワーク負荷を低減し、表示を高速化することができる。

本システムはプロキシサーバのキャッシュ機能を利用し、キャッシュを検索することによって、グループのメンバが過去に検索した結果を再確認することが容易になり、同じ検索キーワードに対して、他のメンバがアクセスしたり、検索して探してきた情報を共有ができる。

本システムでは、キャッシュデータに情報が少ない場合、それを補う形で外部の検索サービス (Google API) を利用する。Google API とは Google の検索エンジンをプログラミング言語から Google の検索結果を利用できる Web サービスである。Google API を利用することによって、検索時利用者自身の意図で検索先 (Google API で検索するか、プロキシサーバのキャッシュに検索するか) を使い分けることができる。

[*1] プロキシサーバとは、「代理」としてインターネットとの接続を行なうコンピュータ、また、そのための機能を実現するソフトウェア。

3　システムの実装

3.1　プロキシサーバのログによる検索キーワードの抽出

本システムでは、グループのメンバが同じ HTTP プロキシサーバを利用して外部ネットワークに接続するため、グループのメンバのアクセス記録は、すべてプロキシサーバのログに残される。プロキシサーバのログは図 1 に表示されたものである。

図 1: プロキシサーバのログ

プロキシのログに一つ記録として以下のようなる。書面の都合上改行しているが、実際は一行である。

1105255341.743 192.168.1.11　http://www.google.co.jp/search?hl=ja&q=mysql+%E7%89%B9%E6%AE%8A+%E7%AC%A6%E5%8F%B7&btnG=Google+%E6%A4%9C%E7%B4%A2&lr=　http://www.googlco.jp/search?

記録形式は以下のようになる。

UNIX タイムスタンプ（UTC(1970/1/1) を基づいて、秒単位で表記されたもの）　アクセス元のアドレス　検索キーワードを含む request　アクセス先の URL

その中にグループのメンバが検索を行う際に使われていた検索キーワードもプロキシサーバのログに記録される。

しかし、こういった検索キーワードはエンコードされており、人間に読み易くするため、さらに抽出してデコードする必要がある。

本システムはスクリプト言語 Ruby [*2]を用いて検索キーワードの抽出、デコードの作業を行う。また抽出された検索キーワードを MySQL [*3]に入力し、管理する。

[*2] Ruby は、手軽なオブジェクト指向プログラミングを実現するための種々の機能を持つスクリプト言語である。

[*3] MySQL は、オープンソースのリレーショナルデータベース管理システム、無償で入手できる。

3.2 キャッシュデータによる検索

グループのメンバが過去に検索した結果、また他のメンバがアクセスしたり、検索して探してきた情報を再利用するため、プロキシサーバのキャッシュデータに対して検索できるようにする。本システムは Namazu [*4]を利用して、検索を行う。プロキシサーバのキャッシュデータを検索できるようにするには、以下の作業を行う必要である。

- 検索用インデックスを作る
- サーバ側の設定を行う、キャッシュデータを検索できるようにする

図 2 は Namazu によるプロキシサーバのキャッシュデータを検索した結果である。矢印が指す部分はプロキシサーバがキャッシュしたコンテンツである。

図 2: Namazu によるキャッシュデータの検索結果

[*4] Namazu とは高機能でフリーな日本語全文検索システムである。

3.3 検索サービス (Google API) の利用

GoogleAPI を利用するには、新たに設けられた「Google Account」に登録することでライセンスキーを取得する必要がある。このライセンスキーを使った Google API による検索は一日に 1,000 回までの利用が可能である。それを超えた場合は、翌日から再び検索することになる。しかし、グループにおける小規模な検索ニーズに対して、充分利用する価値がある。

3.4 全体の構造

本システムはウェブブラウザを通じて検索を行う。検索結果は**図 3** のように、大きく 3 つの部分に分かれる。画面の左部分は検索キーワード共有リストを表示する画面、右上の部分はプロキシサーバのキャッシュデータによる検索結果を表示する画面、右下部分は Google API による検索結果の表示画面。このように、同じ検索キーワードに対して、三つの検索結果が表示される。

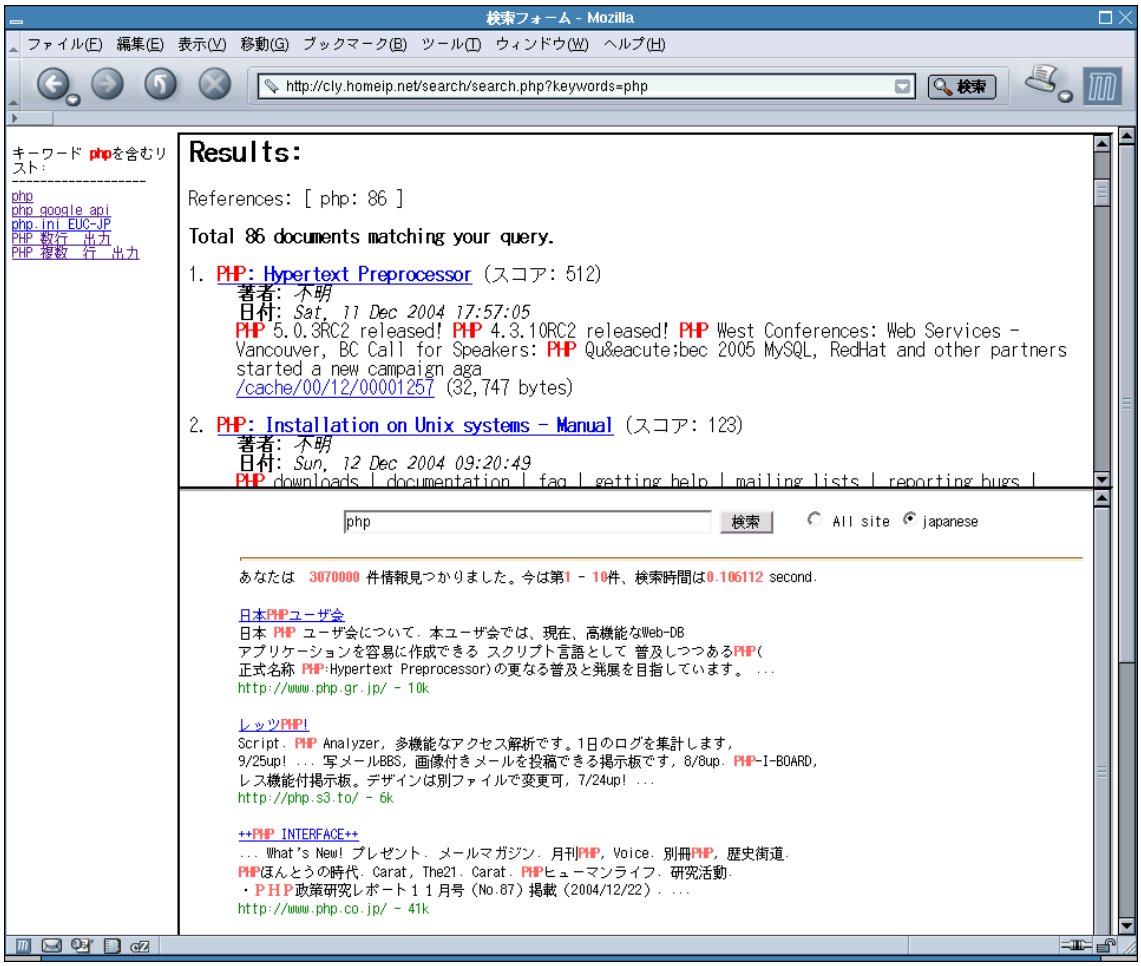

図 3: 検索支援システムによる検索結果画面

4　まとめ

本研究はグループにおける検索支援システムは、グループ範囲内の検索キーワードの共有、またプロキシサーバのキャッシ機能を利用し、グループのメンバがアクセスした情報の再利用を実現した。しかし、収集されたキーワードリストに入力ミスを含むキーワードが存在し、現時点は手作業で削除している。また、プロキシサーバのキャッシュデータによる検索結果が文字化けするなどの問題もある。このような問題の解決は今後の課題として挙げられる。

謝辞

本研究を進める上で、大垣斉 講師、藤井 信夫教授、中村 孝講師にはご指導及びご協力を戴きました。また同じ研究室、および fken.a4w Mail-list でアドバイスをしてくれた方々に深く感謝の意を表します。

5　参考文献

1.「Namazu システムの構築と活用」ISBN4-7973-2338-8
2.「Ruby レシピブック」ISBN4-7973-1408-7
3. PHP マニュアル http://www.php.net/manual/ja/
4. Google Web APIs http://www.google.com/apis/index.html
5. google hacks code http://examples.oreilly.com/googlehks/

URI は 2004 年 12 月 15 日現在

付録A　ソースコード

A.1　プロキシサーバから検索キーワードを抽出スクリプト

```
require 'cgi'
require 'iconv'
require 'mysql'
require 'date'
$KCODE="EUC"

month =Time.now.month
day = Time.now.day

object = Mysql::new('localhost','admin','clycly','referer_db')

object.query("CREATE TABLE ref_#{month}_#{day}
(id INT NOT NULL AUTO_INCREMENT ,
utc_time VARCHAR(20) NOT NULL ,
ip VARCHAR( 20 ) NOT NULL ,
keyword VARCHAR( 100 ) NOT NULL ,
domain VARCHAR( 150 ) NOT NULL ,
PRIMARY KEY (id ));
    ")

file_name = ARGV.shift
f = open(file_name)

while line =f.gets
  f.lineno
array = line.split
        utc_time = array[0]
        ip = array[1]
     key_url=array[2]
        key_domain=array[3]

next if /(q=related:)/ =~ key_url

if /(q=)/ =~ key_url
word_url = key_url.slice(/(q=).*$/)
       if n=/(q=cache:)/ =~ word_url
                  n_plus = /(\+)/ =~ word_url
      if n_end = /(\&)/ =~ word_url
```

```
                            word= word_url[n_plus+1,n_end-n_plus-1]
             else
                     n_end = /(.$)/ =~ word_url
                             word = word_url[n_plus+1,n_end-n_plus-1]
                           end
        else
                     if n_end = /(\&)/ =~ word_url
else
            n_end = /(.$)/ =~ word_url
      end
 word = word_url[2,n_end-2]

end
     # --- change the edcode ----
     p key=Iconv.iconv("eucJP","UTF-8",CGI.unescape(word))

              # --- extra the domain ---
  domain = key_domain.slice(/^(http:\/\/).*\//)
     if index=/\'/=~domain
   domain.insert(index, "'")
     end

end

object.query("INSERT INTO `ref_#{month}_#{day}` (`id`, `utc_time`, `ip`, `keyword`,`domain`)
              VALUES ('', '#{utc_time}', '#{ip}', '#{key}', '#{domain}'); ")

end   # --- while loop is end! ---
```

A.2　共通設定用スクリプト config.php

```
<?php
// 私の Licence Key
$googleAPIKey = "XXXXXXXXXXXXXXXXXXXXXXXXXX" ;

// GoogleSearch.wsdl
$googleWSDL = "http://api.google.com/GoogleSearch.wsdl";

// 検索禁止されているキーワード
$BadWords = "xxx|xxxx";

// 検索禁止されている URL
$BadUrls = "xxx.com|xxx.org";

// My site name
$Sitename = "my domain";

// copyright
$Copyright = "my info";

?>
```

A.3 Google API の検索結果を受け取るスクリプト apisearch.php

```
<?php
require("config.php");
require("nusoap.php");

$langCHK = array("", "", "");
$search = $_GET["keywords"];
$lang = $_GET["lang"];
$start = $_GET["start"];

if(!is_numeric($start)) $start = 0;
$start = abs($start);

if($search <> "") {
switch($lang) {
case "lang_ja":
$langCHK[1] = " checked";
break;
default:
$langCHK[0] = " checked";
}
$BadWords = split("\|", $BadWords);
foreach($BadWords as $aBadWords) {
if(stristr($search, $aBadWords)) {
die("この検索の内容は禁止されています！");
}
}

$soap = new soapclient($googleWSDL, true);
$soap->decodeUTF8(false);

$params = array(
'key' => $googleAPIKey,// Google license key This is a valid license.
                                //But get your own license, by going to www.google.com/api
'q'   => $search,// search term
'start' => $start,// start from result n
'maxResults' => 10,// show a total of n results
'filter' => true,// remove similar results
'restrict' => '',// restrict by topic
'safeSearch' => false,// remove adult links
```

```
'lr' => $lang,// restrict by language
'ie' => 'UTF-8',// input encoding
'oe' => 'UTF-8'// output encoding
);

$res = $soap->call("doGoogleSearch", $params, "urn:GoogleSearch", "urn:GoogleSearch");

$totalCount = $res["estimatedTotalResultsCount"];
$searchTime = $res["searchTime"];
$startIndex = $res["startIndex"];
$endIndex = $res["endIndex"];

if($totalCount == 0) {
$output = "<hr>すみません、キーワード<b>$search</b>は見つかりません！";
}
else {
$output = "<b>$totalCount</b> 件情報見つかりました。
今は第<b>$startIndex</b> - <b>$endIndex</b>件、
        検索時間は<b>$searchTime</b> second.<br><br>";
}

if(is_array($res["resultElements"])) {
foreach($res["resultElements"] as $item) {
$DocSize = $item["cachedSize"];

$DocSnippet = $item["snippet"];
//$DocSnippet = mb_convert_encoding($item["snippet"], mb_internal_encoding(), 'auto');
$DocURL = $item["URL"];
//$DocURL = mb_convert_encoding($item["URL"], mb_internal_encoding(), 'auto');
$DocTitle = $item["title"];
//$DocTitle  = mb_convert_encoding($item["title"], mb_internal_encoding(), 'auto');

$DocTitle = ($DocTitle == "") ? "no title" : $DocTitle;

if($DocSnippet <> "")
$DocSnippet .= "<br>";
$output .= "<a href=\"$DocURL\">$DocTitle</a><br>";
$output .= "$DocSnippet<span class=g>$DocURL - $DocSize </span>";
$output .= "<br><br>";
// $output .= "<span snap='$DocURL' class=hand>[cache]</span><br><br>";
}
}
}
```

```
else {
$langCHK[2] = " checked";
}

$output = preg_replace("/<B>\.\.\.<\/B>/is", "...", $output);
$output .= "<div align=center>";

$theUrl = "<a href=\"" . $_SERVER["PHP_SELF"] . "?keywords=$search&lang=$lang&start=";
          //Google API には一回検索ごとに検索結果は 10 件しか表示できない
if($start > 9)
         //前の 10 件検索記録に戻す
 $output .= $theUrl . ($start - 10) . "\">前へ</a> ";
// 次の 10 件検索記録に進む
if($endIndex - $startIndex == 9)
$output .= $theUrl . ($start + 10) . "\">次へ</a> ";

$output .= "<hr><span class=h>$Copyright</span></div>";
?>

<html>
<head>
<title><?=$Sitename?> google API   search</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<style type="text/css">
<!--
* {font-size: 12px;line-height:1.3}
b {color:#f66}
hr {color:#FF9900}
.g{color:green}
.h {color: #333}
.h a {color: #333;text-decoration:none}
.h a:visited {color: #666;text-decoration:none}
.h a:hover {color: #f33;text-decoration:underline overline}
.hand {cursor:hand;color:#00f}
-->
</style>
</head>
<body leftmargin="50" topmargin="0" >
<br>

<CENTER>
<form method="GET" action="<?=$_SERVER["PHP_SELF"]?>" name=frmSearch>
```

```
<input type=text name="keywords" maxLength=256 size=50 value="<?=rawurldecode($search)?>">
<input type="submit" value="検索">
<input type=radio name="lang" value=""<?=$langCHK[0]?>>
All site
<input type=radio name="lang" value="lang_ja" <?=$langCHK[1]?> >
 japanese<input type=hidden name="start" value=0>
</form>
</CENTER>
<?=$output?>
</body>
</html>
```

A.4　検索支援過システムの検索結果を表示するスクリプト search.php

```
<!DOCTYP:E HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
 "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=euc-jp">
<title>検索フォーム</title>
<style type="text/css">
<!--
body {
margin-left: 5px;
margin-top: 5px;
margin-right: 0px;
margin-bottom: 0px;
}
#list{
width:15%;
height:100%;
padding:1px;
margin:1px;
float:left;
font-size:12px;
}
#namazu{
width:85%;
height:30%;
padding:1px;
margin :1px;
position: relate;
top:1px;
left:15%;
}
#google{
width:85%;
height:70%;
padding:1px;
margin :1px;
overflow:auto;
position: relate;
left:15%;
}
```

```
</style>
</head>

<body>
<div id="list">
<?php
//
$keywd=@$HTTP_GET_VARS["keywords"];
echo "<BR>";
echo "キーワード <b style=color:#FF0000>".$keywd."</b>";
//
$conn=mysql_connect("localhost","xxxxxx","xxxxxxx") or die("Can not open the database!");
mysql_select_db(referer_db);
$sql = "SELECT * FROM ‘kwd_1_10‘ WHERE ‘keyword‘  LIKE ’%$keywd%’";
$result=mysql_query($sql,$conn);
$rows=mysql_num_rows($result);

if($rows==0){
echo "は共育リストにありません。<br>右下の検索エンジンを使ってください！";
}else
{
echo "を含むリスト:<br>-------------------<br> ";
while ($rows = mysql_fetch_array($result)){
 echo "<a href=http://".$_SERVER["SERVER_NAME"]."/search/search.php?keywords=".$rows["keyword"].">";
 echo $rows["keyword"];
 echo "</a>";
 echo "<br>";
}
}
 mysql_close($conn);
?>
</div>

<!--ここからは Namazu による検索フォーム  -->
<object id="namazu" data="
<?php
echo "http://".$_SERVER["SERVER_NAME"]."/cgi-bin/namazu.cgi?query=".urlencode($keywd) ;
?>
"  type="text/html" width="100%" height="30%" >
</object>
<!--ここまでは Namazu による検索フォーム  -->
```

```
<!--ここからは google API による検索フォーム  -->
<object id="google" data="
<?php
echo "http://".$_SERVER["SERVER_NAME"]."/search/api_search.php?keywords=".urlencode($keywd);
?>
" type="text/html" width="100%" height="100%" >
</object>

<!--ここまでは google API による検索フォーム  -->

</body>
</html>
```