Focused Web Crawler For Retrieving Relevant Contents

Manikandan N K[a], and Kavitha.M[b]

[a]
Research Scholar, Department of Computer Science and Engineering, Vel Tech
Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai,
TamilNadu, India.

[b]Professor, Department of Computer Science and Engineering, Vel Tech Rangarajan Dr. Sagunthala R& D Institute of Science and Technology, Chennai, TamilNadu, India.

Article History Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: The rapid growth of the World-Wide Web creates unusual scaling challenges for the purpose of general crawlers and for search engines also. we delineate a new hypertext resource discovery system which is called as Focused Crawler. The main aim of a focused crawler is to seek out the pages selectively which are very relevant to a previous defined set of topics. The topics are specified not only by using keywords, but also using prototypical documents. Instead of collecting and indexing all accessible Web documents we should also be able to answer all the possible ad-hoc queries, The main work of focused crawler is to analyze its crawl boundary to find out the links that are mostly alike for the crawl, and it avoids the unnecessary and irrelevant regions of the Web. This leads to savings in the hardware resources and network resources also and it helps the crawl for more up-to-date. A classifier that evaluates the suitable hypertext documents with respect to the relevant topics, and a distiller is introduced to filter the unwanted topic from the whole content. Focused crawling acquires relevant pages while standard crawling quickly loses its way, even though they are very similar. Focused crawling is robust against large troubles for the initial set of URLs. Focused crawling is very effective for building good quality collections of the Web documents on specific topics, using modest desktop hardware.

Keywords: Support Vector Machine, Repeated Bisection Clustering, World Wide Web, K Nearest Neighbour, Graphical User Interface Uniform Resource Locator, Hypertext Transfer protocol, Hypertext Markup Language.

1. Introduction

The internet has search engine which has at least one web crawler. some of the web crawlers are web spider and bot. The main task of a web crawlers is to crawl every web pages fed to it. The contents of web pages that had been crawled is retrieved and parsed to get the data and hyperlinks. Then the founded hyperlinks would be crawled [1]. The parser will send data to the indexer and the data is stored into the database. If we search anything in search engine it does not search into the real web site instead it searches the search engine database. The output of search engines is a list of web pages with its relevant hyperlinks[2]. The user can open web pages by clicking on the hyperlinks if they needs.

Collecting specific data from a web site is somewhat tricky than gathering contents of the web pages, because sometimes there are no hyperlinks among each entities which makes the regular crawler unable to find it. If we need data about the books automation in Library of Congress web site (http://www.loc.gov), we usually do it using search engines by giving related key words “site:loc.gov automation book” [3]. Google and Yahoo didn’t gave any matched results in the first 300 of its results. On the other hand, the searching facility in the home page of the Library of Congress web site gave us nearly 806 matched results, but we have to retrieve 41 web pages of its results to get the data. Focused web crawler also gave 806 results with fully matched output and it automatically put the data into it’s database.

Sometimes focused web crawlers are also called as vertical web crawlers or specific web crawlers [4] .It is a tool for mining specific data from web databases. The data mined are structured or semi structured because it is retrieved from the database in web sites such as databases from social networks, forums, blogs, online libraries and any web sites that uses database to display their information. If we can collect the data from a web site, then we can retrieve the information and discover the knowledge contained in that.

Overview

A search engine has become an important source forming the data in the World Wide Web (WWW). Since the web crawler is the main part of the search engine it needs to browse Web Pages that are topic specific. A web crawler is basically a software or program which browses the internet and collects data in a repository. In process of crawling the web crawler gathers Web Pages from the web and stores them in a proper way so that the search engine can retrieve them quickly and efficiently.

A web crawler starts with a URL also called as seeds which are stored in the crawler frontier. Then it identifies the hyperlinks while parsing the web pages and adds them to the list of URLs that already exists and the collected data by crawler is sent to storage[5]. This process of crawling depends on the policies defined for the crawler. The frontier consists of the list of unvisited URLs. The crawler fetches a URL from the frontier which has the list of unvisited URLs. The page which corresponds to that URL is fetched from the Web and the unvisited URLs from that page are added back to the frontier. The process of retrieving and extracting the URL goes on till the frontier is empty or some other situation causes it to stop [6]. The main job of the page fetcher is to fetch the pages from World Wide Web corresponding to the URLs which has been retrieved from the crawler frontier. For that purpose, the page fetcher requires a HTTP client for sending the HTTP request and to read the response. Web Repository stores the web pages in the database which it receives from a crawler. All other multimedia and document types are avoided by the crawler .It stores the browsed pages as different files and the storage manager stores the updated version of each page fetched by the crawler.

Generic crawlers do not specialize in specific areas. A traditional crawler periodically crawls the URLs that are previously crawled and replaces the old documents with the newly downloaded documents to refresh its collection. On the contrary, an incremental crawler refreshes the already existing collection of pages gradually by visiting them frequently. This is based upon an estimation of the rate at how often pages change. It also replaces old and less important pages by new and more relevant pages. It resolves the problem of freshness of the data[7]. The advantage of incremental crawler is that only valuable data is provided to the user.

Focused Crawler is a web crawler for fetching web pages that are related to a specific area of interest. It collects the documents that are focused and relevant to a given topic. It is called as a Topic Crawler because of the way it works. The focused crawler determines the relevance of the document before crawling the page. It estimates if the given page is relevant to a particular topic and how to proceeds. The main advantage of this kind of crawler is that it requires less hardware resources.

2 Problem Statement

2.1 Existing System

This is the basic web crawler in which the Crawling starts with a set of seed URLs and URLs to be fetched are stored in a queue structure, called “URL queue”. Then multiple threads executes simultaneously[8]. Each thread gets the required URL from the queue and fetches the related web pages from the server. Later, this page is parsed to extract the links and these links are attached to the URL queue to be fetched later. A real life crawler is much more complex than this structure to consider issues like It does not request many web pages from the same server at the same time

2.2 Issues

- The web crawler locates information on WWW, indexes all the words in a document and follows each and every hyperlink.
- .The web crawler performance is low compared to the focused web crawler. The web crawler takes more time to fetch the pages.

2.3 Proposed System

Focused crawlers uses the vertical search engines which are used to crawl the web pages relevant to the target topic. The only difference between the crawler and focused crawler is the topic classifier which makes more precise than crawler. Each fetched page is classified to the predefined target topic. If the page is predicted to be on-topic, then its links are going to be extracted and appended into the URL queue. Otherwise the crawling process stops and does not proceed from that page. This type of focused web crawler is called “full-page” focused web crawler because it classifies the full page content. It means that the context of all the links on the page is the full page content itself.

2.4 Advantages

- we spend less money, time effort processing Web Pages that are most important to do the project in most efficient way.
- The performance is high in focused web crawler.

3. Methodology

Algorithm

3.1. The Fish Search Algorithm

Fish Search algorithm is an algorithm that was created for efficient focused web crawler. This algorithm is one of the earliest focused crawling algorithms. Fish Search focused crawling algorithm that was implemented to dynamically search information on the Internet. Searching system using Fish search algorithm is more efficient than ordinary search systems as it uses a navigation strategy to automatically browse the web. the Fish Search crawling system was implemented as a client based searching system tool that automatically navigates which webpage to crawl, thereby working more like a browsing user, but acting much faster and follows an optimized strategy. Client-based crawling have some significant disadvantages, like slow operation and resource consumption of the network. The algorithm requires three types of actions:

STEP 1: The most important and difficult is the first step, it requires finding starting URLs, which will be the starting point of searching process.

STEP 2: Web documents are extracted and scanned for the information which is relevant at the receiving end.

STEP 3: The extracted web documents are reviewed to find links to other web documents or URL’s of web pages. The main principle of the Fish Search algorithm can be expressed as the following: at first it takes input data which is seed URL and query, and creates a priority list in dynamic way (according to the seed URLs) of the next web links to be explored. At every step of creation of the list, the first node is retrieved from the URL list and processed. As each text of the documents be available, it is processed and analysed by a scoring tool ranking whether this document is relevant or irrelevant to the search query and, considering the score, the algorithm decides whether to continue the exploration in that route or not: Whenever a web data source is fetched, it is scanned for URLs. The URL nodes acuminated to by these links are each assigned a value of depth [9]. If the parent URL is marked as relevant, the depth of the children URL is set to some predetermined value. Else, the depth of the children URL is set to be one less than the depth of the parent link. When the depth reaches 0, the direction is dropped and none of its children URL is included into the URL list.

The Fish Search system use a principle of the fish school metaphor, where URL presented like a fish, searching for food and producing children (called width). They move in the food's direction. The distance that fish and children can go without food is called depth. When they find fish they live longer and reproduce a lot. If they do not find any food they die and children also. And also polluted water have negative impact on their destiny. The Fish Search criteria is based on the keyword search, regular expression search and external filters. In spite of the advantages of the system, the fish search system have some limitations in time consuming, usage of network resources and it can't be available other WWW browser users.

3.2. The Shark Search Algorithm

The shark search algorithm is an improved version of the Fish Search algorithm. While this algorithm uses the same simple Fish School metaphor, it discovers and retrieves more relevant information in the same exploration time with improved search abilities. The Shark Search system uses better relevance scoring techniques for neighbouring pages before accessing and analysing them. The system have a significant impact on search efficiency because of improvement of the relevance classification system. It uses a score between 0 and 1, instead of the binary evaluation of information relevance. This approach gives much better results than binary classification. The second improvement is the method of inheritance of node's children. System gives every child an inherited score that have a huge impact on the relevance score of the children and children's children. And most significant improvement is that system calculates children's relevance using not only ancestor's heritage, but also use meta-data to analyse its relevance score. According to an experiment results the Shark Search is more effective in quality of information retrieved and operation time than its ancestor.

3.3 The Best-First Search Algorithm

The Best-First algorithm focuses on the retrieval of pages which are relevant to a particular given topic. It's an algorithm that uses a score to define which page has a best score. This algorithm uses a rule to select the best page. In most cases it uses artificial intelligence algorithms (Naïve Bayes, Cosine Similarity, Support Vector Machine, k-nearest neighbour algorithm, Gaussian mixture model, etc.) as a classifier to detect the best result. In many articles this algorithm has the best crawling results.

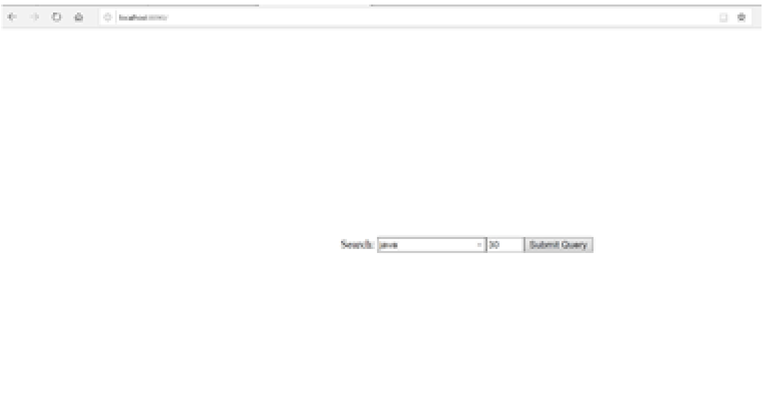
The best-first algorithm pseudo-code

```
Insert in ready queue(seeds)
While true do
    If more links in ready queue then
        link :=dequeue best
        doc :=fetch(link)
        score :=apply rule(doc)
        out links :=extract links(doc)
        save score(out links, score)
    else
        sorted links :=sort(non processed queue)
        insert in ready queue(sorted links)
    end if
end while
```

This algorithm is an algorithm of focused search which explores a graph by expanding the most hopeful node, this node selects according to a specific rule. The Best-first search algorithm principle is in evaluating the promise of node n by a heuristic technique estimation function f(n) which, generally, may depend on the specification of n, the specification of the goal, the information assembled by the search up to that point, and the most important, on any additional knowledge about the problem domain.

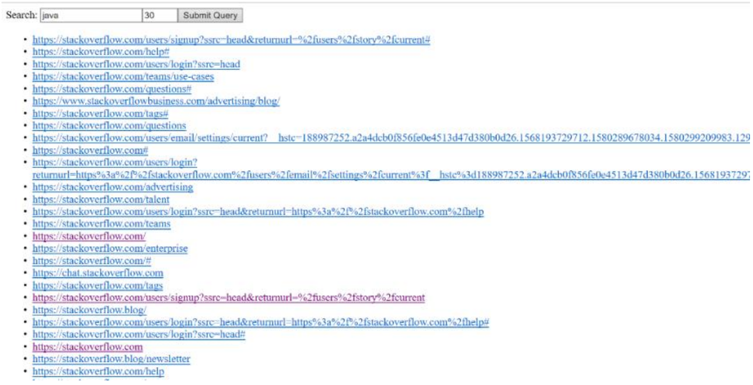
4 Result

4.1 Input



Search: java 30 Submit Query

4.2 Output



Search: java 30 Submit Query

- https://stackoverflow.com/users/signup?ssrc=head&returnurl=%2fusers%2fstory%2fcurrent#
- https://stackoverflow.com/help#
- https://stackoverflow.com/users/login?ssrc=head
- https://stackoverflow.com/teams/use-cases
- https://stackoverflow.com/questions#
- https://www.stackoverflowbusiness.com/advertising/blog/
- https://stackoverflow.com/tags#
- https://stackoverflow.com/questions
- https://stackoverflow.com/users/email/settings/current?__hstc=188987252.a2a4dcb0f856fe0e4513d47d380b0d26.1568193729712.1580289678034.1580299209983.129
- https://stackoverflow.com#
- https://stackoverflow.com/users/login?
returnurl=https%3a%2f%2fstackoverflow.com%2fusers%2femail%2fsettings%2fcurrent%3f__hstc%3d188987252.a2a4dcb0f856fe0e4513d47d380b0d26.1568193729
- https://stackoverflow.com/advertising
- https://stackoverflow.com/talent
- https://stackoverflow.com/users/login?ssrc=head&returnurl=https%3a%2f%2fstackoverflow.com%2fhelp
- https://stackoverflow.com/teams
- https://stackoverflow.com/
- https://stackoverflow.com/enterprise
- https://stackoverflow.com/#
- https://chat.stackoverflow.com
- https://stackoverflow.com/tags
- https://stackoverflow.com/users/signup?ssrc=head&returnurl=%2fusers%2fstory%2fcurrent
- https://stackoverflow.blog/
- https://stackoverflow.com/users/login?ssrc=head&returnurl=https%3a%2f%2fstackoverflow.com%2fhelp#
- https://stackoverflow.com/users/login?ssrc=head#
- https://stackoverflow.com
- https://stackoverflow.blog/newsletter
- https://stackoverflow.com/help

5. Conclusion

The focused web crawler is designed a far away better than the conventional web crawlers which explores the web. The information is sorted out by the relevance of the content through this usage of focused web crawlers. It shows the better performance than the conventional web crawlers. The main limitation of the general crawler is the precision which can be improved by the focused web crawlers. It becomes an expert search of the web. As it is multithreading concept of search which is more precisible than the single thread search concept. The relevance analysis can be carried out for all the URLs and the content can be sorted out.

References

1. P.A. Madhusudan, D. Lambhate Poonam, “Deep web crawling efficiently using dynamic focused web crawler”, International Research Journal of Engineering and Technology, Vol. 4, No. 6, pp. 3303-3306, 2017.
2. H.T. Yani Achsan, W.C. Wibowo, “A Fast Distributed Focused-Web Crawling”, Annals of DAAAM & Proceedings, Vol. 24, No. 1, pp. 492-499, 2013.
3. H. Yan, Z. Gong, N. Zhang, T. Huang, H. Zhong, J. Wei, “Crawling hidden objects with kNN queries”, IEEE Transactions on Knowledge and Data Engineering, Vol. 28, No. 4, pp. 912-924, 2015.
4. M. Gjoka, M. Kurant, C.T. Butts, A. Markopoulou, “Practical recommendations on crawling online social networks”, IEEE Journal on Selected Areas in Communications, Vol. 29, No. 9, pp. 1872-1892, 2011.
5. S.M. Pavalam, S.K. Raja, M. Jawahar, F.K. Akorli, “Web crawler in mobile systems”, International Journal of Machine Learning and Computing, Vol. 2, No. 4, pp. 531-534, 2012.
6. Pal, D.S. Tomar, S.C. Shrivastava, “Effective focused crawling based on content and link structure analysis”, International Journal of Computer Science and Information Security (IJCSIS), Vol. 2, No. 1, pp. 1-5, 2009.
7. N. Jain, P. Rawat, “A study of focused web crawlers for semantic web”, International Journal of Computer Science and Information Technologies, Vol. 4, No. 2, pp. 398-402, 2013.
8. N.S. Kumar, “Improving Efficiency of the Focused Web Crawler by Link Score Calculation”, Journal of Computer Technology & Applications, Vol. 4, No. 1, pp. 16-22, 2019.
9. S. Shang, H. Wu, J. Ma, “An Improved Focused Web Crawler based on Hybrid Similarity”, International Journal of Performability Engineering, Vol 15, No. 10, pp. 2645-2656, 2019.
10. W. Zhu, H. Gao, Z. He, J. Qin, B. Han, “A Hybrid Approach for Recognizing Web Crawlers”, International Conference on Wireless Algorithms, Systems, and Applications, pp. 507-519, 2019.