Smart Web Crawler: Personalized Web Search Engine for Optimal Web Page Downloading

Shinde Rahul Machindra, Virkar Snehal Dattatray, Kaphare Shradha Subhash, Prof.Wavhal D.N.

***Abstract*— Due to the large volume of web resources and the dynamic nature of deep web, achieving wide coverage and high efficiency is a challenging issue. We propose A Smart Web Crawler which Search and Discovers Number Of centre pages from deep web and focus its trajectory towards that topic in first stage i.e Site locating due to which it avoids visiting a large number of pages. Smart web Crawler ranks websites to prioritize highly relevant ones for a given topic. After searching centre pages in first stage it performs in-site exploration by excavating most relevant links with an adaptive link-ranking in second stage. Also there was confliction occurrences according to users interest due to single user so this drawback is also avoided in personalized Web Search engine. In this paper refreshing previously downloaded pages is also minimizes due to which the total staleness of pages in the repository of a web crawler due to which number of HTTP request to servers are avoided and enhance the performance of S/m. The existing crawler issues a large number of HTTP request to web server due to which there is energy consumption and carbon footprint of web servers.**

***Index Terms*— Deep web, two-stage crawler, carbon footprint, ranking, adaptive learning, personalization(profile based), staleness.**

I. INTRODUCTION

A Web Crawler also known as a robot or a spider is a system for the downloading of web pages. To solve this problem and to facilitate better sharing of information, standards have been developed by a variety of organizations. Web crawlers are used for a variety of purposes. Most prominently, they are one of the main components of web search engines, systems that gather a web pages, index them, and allow users to issue queries against the index and find the web pages that match the queries. A related use is web archiving (a service provided by e.g., the Internet archive), where large sets of web pages are periodically collected and archived . A third use is web data mining, where web pages are analyzed for statistical properties, or where data analytics is performed .Finally, web monitoring services allow their clients to submit standing queries, or triggers, and they continuously crawl the web and notify clients of pages that match those queries The deep (or hidden) web refers to the contents lie behind searchable web interfaces that cannot be indexed by searching engines. It is challenging to locate the deep web databases because they are deeply distributed and changes continuosly also there was name conflict with single user if any user has profession doctor and he fired a query for bank then using single user profile there was name confliction,because the doctor is interested in the blood bank so he fires the query only bank but due to single user profile it shows the results like national bank, money bank, banking exams instead of blood bank so this drawback is also covered in this paper.

II. LITERATURE SURVEY

1. "An active crawler for discovering geospatial Web services and their distribution pattern – A case study of OGC Web Map Service."
Author-Wenwen Lia , Chaowei Yanga and Chongjun Yangb.
Abstract-

The proposed crawler achieves good performance in crawling efficiency and results' coverage. In addition, an interesting finding regarding the distribution pattern of WMSs is discussed. We expect this research to contribute to automatic GWS discovery over the large-scale and dynamic World Wide Web and the promotion of operational interoperable distributed geospatial services.

2."Diachronic Linked Data:Towards Long-Term Preservation of structured interrelated information. Author-Theodore Dalamagas.
Abstract-

In this article we focus on the key problem of preserving evolving structured interlinked data. We argue that a number of issues, which hinder applications and users, are related to the temporal aspect that is intrinsic in Linked Data. We present three use cases to motivate our approach, we discuss problems that occur, and
Propose a direction for a solution.

.

3:-"Search Engines going beyond Keyword Search: A Survey"
Author-Mahmudur Rahman.
Abstract-

This paper tries to identify the major challenges for today's keyword search engines to adapt with the fast growth of web and support comprehensive user demands in quick time. Then it surveys different non-keyword based paradigms proposed, developed or implemented by researchers and different search engines and also classifies those approaches according to the features focused by the different search engines to deliver results.

From this paper we have referred:- Concept of keyword based search.Semantic web search and question and answering system

4."Personalization on E-Content Retrieval Based on Semantic web services."
Author- A.B. Gil1.
Abstract-

This model proposes a new approach to filtering the educational content retrieved based on Case-Based Reasoning .It is based on the model AIREH a multi-agent architecture that can search and integrate heterogeneous educational content through a recovery model that uses a federated search. The advantages of the proposed architecture, as outlined in this article, are its flexibility, customization, integrative solution and efficiency.

5." Internet Applications: The Emerging Global Computer."
Author-Technology Research News, LLC 2004.
Abstract-

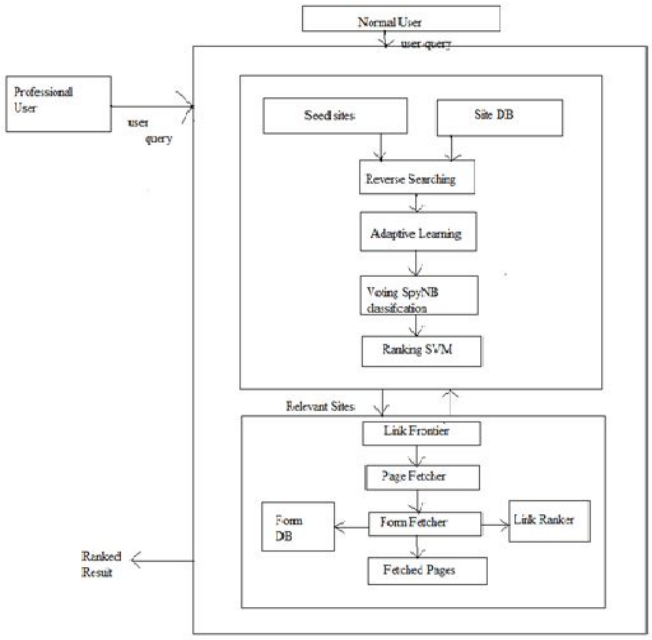
The initiative includes a sort of grammar and vocabulary that provide information about a document's components; this information will enable Web software to act on the meaning of Web content. Semantic Web software includes a special set of Extensible Markup Language (XML) tags that includes Uniform Resource Identifiers (URIs), a Resource Description Framework (RDF), and a Web Ontology Language (OWL).

III. EXISTING SYSTEM

Existing strategies were dealing with creation of a single profile per user, but conflict occurs when user's interest varies for the same query Eg. When a user is interested in banking exams in query "bank" may be slightly interested in accounts of money bank where not at all interested in blood bank. At such time conflict occurs so we are dealing with negative preferences to obtain the fine grain between the interested results and not interested. Also there was number of HTTP requests to web server due to which it increases energy consumption and carbon footprint of the web servers so computational resources are used at the time of serving the request.

IV. PROPOSED SYSTEM

To efficiently and effectively discover deep web data sources, Smart Web Crawler is designed with a two stage architecture, site locating and in-site exploring, The first site locating stage finds the most relevant site for a given topic, and then the second in-site exploring stage uncovers searchable forms from the site. Specifically, the site locating stage starts with a seed set of sites in a site database . Seeds sites are candidate sites given for Smart Web Crawler to start crawling, which begins by following URLs from chosen seed sites to explore other pages and other domains. When the number of unvisited URLs in the database is less than a threshold during the crawling process, Smart Web Crawler performs "reverse searching" of known deep web sites for centre pages (highly ranked pages that have many links to other domains) and feeds these pages back to the site database. Site Frontier fetches homepage URLs from the site database, we going to rank the relevant information also it minimize the query conflict using personalization(profile based).The number of HTTP requests to web crawler minimized and also carbon footprint decreased in proposed s/m using page refreshing policy.



Mathematical Calculation
Let us consider S as a system Smart Crawler
S=
Input-
Identify the inputs
F= f1, f2, f3 ....., fn— F as set of functions to execute commands.
I= i1, i2, i3—I sets of inputs to the function set
O= o1, o2, o3.—O Set of outputs from the function sets
S= I, F, O
I = Query submitted by the user, ...
O = Output of desired query,...
F = Functions implemented to get the output, Clustering algorithm.

Output-
Data Structure:-Link List.
Success Conditions:-
Formula for Concept Extraction:-
jcij is the number of terms in the keyword/phrase ci.
sf (ci) is the snippet frequency of the keyword/phrase ci.
ci as a concept for the query.
n is the number of web-snippets returned.
If the support of a key- word/phrase ci is greater than the threshold s
(s = 0:03), we treat ci as a concept for the query q.
So this problem is NP Complete.
Failure Conditions:-
Problem is not NP complete.
ALGORITHM
Algorithm 1: Reverse searching for more sites.
input : seed sites and harvested deep websites
output: relevant sites
while *# of candidate sites less than a threshold* do
// pick a deep website
site = getDeepWebSite(siteDatabase, seedSites)
resultP age = reverseSearch(*site*)
links = extractLinks(*resultP age*)

foreach *link in links* do
page = downloadPage(*link*)
relevant = classify(*page*)
if *relevant* then
relevantSites =extractUnvisitedSite(*page*)
Output *relevantSites*
end
end
end.

Algorithm 2: Incremental Site Prioritizing.
input : siteFrontier
output: searchable forms and out-of-site links
HQueue=SiteFrontier.CreateQueue(HighPriority)
LQueue=SiteFrontier.CreateQueue(LowPriority)
while *siteFrontier is not empty* do
if *HQueue is empty* then
HQueue.addAll(LQueue)
LQueue.clear()
end
site = HQueue.poll()
relevant = classifySite(site)
if *relevant* then
performInSiteExploring(site)
Output *forms* and OutOfSiteLinks
siteRanker.rank(OutOfSiteLinks)
if *forms is not empty* then
HQueue.add (OutOfSiteLinks)
end
else
LQueue.add(OutOfSiteLinks)
end
end
end

ACKNOWLEDGMENT

We might want to thank the analysts and also distributers for making their assets accessible. We additionally appreciative to commentator for their significant recommendations further more thank the school powers for giving the obliged base and backing.

DESIGHN AND IMPLEMENTATION

1.Modules-1

Design and Evaluation of Smart Crawler includes the following modules:

1.Module-1

Registration of user in the database of web crawler.

2.Module-2

Personalized web search.

In this module web crawler do the web searching on the basis of users profession and user got the result as per his profession.

3.Module-3

Reverse searching.

This module works on the basis of reverse searching i.e.working on the basis of UAT criterion mens do the deep web searching and provide more convenient response to the user as per uers request.

4.Module-4

Green crawling.

In this module web crawler reduces the total staleness of web pages in the web repository and dynamically update the content of web pages which is add as bookmark by the user.This module reduce the drawback of google search engine.

Conclusion

Propose an effective harvesting framework for deep web interfaces namely Smart crawler. Smart crawler is a focused crawler consisting of two stages:efficient site locating and balanced insite exploring with neglecting confliction in the user query i.e.personalization and minimizing the HTTP request to the web server and also minimizing the total staleness of pages in the repository of the web crawler.

REFERENCES

[1] An active crawler for discovering geospatial Web services and their distribution pattern – A case study of OGC Web Map Service-WenwenLia*, Chaowei Yang.

[2] Search Engines going beyond Keyword Search: ASurvey-MahmudurRahman.

[3] Diachronic Linked Data: Towards Long-Term PreservationOf Structured Interrelated Information: Sören Auer, François Bancilhon, Peter Buneman, VassilisChristophides.

[4] Personalization on E-Content Retrieval Based on Semantic Web Services -A.B. Gil1, S. Rodríguez1, F. de la Prieta1 and De Paz J.F.1.

[5] Internet Applications: The Emerging Global Computer

[6] J.Teevan, S.T. Dumais, and E.Horvitz, "Personalizing Search via Automated Analysis of interests and Activities," Proc.28th Ann.Int'l ACM SIGIR conf.Research and developement in information retrieval,pp.449-456, 2005.

[7] V.Hatzi, B.B. Cambazoglu, and I.Kousopoulos ,"Web page download scheduling policies for green web crawling," in Proc. 22nd Int. Conf. Software Telecommun. Comput. Netw. (SoftCom), 2014, pp. 56-60.

[8] Vertatique. (Oct. 15, 2009). Carbon Footprints Of Servers Can Vary By 10X [online]. Available: http://www.vertatique.com/carbon-footprints-servers-can-vary-10X.