Intelligent Spider’s Algorithm of Search Engine Based on Keyword

Yajun Du\textsuperscript{1,2}, Haiming Li\textsuperscript{3}, Zheng Pei\textsuperscript{1}, and Hong Peng\textsuperscript{1}, Non-members

ABSTRACT

It is very difficult for search engine to find the exact information, which give expression to user’s interests. Because the user’s interest changes as time lapses, there is not a mathematical mode to express user’s interests. At this paper, we discuss a disfigurement of the current search engine. Personal intelligent search engine (PISE) is proposed. We set forth its system architecture and basic principle and program interface. The experiment makes clear that the more times user use PISE, the wiser PISE become. An intelligent spider has three characteristics: (1) Intelligently guarded by the user interests; (2). A much smaller user search space of internet; (3) online with the query keywords and history interest. It can understand means what user query and analyze and conclude and crawl intelligently internet by user history interest; And so, the spider spends 210 minutes, the search time breaks the 28 day’s record of google spider.

Keywords: Crawler’s algorithm; Crawler’s program; Intelligent Spider

1. INTRODUCTION

Computer network robot is named also computer robot or spider, wanderer, web crawler. The spider acts as information retriever for some web service systems and search engines. The developer regards linkage of the web page as a graph structure. The Breadth-First Search (BFS) and Depth-First Search (DFS) algorithm is adopted to crawl from a web page to other web page. In 1993, Colorado University have developed the first generation spider, named the world wide web worm, it’s function realize to statistics the number of web server, there after the system is used to catch User Resource Locate (URL) in internet. In 1994, Jerry Yang and David Filo have developed a program to catch UR, which named Wander, in same year, P. DeBra and R. Post, have developed information retrieval system, named TueMosaic[1], the user can input keyword, the TueMosaic retrieved the information suiting the user query keyword, Pinkerton reported a highly speed web crawler[2] Finding what people want. These system considered that an information retriever can start from some preliminary Url, and spread all over internet by web page linkage structure, they first put forward the viewpoint which match user keyword to the anchor text of web page. In 1995, a webcrawler[3] system to load down the web page was developed successful. In search engine, the information retriever based on the Washington webcrawler system was widely developed. In 1997, Hsinchum Chen put forward an individuality spider [3] based on web page context, they computed a page rank in queue, made use of hopfield net and genetic algorithm to solve lots of problem crawling all internet. In 1998, Christopher C. Yang and Jerome Yen put forward Simulated annealing spider to solve lots of problem crawling all internet web page [4]. In order to improve the search quality, the people start to design the spider algorithm and program with artificial intelligence method, some intelligent spider is listed: in 2001, Bernd Thomas first introduced artificial intelligence into spider research, analyzed the intelligent action of spider, put forward AI spider [5], in 2002, Gautam pant put forward Your Own Intelligent Web Crawlers [6]. In order to improve crawling speed, Shkapenyuk and Vladislav designed the distributed Web crawler [7,8].

Search engine become an important tool how we reasonably and efficaciously, real-time make use of the Internet information resource. The technology about search engine, for example, search, index, select, etc became an important item in fields of information science and database. In search engine such as Google, Alta vista, some urgency problem should be solved, for example:(1). Low recall rate, (2). Low coverage rate, (3). Low precision. The main reasons see below: (1). In implement style of retrieving information, the currently search engines crawl periodically every web page [3]. But, web pages are distributed across multiple web servers, Web masters continue to update, add and delete web pages. Search engines cannot retrieve information of these web pages in time, and so it leads to decrease the recall rate and precision of search result etc.

(2). The currently search engines are mainly keyword-based. It cannot satisfy many informational
requirements of their users. These search engines accept some query keywords from the user, and then select from one or more index databases. Despite of its simplicity, these engines typically return millions of documents in response to a simple keyword query, which often makes it impossible for an user to find the needed information.

(3). The information search space is all web pages in Internet. But the user query keywords are relational to a specific themselves knowledge fields, and is not all web pages in internet. But the current search engines try to process all web pages in Internet, then parse them and pick up information, establish key word index database. Thereby, it causes that some irrelevant information return to the user, and the same information return to different user whose they input the same key words.

In order to solving above problem, some search engines put forward lots of intelligent method retrieving information from Internet. There are some methods: Hopfield net spider [8] in neural network, genetic algorithm spider, simulated annealing spider [3], FAQ(frequently asked questions) spider [3]. On the other hand, some search engines adopted to classify technology on search result, such as PWM [5]. But these methods only solve ultimately the surface, cannot solve the essential. At this paper, we bring forward a mode of search engine objecting to personal interest. Search engine suiting for every user interest, named personal intelligent search engine (shorting for PISE). The search engine takes on characteristics:

(1). Real-time search. When users input a keyword, a spider immediately crawl Internet URLs of user knowledge domain. It can retrieve information of her requirement existing in Internet. We conquer a difficulty of the currently search engine, which crawls Internet by a cycle and it brings on useless information;

(2). Search process with user personal interest and topic. PISE substitutes for the tradition search engine with personality, is truly search engine based on user knowledge.

(3). Rapid speed. In these search engines such as Google, Yahoo, Chinaren etc, it spends almost 28 days to crawl all web pages in Internet. PISE crawls the user domain only 1 to 20 minutes.

(4). High satisfaction degree between the search result and user’s query requirement.

On our viewpoint, the crawling speed and quality of the excellent spider is not only relational to an excellent algorithm and it’ Implement of program, but also the user intelligent action and intelligence is important in process of search information. We researched the user’s history sequence and structure which user click the web page in a search engine, put forward the interest rank concept of a web page. Meanwhile, we studied the process which the user searches information in Internet. A thought, W.Y.Zou, and Y.Y.Wu, “the anchor text takes priority, title takes second place, content of web page takes last” is put forward, it can be used to match the user query keywords; we considered also the user knowledge fields were limited, and so it is important to construct a search tree space of user query keywords. Finally, we design a real time algorithm with user keyword and user history interest to crawl web page in the user search space.

2. THE CRAWLER BASED ON WEB PAGE LINKAGE STRUCTURE

We can define the relation among web pages in internet as a directed graph $G=(W,V)$, $W$ is a set of web pages in internet, $V$ is a set of urlij that the linkage exists between web pagei and web pagej. An information retriever is necessary to search web pages in internet. The implement method of information retriever depends mainly on the linkage structure among web pages. The linkage structure based on two Assumptions [8]:

Assumption 1. A hyperlink from web page A to web page B is a recommendation of page B by the author of page A.

Assumption 2. If web page A and web page B are connected by a hyperlink, then they might be on the same topic.

And so, the page rank value of every web page is computed by Eqs.(1). The principle of spider can express by figure 1[8], the main steps were listed [9]:

![Spider based linkage structure of web page](image)

Setp1.Initializes URL table of spider,
Setp2.GETs web page order by page rank of URL,
Setp3.Load down the content of web page including user query requirement,
Setp4.Parses new URL in web pages retrieved from Internet, and adds the new URL and it’s keywords into some index database,
Setp5.Updates the page rank by computing page rank, and adds some URL which it satisfactory user.
query requirement to URL table of spider, go to step2, until not new URL is found.

3. AN INTELLIGENT SPIDER

When the user uses the spider at commencement, the spider is a fool program. However the spider intelligence can improve as user uses the spider time and times. The spider can collect the user actions, save the knowledge to a database (web user log), and so we can make fully use of these history knowledge, it makes the spider wiser and wiser. The structure of web user log is:

(User-id, Url, Date, Start-time, End-time, Keywords, Interest-rank)

3.1 User Search Space

When the user input some query keywords to the currently search engine, the search result directly is selected from an index database in which every record have stored the URL and keywords of web page of URL in web page server. The URL and web page have retrieved by spider of search engine. And so, in these style, the search space of the user query keywords is apparently made up of all web pages in internet. But we regard that the spider should be materialized some factors, in order to improve the search precision which the currently spider spreads all internet. In order to improve the search speed, the currently search engine matches completely user query keyword and the search result. But it must be changed these approaches to spread and match, the main reasons see below:

(1). The user query keywords are relational to the user history knowledge, the spider should make fully use of the user’s history knowledge of web user log in process of crawling web page.

(2). The search result should satisfy user query requirement. For the user, it is impossible to need amount of web page in short time, the user want to find some fewness, almightiness, exactness, real-time information, where fewness means that the limited and usefully information is return; almightiness means that the information relating to user query keywords will be return; exactness means that the user will get all information of his query requiring information; real-time means that the user should get some new information and wait result in short time.

(3). The user query keywords reflect a special meaning and special information space, the space should be constructed by a few part of web pages in internet. These web pages are relational to user query keywords.

Sum up above reason, we thought the web pages, which constructed the search space of the user keywords, forms a forest (see fig2), it makes up of some interest tree in which every nodes appends a special mark note-URL. A spider can only crawl in fields of web pages in the tree. How to generate the interest tree? The spider starts to work from an initiated URL table, every URL in an initiated URL table is root node of the interest tree, the other nodes of the interest is these web page crawled by the spider in accord to the strategy.

![Fig.2: The user search space](image)

3.1.1 Constructing an Initiated URL Table

An initiated URL table is made up of two parts URL of web pages:

Select url for UserLog where userid =loginser and interest_rank > \

![Fig.3: Constructing an Initiated URL Table](image)

(1). When the user uses the search engine, at first, we offer a client interface program to collect user interest, the interface includes: 1). The user usually uses the user resource locate of web page, and it’s main content, 2). The user is usually interested on the user resource locate of web pages, and it’s main content. We pick up some keywords of these main content and write these keywords and URL into web user log database.

(2). Web user log database have recorded the user history visiting web page, we can mine some interest URL, on accord to URL clicked sequence, and compute interest rank. When user queries keywords, we
match user query keywords to keywords of web pages, and create a initiated URL (fig 3)

3.1.2 Generating the User Interest Tree

A search space of the user keywords \( G', G' \subseteq G, G' = \{ T_1, T_2, \ldots, T_n \} \) is an interest tree, the root node of every tree is an URL which is stored in initiated URL table. Every nodes of interest tree is the URL of web page which the spider crawls from URL T by DFS, in process of search web page, the spider must abide by the strategy – the anchor text takes priority, title takes second place, content of web page takes last. Every node of interest tree is consider as the interest node, because these node’s web page is relational to user query keywords, the interest rank of every nodes is used to make a choice that the spider crawl URL order. the interest rank of every nodes is computed in accord to records of web user log.

3.2 Compute Interest-rank

In the currently Search engine such as google, the page rank [7] of web page, based on the linkage structure, was put forward. It is an index scaling a web page important degree. The main thought of page rank is that the important the web page become, the more the web page link other web page. And so Jill Whalen and Yuri Baranov put forward a calculating formula (see eqs. (1)of page rank(t the page rank of web page A is denoted PR(A)) [8,9].

\[
PR(A) = (1 - d) + d \left( \sum_{i=1}^{n} \frac{(PR(T_i))}{C(T_i)} \right)
\]

Where (1) \( PR(A) \) is Page Rank value of web page, (2) \( d \) is an attenuation factor, often equal to 0.85 in process of computing page rank, (3) \( PR(T_1), PR(T_2), \ldots, PR(T_n) \) are the page rank of web page \( T_1, T_2, \ldots, T_n \), these web pages must link to web page A. (4) \( C(T_1), C(T_2), \ldots, C(T_n) \) is link-out number of web page \( T_1, T_2, \ldots, T_n \).

In order to effectually compute the page rank of every web page, A reiterating algorithm (see eqs. (2)) is adopted in the currently search engine. The page rank of web page A is not relational to the initiated value of web page \( T_1, T_2, \ldots, T_n \).

\[
PR(A)^{t+1} = (1 - d) + d \left( \sum_{i=1}^{n} \frac{(PR(T_i)^t)}{C(T_i)^t} \right)
\]

The usage of page rank lies in the order of web pages which the currently search engine returns the user interface, it is important to improve the query result. In this paper, we put forward the computing interest rank algorithm (see Algorithm 1) which can mines the user history interest. The algorithm is based on google’s page rank.

Some factors and the basic assumptions which the user is interested in web page A is discussed:

1. The times and time which the user clicks web page A. The more times and time the user clicks, the more is interested in web page A.
2. The times and the time which the user clicks and browses the other web page relating web page A. The web pages relating to web page A are some web pages which the user clicks web page sequences from web page A start. For example, a clicking sequence of the user is \( url_A \rightarrow url_B \rightarrow url_C \), the sequences indicate that the user is interested web page A. At same time, the user is interested in web page B and web page C.
3. If it exists an \( url_A \rightarrow url_B \rightarrow url_C \) sequence, then the sequence indicates that the user first is interested in web page A, and web page B, web page C, the interest rank of web page A should is allocated to web page B and web page C, it will weaken the web page A. When the interest rank of web page B is computed, it will weaken the web page B and allocate to web page C, etc.

The interest rank of web page A ( IR(A) ) is computed by Eqs. (3).

\[
IR(A)^{t+1} = (1 - d)^*IR(A)^t + d \sum_{i=1}^{n} \left( \frac{IR(T_i^t)}{C(T_i^t)} \right)
\]

Where \( d \subseteq (0,1) \), \( IR(T_i) \) is the interest rank of web page relating to web page A, \( C(T_i) \) is the clicking times of web page \( T_i \).

Algorithm 1:

Step1.Select url, date, start-time, end-time, interest-rank From User log Where User-id=loginuser Group by date Order by start-time Into Compute-table.

Step2.Construct the space of user query keywords

\[
IG = (U, W),
U = \{Url_i | i = 1, 2, \ldots, n\}, 
W = \{Url_i \downarrow_{dj} Url_i\}
\]

Step3.Compute interest-rank of Url_i

3.1 Initialize interest-rank

\[
Url_i, IR \left\left\left\left. (TotalTime(Url_i)) / \sum_{i=1}^{n} TotalTime(Url_i) \right) \right) \right) \right)
\]

3.2 Iterations by Eqs.(3)

Step4.Update weblog,interest-rank. Example1. the username is zhang saving in web user log file (table1). We can construct a user clicking URL figure (fig 4), the initialized interest rank of UrlA, UrlB, UrlC, UrlD are IR(A)= 0.54, IR (B)= 0.17, IR (C)= 0.17, IR (D)= 0.12.

After it reiterative 23 times, the interest rank of UrlA, UrlB, UrlC, UrlD are:

\[
\begin{align*}
\text{interestrank(A)} &= 0.477290511842211, \\
\text{interestrank(B)} &= 0.285229053635642, \\
\text{interestrank(C)} &= 0.770112397602665, \\
\text{interestrank(D)} &= 0.285229053635642,
\end{align*}
\]
Table 1: The web log record of user zhan

<table>
<thead>
<tr>
<th>Userid</th>
<th>Url</th>
<th>Date</th>
<th>Starttime</th>
<th>Endtime</th>
<th>IIR</th>
<th>EIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhang</td>
<td>A</td>
<td>11-09-2003</td>
<td>8:03:05</td>
<td>8:04:05</td>
<td>0.54</td>
<td>0.477</td>
</tr>
<tr>
<td>Zhang</td>
<td>B</td>
<td>11-09-2003</td>
<td>8:04:05</td>
<td>8:10:05</td>
<td>0.17</td>
<td>0.285</td>
</tr>
<tr>
<td>Zhang</td>
<td>C</td>
<td>11-09-2003</td>
<td>8:10:05</td>
<td>8:12:05</td>
<td>0.17</td>
<td>0.770</td>
</tr>
<tr>
<td>Zhang</td>
<td>A</td>
<td>11-09-2003</td>
<td>8:12:05</td>
<td>8:18:05</td>
<td>0.54</td>
<td>0.477</td>
</tr>
<tr>
<td>Zhang</td>
<td>A</td>
<td>11-10-2003</td>
<td>21:32:05</td>
<td>21:33:05</td>
<td>0.54</td>
<td>0.477</td>
</tr>
<tr>
<td>Zhang</td>
<td>D</td>
<td>11-10-2003</td>
<td>21:33:05</td>
<td>21:37:05</td>
<td>0.12</td>
<td>0.285</td>
</tr>
<tr>
<td>Zhang</td>
<td>C</td>
<td>11-10-2003</td>
<td>21:37:05</td>
<td>21:40:05</td>
<td>0.17</td>
<td>0.770</td>
</tr>
<tr>
<td>Zhang</td>
<td>A</td>
<td>11-11-2003</td>
<td>12:25:05</td>
<td>12:28:05</td>
<td>0.54</td>
<td>0.477</td>
</tr>
<tr>
<td>Zhang</td>
<td>C</td>
<td>11-11-2003</td>
<td>12:28:05</td>
<td>12:29:05</td>
<td>0.17</td>
<td>0.770</td>
</tr>
<tr>
<td>Zhang</td>
<td>A</td>
<td>11-11-2003</td>
<td>12:29:05</td>
<td>12:31:05</td>
<td>0.54</td>
<td>0.477</td>
</tr>
</tbody>
</table>

3.3 The Crawling Algorithm of Spider

When the user input some query keywords, the intelligent spider starts to crawl in Internet along a direction which an URL the initiated URL table marks up. Because the spider crawls web page in real time, the crawling speed become very important, the crawling speed have two infective factor: the crawling strategy and the crawling algorithm.

3.3.1 The crawling strategy

The Crawling strategy is a principle matching the user query keywords to web pages. At this paper, “the anchor text takes priority, title takes second place, content of web page takes last” is put forward. The crawling strategy can describe below:

if url exist a anchor text then
  if matching(keyword, anchor text) then crawl url
else if url not exist a anchor text or not matching
  (keyword, anchor text) then
  if matching(keyword, title text) then crawl url
else if url not exist title text or not matching
  (keyword, title text) then
  if matching(keyword, web page text) then crawl url
else stop crawl

(1). Anchor text

The web page in internet is linked together by linkage each other, the format of the linkage structure in html document and asp document, xml document is : ⟨ a href="http://www.xihua.edu.cn") My University (/a). Where “ My University” is anchor text, if the user query keywords is matched to anchor text, then the URL(http://www.xihua.edu.cn) become the URL which we want to search for the user query keyword.

(2). Title text

The title text of web page usually substitute for a web page content, The title text in html document and asp document, xml document appear the head of text as ⟨ tile⟩ student number ⟨/ tile⟩, where “ student number” is the title text. If keyword appear in the title text then the url become the url which we want to search for the keyword.

(3). Web page text

Web page content includes two aspects information: The keyword vector is got to adopt the word frequency statistics. The keyword vector can represent the main content of the web page. On other hand, some linkages of web pages can be picked up, if the keyword of web page match absolutely or alike to user query keyword, then we consider the URL and the keyword vector as the user query keywords requirement.

3.3.2 The Crawling Algorithm

When the spider starts to work, we select some URLi from user web log database. Thereafter, we construct an interest tree Ti from URLi. The number of URLs which we select URL from the user web log database decides the number of the tree in G’.

In order to improve the crawling speed of the spider, we adopt the agent program or the multiple-thread program. Every agent or thread program competes to crawl the web page of an interest tree. A key technology of the intelligent spider is how it constructs the user interest tree. According to strategy matching user keyword with web page, we make use of

Fig.4: User interest graphic
two kinds of linkage of web page (out-linkage and in-linkage) and decide the crawling directs of the spider. Two data stack structure (wait stack and complete stack) is adopted to distinguish the crawling URL and completed URL. The working principle of intelligent spider is divided into five steps (fig 5).

1. **GetDocument(url:string)** snatches web page in internet and parses some useful URL, and return a text, some URL.
2. **MatchAnchor(URLS)** matches the user query keywords with anchor text of the parameter URLs and pushes the URLs, which user query keywords match successful with anchor text of parameter URLs, into the waiting stack, if user query keywords does not match successful with anchor text of parameter URLs then spider call **MatchTopic(URLS)**. It can match the user query keywords with the title text of parameter URLs, if the user query keywords matches successful with the title text of parameter URLs, then the spider program pushes these URL into the waiting stack, if it failed, then the spider calls **MatchContent(URLS)**. **MatchContent(URLS)** returns the URL which user query keyword matches successful with title text of parameter URLs, if the web page content of parameter URLs matches successful with the user keyword, then the spider push these URL into the waiting stack and save the keywords of web page to database which spider return to the user, else stop spider to continue.
3. **Analyzer(f:text)** analyzes web page document. This function picks up some URL and syncopates words, finds some main keywords by the word frequency statistics method and saves (url,keyword) to database which spider return to user.
4. The completed stack will save some URLs, which the spider have crawled URLs of web pages. Some URLs which save the waiting stack, can’t save the completed stack. If the URLs have saved the completed stack, the spider gives up these URLs and takes out her URLs from the waiting stack.
5. It is availability to crawl from one of Initializes URL table and spread all over the user interest tree, and so the waiting stack and completed stack is designed.

**Example2.** The user interest tree(fig 6a), the movement of the waiting stack in process of spider show fig 6b. Algorithm: IntelligentSpider

```
INPUT
url; // one of Initializes URL table
WaitF,CompleteF; // a queue
ResultBuff;  Document //textfile
URLS;//Array String
BEGIN
    CompleteF ← url; WaitF ← url
    URLS ← NULL
    Document ← GetDocument(url)
    WHILE NOT Empty(WaitF ) DO
        BEGIN
            URLS ← Analyzer(Document,KeyWords)
            ResultBuff ← KeyWords
            If (URLS1 ← MatchAnchor(URLS))
                Popqueue(URLS1,WaitF)
            Else If (URLS1 ← MatchTopic (URLS))
                Popqueue(URLS1,WaitF)
            Else IF (URLS1 ← MatchContent (URLS))
                Popqueue(URLS1,WaitF)
            url ← Pushqueue(WaitF)
            If url in URLS
                url ← Pushqueue(WaitF)
            Else
                Document ← GetDocument(url)
        END WHILE
    END.
    Output ResultBuff.
```

### 4. THE SYSTEM ARCHITECTURE OF PISE

Because these are the rich information in Internet, under condition of the current network speed, it is an impossibility that the spider spends a little time to
spread all over Internet. But, in order to obtaining the real-time information and improving the coverage rate and precision of search result, architecture of search engine must break away from the tie of currently search engine architecture (fig 7).

These are three stages in currently search engine architecture. First, the user input his query keywords by user interface of the search engine. The search engine sends the query expressions of these keywords to server by user computer. The search engine server select the query expressions from information DB and return some information (user’s web document). Secondly, the spider program running in the search engine server crawls periodically web page in Internet and retrieves web document. Thirdly, the web document parsers distill some keywords from web document and write them into information db. Then, the service system of the currently search engine is a classical browse/server model.

Fig.7: The architecture of the currently search engine

The currently search engine cannot embody the man intelligence of information search. In order to put some intelligence into search engine, we developed the personal intelligent spider (PISE) by above intelligent spider algorithm. The PISE model can divide into four stages (fig 8). First, user input keywords by user interface of search engine. The search engine sends the user query expression to user’s knowledge db in the search engine server. Secondly, the server selects the user query expression from knowledge DB, and generates user’s URLs domain (initiated URL table). Thirdly, a spider program in server of search engine crawls user’s URLs domain and cannot permit to crawls all Internet, generates the user search space. In crawling process, the web document parsers distill some keywords from web document and write into temp db. Fourth, server returns all information of the temp db to document analyser. The document analyzer recommends the best relational web page to user. The information of these web pages is displayed in user interface.

This architecture of PISE is consisted of six parts: user interface, question processor, knowledge database, user’s URLs domain, spider, web document parser, document analyzer.

1) (User Interface) The user Interface is friendly software system, which it can express accurately user’s query requirement by PISE. At same time, it display the result which PISE can correctly index the web page information. And the user interface can offer some function collecting the user interest for user knowledge database.

2) (Question Processor) The question processor analyzes relation between the user keyword of user query key words and the user URLs domain, and mines homology and distinction along some interests, and matches the user interest of keyword with his interest of knowledge and products some URLs of the user URLs domain.

3) (User URLS Domain) The user URLs domain is a two dimension relational table named UUD(url, *interest). It construct user’s initiated URL table and the spider generates the user query space of the user query keywords.

4) (Spiders) The spider is a robot program. When user submits his query keywords, the spiders start to work in Internet at once. In crawling process, the spiders make an URL attaches a user interest and parse web page and mine knowledge of web page and select pages. When a web page knowledge can not match user interest, the URL linkage is abandoned. The precision of our spider decides the user’s initiated URL table. For an isolated island in G, we is studying to let these web page into the user’s initiated URL table.

5) (Web document Parser) When the web page is crawling by spiders, immediately the web document parsers extract the web page information into some important keywords describing web page into knowledge DB. The knowledge DB is enhanced as the user visits our search engine. If the web page can match user interest expressing keyword, the vector and URL of web page and href information return into a temporarily database. In other hand, the web document can identify URL linkage. If the web page cannot match user interest with the web page pass, the Spiders continue other URL of the user URLs domain.

6) (Document analyser) Document analyzer can analyze user interest and recommend some web
pages by user query keywords. According to user history record storing in user web log database, it decides information of the temporarily database to divide into some scheme and recommend a finest result to user.

The spider is the general model, in the future, we let spider learn the mankind intelligence to search information. There are five intelligence behaviours about human being: fell behaviour, memory behaviour, study behaviour, thought behaviour, comprehension behaviour, these behaviour will embody PISE.

5. EXPERIMENT

We program PISE written by jbuilder8.0. We select amount of the suitable volunteers (5000 student, they are studying in xihua university) from different realms and test queries (100000 user query keywords, every volunteer 50 group query keywords) substituting for user Interest, we get Initiated URLs table for every user and key words group. In experiment, we consider to number (10, 20, 30, 40, 50) matching web page and keywords as the terminating condition of spider, and retrieve the crawled URL number, the crawled Time, the crawled Depth, crawled Width. We choose some import test result data which can substitute for other test result: user query keywords, such as computer, architecture, blues, gardening, etc, and user, such as 4503-th user and 20-th user. From a great deal of result, it makes know that our spider is provided with the high speeds (table2):

The average search time is about 11.4 to 107.8 Seconds. The average crawled depth is about 9 to 32 web pages. The average crawled width is about 5 to 102 web pages. In comparison with the currently spider, such as google and web crawler, the crawling speed is improved. The intelligence of PISE is demonstrated in table2, the 4503-th user is interested in computer science and 20-th is interested in architecture design, they submit same key words: computer, but PISE returns the different result. In differential time, the PISE return the web pages which the more and more is close to the user interest, the less and less time is spent to crawl user URLs domain.

ACKNOWLEDGMENT

The project was supported by National Natural Science Foundation (Grant No. 60474022), and science and technology department of zhejiang province Grant No. 2005C31005). This extended paper is based on [10] and [11].

References


Yajun Du was born in 1967. He is a professor of school of Mathematical & Computers Science, Xihua University and is pursuing the Ph.D. degree in intelligent information process from southwest jiaotong university, china. His research interests include search engine, data mining, formal concept analysis.

Haiming Li was born in 1964. He is a professor of Zhejiang Forestry University. He received the Ph.D. degree in Information Engineering and Control from Southwest Jiaotong University, China in 2003. His research interests include intelligent information processing, lattice-valued logic, automated reasoning, data mining.

Zheng Pei was born in 1968. He is a Professor of School of Mathematic & Computers at Xihua University. His research interests are linguistic information in decision making, Logic algebra, and fuzzy logic-based systems.

Hong Peng was born in 1966. He is a professor of School of Mathematic & Computers at Xihua University. His research interests include data mining, support vector machines, neural networks.
Table 2: The intellect test of PISE for same query keywords and different user

<table>
<thead>
<tr>
<th>User No</th>
<th>Keywords / Initiated URLs</th>
<th>MN (S)</th>
<th>CN</th>
<th>CT</th>
<th>CD</th>
<th>CW</th>
</tr>
</thead>
<tbody>
<tr>
<td>4503</td>
<td>Key words: computer <a href="http://www.computer.com">www.computer.com</a> <a href="http://www.cs.cmu.edu">www.cs.cmu.edu</a> <a href="http://www.cs.stanford.edu">www.cs.stanford.edu</a></td>
<td>10</td>
<td>64</td>
<td>27</td>
<td>17</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>98</td>
<td>63</td>
<td>20</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30</td>
<td>126</td>
<td>66</td>
<td>21</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40</td>
<td>195</td>
<td>14</td>
<td>21</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>253</td>
<td>25</td>
<td>23</td>
<td>102</td>
</tr>
<tr>
<td>20</td>
<td>Key words: computer <a href="http://www.centrolatino.org">www.centrolatino.org</a> <a href="http://www.ph.unimelb.edu.au">www.ph.unimelb.edu.au</a></td>
<td>10</td>
<td>27</td>
<td>11</td>
<td>31</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>51</td>
<td>26</td>
<td>30</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30</td>
<td>69</td>
<td>33</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40</td>
<td>90</td>
<td>61</td>
<td>29</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>139</td>
<td>78</td>
<td>27</td>
<td>20</td>
</tr>
<tr>
<td>4503</td>
<td>Keywords: architecture www-viz.tamu.edu <a href="http://www.onearchitecture.nl">www.onearchitecture.nl</a> <a href="http://www.toysatellite.org">www.toysatellite.org</a></td>
<td>10</td>
<td>14</td>
<td>78</td>
<td>30</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>24</td>
<td>16</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30</td>
<td>37</td>
<td>24</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40</td>
<td>110</td>
<td>42</td>
<td>31</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>77</td>
<td>57</td>
<td>9</td>
<td>21</td>
</tr>
<tr>
<td>20</td>
<td>Keywords: architecture <a href="http://www.greatbuildings.com">www.greatbuildings.com</a> <a href="http://www.aplusd.org">www.aplusd.org</a></td>
<td>10</td>
<td>21</td>
<td>82</td>
<td>13</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>59</td>
<td>24</td>
<td>13</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30</td>
<td>106</td>
<td>48</td>
<td>18</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40</td>
<td>186</td>
<td>74</td>
<td>20</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>250</td>
<td>107</td>
<td>21</td>
<td>74</td>
</tr>
</tbody>
</table>

*MN: Match URL Number, CN: Crawled URL number, CT: Crawled Time, CD: Crawled Depth, CW: Crawled Width

Table 3: The intellect test of PISE for same query keywords and user

<table>
<thead>
<tr>
<th>Time</th>
<th>Initiated URLs</th>
<th>MN (S)</th>
<th>CN</th>
<th>CT</th>
<th>CD</th>
<th>CW</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005-6-20</td>
<td><a href="http://www.computerworld.com">www.computerworld.com</a></td>
<td>10</td>
<td>32</td>
<td>11</td>
<td>45</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20</td>
<td>43</td>
<td>23</td>
<td>43</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30</td>
<td>26</td>
<td>15</td>
<td>44</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40</td>
<td>59</td>
<td>17</td>
<td>44</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>70</td>
<td>25</td>
<td>37</td>
<td>14</td>
</tr>
<tr>
<td>2005-6-21</td>
<td><a href="http://www.computerworld.com">www.computerworld.com</a></td>
<td>10</td>
<td>54</td>
<td>21</td>
<td>42</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.computer.org">www.computer.org</a></td>
<td>20</td>
<td>56</td>
<td>25</td>
<td>41</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30</td>
<td>51</td>
<td>27</td>
<td>49</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>40</td>
<td>57</td>
<td>25</td>
<td>56</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>79</td>
<td>31</td>
<td>52</td>
<td>16</td>
</tr>
<tr>
<td>2005-6-22</td>
<td><a href="http://www.computerworld.com">www.computerworld.com</a></td>
<td>10</td>
<td>59</td>
<td>45</td>
<td>30</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.computer.org">www.computer.org</a></td>
<td>20</td>
<td>101</td>
<td>57</td>
<td>24</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.gz-computer.de">www.gz-computer.de</a></td>
<td>30</td>
<td>152</td>
<td>51</td>
<td>41</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.egov.pl">www.egov.pl</a></td>
<td>40</td>
<td>189</td>
<td>63</td>
<td>23</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50</td>
<td>201</td>
<td>70</td>
<td>19</td>
<td>21</td>
</tr>
</tbody>
</table>