A Comparative Study Of Various Web Search Algorithms For The Improvement Of Web Crawler

S. Jaiganesh[1] , P. Babu[2], K. NimmatiSatheesh[3]
[1] *Associate Professor*
[2]*Associate Prpfessor,*
[3]*Assitant Professor,*
Department of Computer Applications
PSNA College of Engineering and Technology
Dindigul – 624 622, Tamil Nadu, India.

Abstract--- **The crawler is a vital module of a web search engine. The effectiveness of a crawler directly affects the searching quality of the web search engines. As the crawler interacts with millions of hosts or servers over a period of weeks or months, the issues of validity, flexibility and manageability are of major importance. Also crawler could retrieve some other information, which may be of unimportant to the search from the HTML files as it is parsing them to get the new URLs. In this paper, an attempt has been made to improve the performance of the Web Crawler by comparing certain features of several algorithms such as best-first, breadth-first, pagerank, shark search and Hits. For this, various performance parameters such as precision, recall, accuracy and F-Score are taken into consideration. Based on the output parameters, an analysis is made for the improvement of web crawler towards web searching.**

Keywords: **Web Crawler, best-first, breadth-first, pagerank, shark-search, HITS, precision, recall, accuracy and F- measures**

I. INTRODUCTION

Web crawlers are said to be spiders or robots, they are the programs that automatically retrieve the Web pages when a query is placed in the search engine. Since information on the Web is scattered among billions of pages served by millions of servers around the globe, users who browse the web can follow hyperlinks to access information, virtually moving from one page to the next.

A crawler can visit many sites to collect the information that can be analyzed and mined in a central location[1]. A lot of machine learning approaches are being employed to estimate their significance with respect to the user queries. This is a critical task because it greatly influences the perceived effectiveness of a search engine. Users often look at only a few top hits, making the precision achieved by the ranking algorithm of dominant importance. Many search engines ranked pages principally based on their lexical similarity to the query.

The main objective of this work is to compare five Crawler Algorithms such as PageRank, Breadth-First, Best-First, shark search and HITS to asses their performance measures like precision, recall, accuracy and F-Score.

II. FUNCTIONALITY OF A CRAWLER

A crawler starts from a set of seed pages (URLs) and then uses the links within them to fetch other pages. The links in these pages are, in turn, extracted and the corresponding pages are visited. The process repeats until a sufficient number of pages are visited or some other objective is achieved. In fact, Google founders Sergey Brin and Lawrence Page, in their seminal paper [2], identified the Web crawler as the most sophisticated yet fragile component of a search engine.

Frontier [1]is the one where the Crawler maintains a list of unvisited URLs. The list is initialized with seed URLs which may be provided by the user or another program. In each iteration of its main loop, the crawler picks the next URL from the frontier, fetches the page corresponding to the URL through HTTP, parses the retrieved page to extract its URLs, adds newly discovered URLs to the frontier, and stores the page in a local disk repository. The crawling process may be terminated when a certain number of pages have been crawled.

A crawler is, in essence, a graph search algorithm. The Web can be seen as a large graph with pages as its nodes and hyperlinks as its edges. A crawler starts from a few of the nodes (seeds) and then follows the edges to reach other nodes[3]. The process of fetching a page and extracting the links within it is analogous to expanding a node in graph search. The frontier is the main data structure, which contains the URLs of unvisited pages[8].

Typical crawlers attempt to store the frontier in the main memory for efficiency. Based on the declining

price of memory and the spread of 64-bit processors, quite a large frontier size is feasible. Yet the crawler designer must decide which URLs have low priority and thus get discarded when the frontier is filled up[9]. Note that given some maximum size, the frontier will fill up quickly due to the high fan-out of pages. Even more importantly, the crawler algorithm must specify the order in which new URLs are extracted from the frontier to be visited[4]. These mechanisms determine the graph search algorithm implemented by the crawler.

III. CRAWLER ALGORITHMS

This chapter deals with various crawler algorithms such as PageRank, Breadth-First, Best-First, Shark search and HITS under comparison in this study.

A. Page Rank

PageRank was proposed by Brin and Page [2] as a possible model of user surfing behavior. The PageRank of a page represents the probability that a random surfer (one who follows links randomly from page to page) will be on that page at any given time[5]. A page's score depends recursively upon the scores of the pages that point to it.

```
PageRank (Topic, StartingUrls[], frequency)
        {
                for (i=0;i<=StartingUrl;i++)
                ENQUEUE(Frontier, Link);
        do
        {       if (multiplies(visited,frequency))
                        {       recomputed_scores_pr;
                        }
        }
        while (visited < MaxPages);
                Link = DequeueTopLink(frontier);
                Document=Fetch(Link);
                ScoreSim = Sim(Topic,Doc);
                Enqueue(BufferedPages,Doc,ScoreSim);
                if (BufferedPages >= MaxBuffer}
                        {

                DequeueBottomLinks(BufferedPages)
                        }
                Merge(Frontier,
                ExtractLinks(Doc), ScorePr);
                if (Frontier > MaxBuffer)
                        {
                        DequeueBottomLinks(Frontier)
                        }
        }
```

B. Best-First

Best-First crawlers have been studied by Cho et al. [2] and Hersovici et al. [2]. The basic idea is that given a frontier of URLs, the best URL according to some estimation criterion (Precision, Recall, Accuracy and F-Score) is selected for crawling, using the frontier as a priority queue. In this algorithm, the URL selection process is guided by the lexical similarity between the topic's keywords and the source page of the URL[4]. Thus the similarity between a page *p* and the topic keywords is used to estimate the relevance of all the outgoing links of *p*.

```
BestFirst ( StartingUrls)
{
for (i=0;i<=StartingUrl;i++)
                ENQUEUE(Frontier, url,MaxScore);
        do
        {
                url=Dequeue(Frontier);
                Page=Fetch(Url);
                Score=GetTopicScore(Page);
                Visited=Visited+1;
                Enqueue(Frontier,ExtractLinks(Page),Sc
                ore);
                }
        while (Visited < MaxPages && Frontier != Null);
    }
```

C. Breadth-First

Breadth-First algorithm is the simplest strategy for crawling. It does not utilize heuristics in deciding which URL to visit next. It uses the frontier as a FIFO queue, crawling links in the order in which they are encountered.

```
BreadthFirst ( StartingUrls)
{
for (i=0;i<=StartingUrl;i++)
                ENQUEUE(Frontier, url);
        do
        {       url=Dequeue(Frontier);
                Page=Fetch(Url);
                Visited=Visited+1;
                Enqueue(Frontier,ExtractLinks(Page));
        }
        while (Visited < MaxPages && Frontier != Null);
}
```

D. Shark-search

Shark-Search[6] is a more aggressive version of Fish-Search. In Fish-Search, the crawlers search more extensively in areas of the Web in which relevant pages have been found. At the same time, the algorithm discontinues searches in regions that do not yield relevant pages. Shark-Search offers two main improvements over Fish-Search. It uses a continuous valued function for measuring relevance as opposed to the binary relevance function in Fish-Search. In addition, Shark-Search has a more refined notion of potential scores for the links in the crawl frontier.

```
Shark (topic, startingUrls) {
        Foreach link (startingUrls) {
        Set_depth(link,d);
```

```
        Enqueue(Frontier,links);
        }
        While (visted<MaxPages) {
        Link=DequeueTopLink(Frontier);
        Doc=fetch(Link);
        DocScore=sim(topic,doc);
        If (depth(link)>0) {
           Foreach outlink (extractLink(doc)) {
            Score = (1-r) * neighborhoodScore(outlink)
                      + r * inheritedScore(outlink);
            If DocScore > 0) {
                      setDepth(outlink, d); }
            else  {
                    setdepth(outlinkl, depth(link) – 1);
                    }
                    Enqueue(Frontier,outlink,score);
                    }
                    If (Frontier > MaxBuffer) {
                    dequeueBottomLink(frontier);
                    }
              }
        }
}
```

E. HITS

In the HITS algorithm, the first step is to retrieve the set of results to the search query. The computation is performed only on this result set, not across all Web pages. Authority and hub values are defined in terms of one another in a mutual recursion. An authority value is computed as the sum of the scaled hub values that point to that page. A hub value is the sum of the scaled authority values of the pages it points to. Some implementations also consider the relevance of the linked pages.

```
Hits ( Pages) {
G= SetofPages
For(p=0;p<G;p++)
{
p.auth = 1
p.hub = 1
HubsAndAuthorities(G){
for (i=0;i<k;i++)
 for (p=0;p<G;p++)
  for (q=0;q< p.incomingNeighbors; q++)
p.incomingNeighbors
p.incomingNeighbors
   p.auth += q.hub
   for (p=0;p<G;p++)
     for (page=0;PAGE<R;PAGE++)
       p.hub += r.auth
}          }          }
```

IV. EXPERIMENTAL SETUP

Experiments are conducted using the Lund dataset containing 100 attributes. Three categories of experiments for the performance measures namely :

1. The set of starting URLs **with most links**
2. The set of starting URLs with **the highest topic score**
3. The set of starting URLs with **the lowest topic score**

In the first experiment, the set of starting URLs was the first ten pages with most links

Table I- URLs with most links

	# of pages visited	# of relevant pages visited
Breadth First	100	28.797
Best First	100	33.93
Page Rank	100	35.09
Shark Search	100	27.06
HITS	100	31.23

In the second experiment, the set of starting URLs was the first ten pages with the highest topic score

Table II- URLs with the highest topic score

	# of pages visited	# of relevant pages visited
Breadth First	100	32.52
Best First	100	34.23
Page Rank	100	35.363
Shark Search	100	31.5
HITS	100`	32.6

In the third experiment, the set of starting URLs was the first ten pages with the lowest topic score

Table III- URLs with the lowest topic score

	# of pages visited	# of relevant pages visited
Breadth First	100	32.95
Best First	100	33.78
Page Rank	100	35.18
Shark Search	100	34.2
HITS	100	31.0

V. RESULT AND DISCUSSION

Based on the data set experiments with the following performance measures like precision, recall, accuracy and F-Score are taken into account for their assessment [7].

Precision [10] is defined as the fraction of the documents retrieved that are relevant to the user's information need.

$$\text{Precision} = TP/(TP+FP)$$

Accuracy [10] is defined as the proportion of true results (both TP and TN) in the population

$$\text{Accuracy} = (TP+TN)/(TP+FP+FN+TN)$$

F-Score[10] is defined as the weighted harmonic mean of precision and Recall

$$\text{F-Score} = \frac{2 * (\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

Recall [10] is defined as the fraction of the documents that are relevant to the query that are successfully retrieved.

$$\text{Recall} = \frac{\text{True Positive}}{(\text{True Positive} + \text{False Negative})}$$

Based on the experiment results shown below Table IV and Table V, an analysis is made on the various algorithms.. In the first case(with most links) we observe that precision is better in Page rank, Recall is better in Best First ,Accuracy is better in HITS and F-Score is better in Page Rank. In the second case(Highest Topic Score), we found that, precision is better in Page rank, Recall is better in Best First, Accuracy is better in Shark search and F-Score is better in Page Rank. From the third case(Lowest Topic Score), it is observed that precision is better in Page rank, Recall is better in Best First, Accuracy is better in Shark search and F-Score is better in Page Rank.
Graphical view of an analysis is shown in fig 1, 2, 3, 4. It is also realized that there is a strong need to further probe into this to develop a robust crawler algorithm for better performance in all the above mentioned counts.

Table IV (a)

	Breadth First			
	Precision	Recall	Accuracy	F-Score
Most Links	28.8	40.4	43.12	33.61
Highest Topic Score	32.5	46	47.49	38.12
Lowest Topic Score	33	46	47.51	38.41
Average	31.43	44.3	46.04	36.71

Table IV (b)

	Best First			
	Precision	Recall	Accuracy	F-Score
Most Links	33.93	47	47.7	39.41
Highest Topic Score	34.23	47.7	48.34	39.85
Lowest Topic Score	33.78	47.1	47.03	39.34
Average	33.98	47.3	47.69	39.54

Table IV (c)

	Shark Search			
	Precision	Recall	Accuracy	F-Score
Most Links	27.05	41.23	44.25	32.66
Highest Topic Score	31.5	44.33	44.98	36.83
Lowest Topic Score	34.28	46.82	47.67	39.58
Average	30.94	44.13	45.63	36.36

Table IV (d)

	Page Rank			
	Precision	Recall	Accuracy	F-Score
Most Links	35.09	44.63	45.78	39.29
Highest Topic Score	35.363	45.37	46.41	39.75
Lowest Topic Score	35.181	46.21	47.12	39.95
Average	35.21	45.4	46.44	39.66

Table IV (e)

	HITS			
	Precision	Recall	Accuracy	F-Score
Most Links	31.23	46.77	47.85	37.45
Highest Topic Score	32.6	46.47	47.53	38.32
Lowest Topic Score	30.78	45.63	46.86	36.76
Average	31.54	46.29	47.41	37.51

Graphical View of the Performance Measures

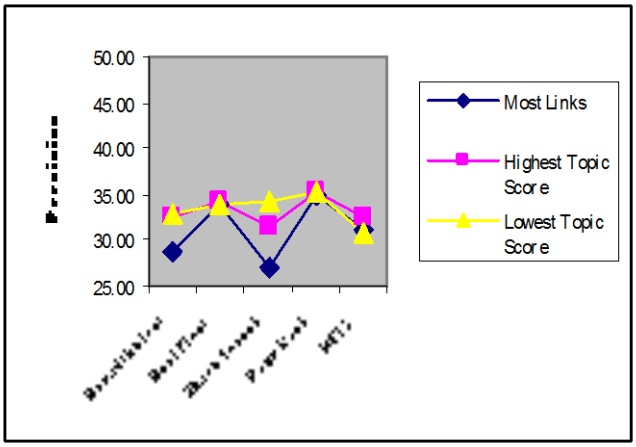

Figure 1: Performance measure by Precision

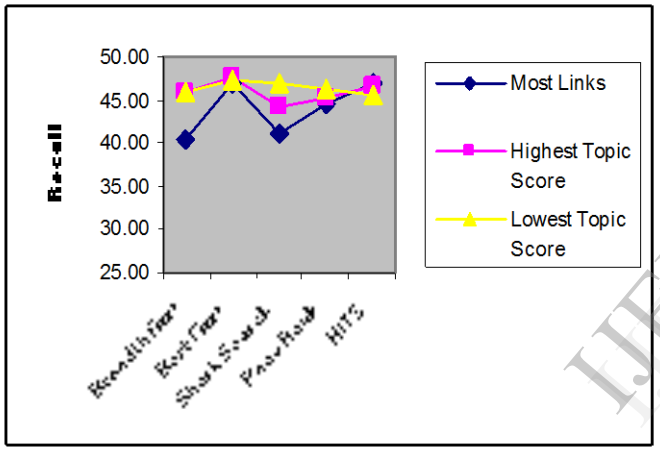

Figure 2: Performance measure by Recall

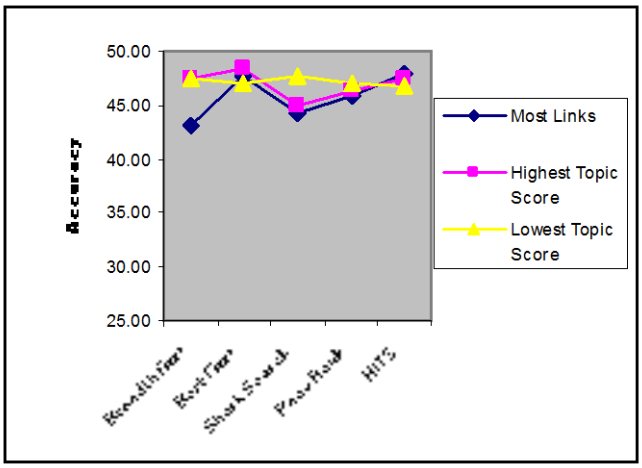

Figure 3: Performance measure by Accuracy

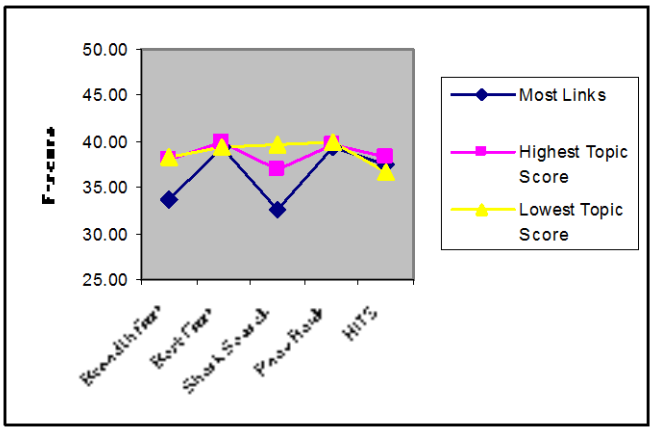

Figure 4: Performance measure by F-score

VI. CONCLUSIONS

In this paper, an attempt has been made to improve the efficiency of the Web Crawler for improving the web searching with the help of comparing certain features of several algorithms such as best-first, breadth-first, pagerank, shark search and HITS. For this, various performance parameters such as precision, recall, accuracy and F- score are taken into consideration. Based on the output parameters, it is observed that pagerank algorithm outperforms over other algorithms by various performance measures. This result leads to the conclusion that the pagerank algorithm improves the performance of web crawler for quick information retrieval. Any efficient web search algroritm for web crawler should focus on precision and F-score for the betterment of web search.

VII. REFERENCES

[1]. Aggarwal, C., Al-Garawi, F., and Yu, P. 2001. Intelligent crawling on the World Wide Web with arbitrary predicates. In Proceedings of the 10th International World Wide Web Conference. 96–105.

[2]. Brin, S., Page, L., Motwami, R., Winograd, T. The pagerank citation ranking: bringing order to the web. Technical report, Standford digital library technologies project, Stanford University, Stanford, CA, USA, 1998

[3]. Chakrabarti, S., Van Den Berg, M., and Dom, B. 1999. Focused crawling: A new approach to topicspecific Web resource discovery. Comput. Netw. 31, 11–16, 1623–1640.

[4]. Cho, J.,Garcia-Molina, H., and Page, L. 1998. Efficient crawling through URL ordering. Comput. Netw. 30, 1–7, 161–172.

[5]. Filippo Menczer , Indiana University ,Gautam Pant, University of Utah and Padmini Srinivasan, University of Iowa , November 2004. Topical Web Crawlers: Evaluating Adaptive Algorithms. ACM Transactions on Internet Technology, Vol. 4, No. 4,

[6]. Pau Vall_es Fradera and Ricard Gavald_a Mestre Universitat Polit_ecnica de Catalunya, Jan ,2009 .

Personalizing web search and crawling from click stream data.
[7]. Rafael Romero Trujillo, 2006 - Simulation tool to study focused web crawling strategies
[8]. Roger P. Menezes and Prof. Soumen Chakrabarti,IIT,Bombay,2004. Crawling the Web at Desktop Scales
[9]. K.Shchekotykhin , G.Friedrich , University Klagenfurt, Austria , Nov 2009. xCrawl : a high – recall crawling method for Web mining. Springer 2009.
[10]. Accuracy, Precision, Recall and F-Score - Wikipedia, the free encyclopedia.