*International Journal of Scientific Engineering and Applied Science (IJSEAS) – Volume-2, Issue-3,March 2016*
*ISSN: 2395-3470*
*www.ijseas.com*

# A Survey On Various Kinds Of Web Crawlers And Intelligent Crawler

**Mridul B. Sahu[1], Prof. Samiksha Bharne[2]**
[1] M.Tech Student, Dept. Of Computer Science And Engineering, (BIT), Ballarpur, India
[2] Professor, Dept. Of Computer Science And Engineering, (BIT), Ballarpur, India

**Abstract**
This Paper presents a study of web crawlers used in search engines. Nowadays finding meaningful information among the billions of information resources on the World Wide Web is a difficult task due to growing popularity of the Internet. This paper basically focuses on study of the various kinds of web crawler for finding the relevant information from World Wide Web. A web crawler is defined as an automated program that methodically scans through Internet pages and downloads any page that can be reached via links. A performance analysis of performance of intelligent crawler is presented and data mining algorithms are compared on the basis of crawlers usability.

Keywords: Web Crawler, Data Mining, K-Means, machine learning, SVM, C4.5.

## 1. Introduction

The internet has becoming the largest unstructured database for accessing information over the documents. [8] It is well recognized that the information technology has a profound effect on the conduct of the business, and the Internet has become the largest marketplace in the world. Innovative business professionals have realized the commercial applications of the Internet for their customers and strategic partners. [2] With the rapid growth of electronic text from the complex the WWW, more and more knowledge you need is included. But, the massive amount of text also takes so much trouble to people to find useful information. For example, the standard Web search engines have low precision, since typically some relevant Web pages are returned mixed with a large number of irrelevant pages, which is mainly due to the situation that the topic-specific features may occur in different contexts. So, one appropriate way of organizing this overwhelming amount of documents is necessary.[1] The World Wide Web is an architectural framework for accessing linked documents spread out over millions of machines all over the Internet.

The visible Web with its estimated size of at least 20billion pages, offers a challenging information retrieval problem. Even with increasing hardware and bandwidth resources at their disposal, search engines cannot keep up with the growth of the Web [3]. The retrieval challenges further compounded by the fact that Web pages also change frequently. Thus, despite the attempts of search engines to index the whole Web, it is expected that the subspace eluding indexing will continue to grow. Therefore, collecting domain-specific documents from the Web has been considered one of the most important strategies to build digital libraries capable of covering the whole we band take benefit from the large amount of information and resources that can be useful.

## 2. Background

The World Wide Web provides a vast source of information of almost all type. However this information is often scattered among many web servers and hosts, using many different formats. We all want that we should have the best possible search in less time. For any crawler there are two issues that it should consider. First, The crawler should have the capability to plan, i.e., a plan to decide which pages to download next. Second, It needs to have a highly optimized and robust system architecture so that it can download a large number of pages per second even against crashes, manageable, and considerate of resources and web servers.

### 2.1 Web Crawler

A Web crawler is a program that automatically traverses the Web's hyperlink structure and downloads each linked page to a local storage. Crawling is often the first step of Web mining or in building a Web search engine. Although conceptually easy, building a practical crawler is by no means simple. Due to efficiency and many other concerns, it involves a great deal of engineering. There are two types of crawlers: universal crawlers and topic crawlers [7].

A universal crawler downloads all pages irrespective of their contents, while a topic crawler downloads only pages of certain topics. The difficulty in topic crawling is how to recognize such pages. Web crawler is an Internet that systematically browses the World Wide Web, typically for the purpose of Web indexing. It also called as Web spider, an ant, an automatic indexer, Web Scutter.

Web search engines and some other sites use Web crawling or spidering software to update their web content or indexes of others sites' web content. Web crawlers can copy all the pages they visit for later processing by a search engine that indexes the downloaded pages so that users can search them much more quickly [19]. The figure 1 shows that the design of a web crawler.

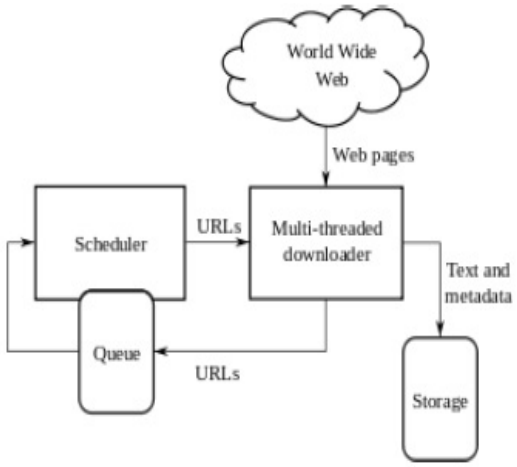


Figure 1. Design Of Web Crawler

### 2.2. Working Of Web Crawler

A Search Engine Spider (also known as a crawler, Robot, Search Bot or simply a Bot) is a program that most search engines use to find what's new on the Internet. Google's web crawler is known as GoogleBot. There are many types of web spiders in use, but for now, we're only interested in the Bots that actually "crawls" the web and collects documents to build a searchable index for the different search engines. The program starts at a website and follows every hyperlink on each page.

So we can say that everything on the web will eventually be found and spidered, as the so called "spider" crawls from one website to another. Search engines may run thousands of instances of their web crawling programs simultaneously, on multiple servers. When a web crawler visits one of your pages, it loads the site's content into a database. Once a page has been fetched, the text of your page is loaded into the search engine's index, which is a massive database of words, and where they occur on different web pages. All of this may sound too technical for most people, but it's important to understand the basics of how a Web Crawler works.

So, there are basically three steps that are involved in the web crawling procedure. First, the search bot starts by crawling pages of your site. Then it continues indexing the words and content of the site, and finally it visit links (web page addresses or URLs) that are found in your site. When the spider doesn't find a page, it will eventually be deleted from the index. However, some of the spiders will check again for a second time to verify that the page really is offline.

The first thing a spider is supposed to do when it visits your website is look for a file called "robots.txt". This file contains instructions for the spider on which parts of the website to index, and which parts to ignore. The only way to control what a spider sees on your site is by using a robots.txt file. All spiders are supposed to follow some rules, and the major search engines do follow these rules for the most part. Fortunately, the major search engines like Google or Bing are finally working together on standards.

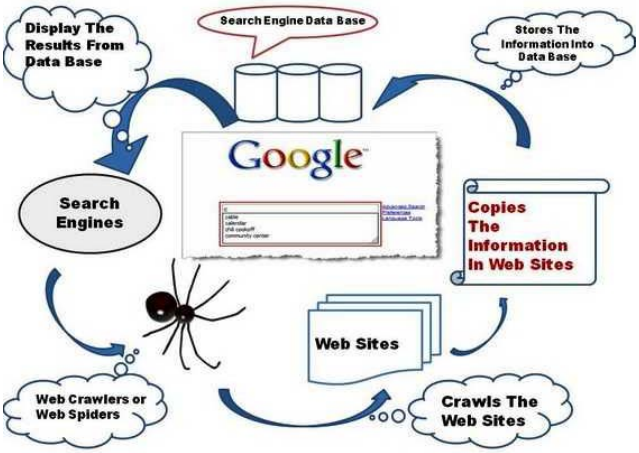


Figure 2. Working Of Web Crawler

Following is the process by which Web Crawler's work.

- Download the Web page.
- Parse through the downloaded page and retrieve all the links.
- For each link retrieved, repeat the process.

**2.3 Methods Of Web Crawling**

A. Distributed Crawling

Indexing the web is a challenge due to its growing and dynamic nature. As the size of the Web is growing it has become imperative to parallelize the crawling process in order to finish downloading the pages in a reasonable amount of time. A single crawling process is insufficient for large – scale engines that need to fetch large amounts of data rapidly. When a single centralized crawler is used all the fetched data passes through a single physical link. Distributing the crawling activity via multiple processes can help build a scalable, easily configurable system, which is fault tolerant system. Splitting the load decreases hardware requirements and at the same time increases the overall download speed and reliability. Each task is performed in a fully distributed fashion, that is, no central coordinator exists [4].

B. Focused Crawling

A general purpose Web crawler gathers as many pages as it can from a particular set of URL's, Where as a focused crawler is designed to only gather documents on a specific topic, thus reducing the amount of network traffic and download. The goal of the focused crawler is to selectively seek out pages that are relevant to a pre-defined set of topics.

The topics are specified not using keywords, but using exemplary documents. Rather than collecting and indexing all accessible web documents to be able to answer all possible adhoc queries, a focused crawler analyzes its crawl boundary to find the links that are likely to be most relevant for the crawl, and avoids irrelevant regions of the web. This leads to significant savings in hardware and network resources, and helps keep the crawl more up-to-date..

The Focused crawlers have two main components that are

used to guide the process of crawling: Classifier and Distiller [4]. A classifier that is used to calculate the relevance of the document with that of the focused topic that is being searched for i.e. the classification of relevant and non relevant web pages is done in this module of the focused crawler. A distiller that is used to search for the efficient access points that leads to a large number of relevant documents by using lesser number of links i.e. finds the good access nodes from this complete web graph.

The structure of the focused crawler was first given by [4].

The general structure of the focused crawler is as shown in Fig 3. For the focused crawlers the complete web is not of interest, but it is interested in only a specific domain. The focused crawler loads the page and extracts the links of that page. These links are then stored in the crawl frontier. Then by some relevance calculation it is decided which page to move next in the queue of the URLs for the pages stored in the frontier. Now-a-days the focused crawlers are using different methods to check for the relevancy.

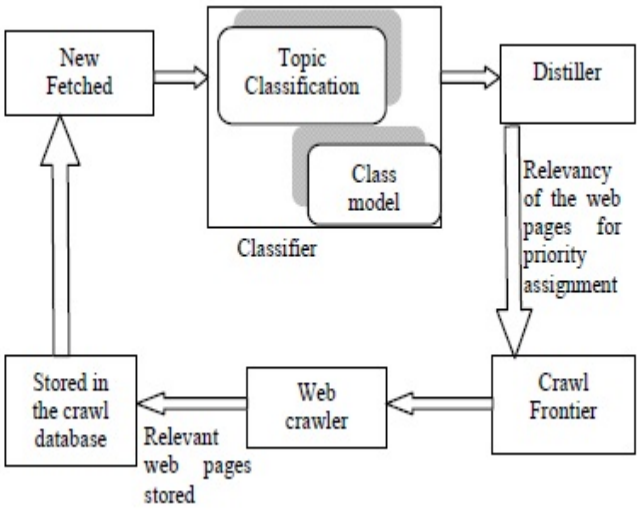


Figure 3. The Structure Of Focused Crawler

## 3.Intelligent Crawler

While introducing intelligence, two major approaches dominate the decisions made by the crawler. First approach decides its crawling strategy by looking for the next best link amongst all links it can travel. This approach is popularly known as supervised learning whereas the second approach computes the benefit of traveling to all links and ranks them, which is used to decide the next link. Both the approaches may sound similar because in human brain a hybrid approach of both the algorithms is believed to aid decision making. But if noticed carefully, supervised learning requires training data to help it decide the next best step, while unsupervised learning doesn't. Collecting and making the program understand sufficient amount of training data may be a difficult task.

The crawling strategy used can be classified as vertical topical strategy. The crawler follows a focused thematic approach, and the pages which it fetches will be guided by the interest of the user and the introduced intelligence.

To apply intelligence heavy processing of memory-resident data is required. In addition to this heavy processing, topical crawler is a multi-threaded process, thus use of multi-core processing is an implicit need. Optimization of hardware also comes into picture.

## 4. Performance Metrics

Crawlers start crawling from a seed page. The seed page plays a critical role in guiding the crawler and to find path leading to target page. By taping performance parameters of a crawler to reach the target page can be optimized . Consider a crawler*l*, its output for a given topic can be written as a temporal sequence give as where*ui*is URL *of i*th page crawled and $M$ is the maximum number of pages crawled. We should also be able to measure performance of the crawler, thus we need to define a function $f$ that maps the sequence $Si$ to a sequence where $ri$is a scalar quantity that represents the relevance of the $i$th page crawled to the topic. The sequence $Rl$will help in summarizing the results at various points during the path of crawl.

### 4.1 Harvest Rate

This rate calculates the rate of crawled pages that form relevance linking to the topic along with all the pages that have been crawled. We vastly depend on classifiers to make this type of conclusion. The classifiers used as a part of data mining intelligence can perform such critical decisions.

The Harvest rate after first $t$pages is computed using formula:

$$H(t) = \frac{1}{t}\sum_{i=1}^{t} r_i$$

Here $ri$ is binary relevance score of page $i$. This score is provided by the classifier. The score is subject to changes depending on the strategy used by the classification software.

General representation of average harvest rate is $h$c,p

Where $c$ is the classifier used and $p$ is the number of pages crawled. This value ranges from 0-1 where 0 being the worst case scenario performance and 1 being the best case performance.

To measures the performance of crawler basically three algorithms are used namely:

1. SVM(Support Vector Machines)
2. C4.5 (Statistical Classifier Algorithm)
3.K-Mean Algorithm

## 4. Conclusions

With the study and analysis of web crawling methods, techniques in the web. To extract information from the web, crawling techniques are discussed in this paper. Efficiency improvements made by data mining algorithms included in the study on crawlers. We observed the learning procedure required by a crawler to make intelligent decisions while selecting its strategy.

## Acknowledgments

I would like to extend my gratitude to many people who helped me to bring this paper fruition. First I would like to thank Prof. Samiksha Bharne. I am so deeply grateful for her help, professionalism, and valuable guidance throughout this paper. I would also like to thank to my friends and colleague. This accomplishment would not have been possible without them. Thank you.

## References

[1] Lu LIU, Tao PENG "Clustering-based topical Web crawling using CFu-tree guided by link-context" in Higher Education Press and Springer-Verlag Berlin Heidelberg 2014.

[2] Abhiraj Darshakar,"Crawlers intelligence with Machine Learning and Data Mining integration", International Conference on Pervasive Computing (ICPC),2015.

[3] Hai Dong, Farookh Khadeer Hussain, and Elizabeth Chang "Ontology-Learning-Based Focused Crawling for Online Service Advertising Information Discovery and Classification" in Springer-Verlag Berlin Heidelberg 2012

[4] S. Lawrence and C. L. Giles, .Searching the World Wide Web,"Science, vol. 280, no. 5360, pp. 98-100,1998

[5] S. Chakrabarti, M. Berg, and B. Dom, "Focused Crawling: A New Approach to Topic-specific Web Resource Discovery," *Journal of Computer Network*, vol. 31, no. 11-16, pp. 1623-1640, 1999.

[6] R.Eswaramoorthy, M.Jayanthi "A Survey on Detection of Mining Service Information Discovery Using SASF Crawler" in International Journal of Innovative Research in Computer and Communication Engineering Vol. 2, Issue 10, October 2014 .

[7] SalvatorRugier. Efficient C4.5 ACM/IEEE joint conference on Data mining.

[8] S.Balan1, Dr.P.Ponmuthuramalingam2," A Study on Semantic Web Mining And Web Crawler", International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 2 Issue 9 Sept., 2013 Page No. 2659-2662

[9] Risha Gaur1, Dilip kumar Sharma2," Focused Crawling with Ontology using Semi- Automatic Tagging for Relevancy", 978-1-4799-5174/14/$31.00 ©2014 IEEE.

**First Author**: Mridul B. Sahu has obtained Bachelor degree from Rashtrasant Tukadaji Maharaj Nagpur University, Nagpur. Presently M.Tech student at Ballarpur Institute Of Technology, Bamni, Ballarpur, Chandrapur, Gondwana University, Maharahstra, India.

**Second Author** Samiksha Bharne is Assistant Professor at Ballarpur Institute Of Technology, Bamni, Ballarpur, Chandrapur Gondwana University, Maharahstra, India