International Journal of Control Theory and Applications

ISSN : 0974-5572

© International Science Press

Volume 9 • Number 45 • 2016

Advances in Web Crawlers

Sawroop Kaur[a] and G. Geetha[b]

[a]Department of computer science and engineering, Lovely professional university Phagwara, India

[b]Department of Research and Development, Lovely professional university, Phagwara, India

E-mail: gitaskumar@yahoo.com

Abstract : Dynamic nature of Internet requires robust and flexible Information mining systems. Search engines are commanding tools to get helpful information that is scattered on the Web. Search engines these days are praise worthy for their speed and information retrieval. Still most prominent search engines give not very useful result pages as answer to the user query. When information for a particular topic set is required focused crawling is used. What follows here is a review of existing challenges in information retrieval and focused crawling. This work is an attempt to investigate different web crawlers and their types. On the basis of literature reviewed, different techniques and methodologies adopted in focused crawling are explored. This paper is wrapped up with a foretell of what we assume are the quick next steps in focused crawling

Keywords: *Information retrieval, Search engines, Focused crawler, Web crawlers.*

1. INTRODUCTION

Information retrieval system (IR) is characterized as the discipline of discovering significant records instead of straight forward matches to examples in a question. The basic viewpoint of IR lies in the relevance of results surveyed with respect to the needed information. The task of information retrieval is discovering records portrayed by an unstructured nature that fulfil a data need from vast accumulations, put away on PC's. Data is structured in case of Extensible Markup Language (XML) and relational databases which retrieves data by satisfying some conditions expressed through query language. Besides, returned results are only exact matches. A web crawler must discover and channel the most pertinent data coordinating a user query, and afterward exhibit that data in a way that makes the data most promptly satisfactory to the client [5].

Challenges of web information retrieval include :

1. Maintaining the freshness and completeness of the index, excluding noisy and incomplete data from search results. User feedback, duplicate detection and improving the query language [6].
2. Processing of large data in less time [33].
3. Providing automation and reducing human help [33, 53].

1.1. Search Engine

The term web search engine can be referred to as some sort of inquiry record, a tremendous database with data from individual web destinations. Modern search engines are asynchronous, centralized, massive in size and dynamic in nature [66].

1.2. Classification of Search Engines Search

1.2.1. Primary Search Engines

A primary search engines like Google, Yahoo and MSN try to index all possible web sites on the web. Each primary search engine is different from others and has its own crawler to list the web pages [42, 48]. Google is distributed in nature and use multiple computers for crawling web pages [87]. It is crawler based search engine that automatically create their listings to crawl the web, people search through the listing gathered [50]. Google's crawler is Google bot [14].

1.2.2. Secondary Search Engines

Unlike primary search engines, goal of secondary search engines cover smaller audiences. These are helpful for searches that are narrowly focused. Secondary search engines are different in the way they rank search results using keywords, back links and Meta tags etc [46].

1.2.3. Focused or Topical Search Engines

Topical web search engines are engaged towards a particular topic, like sports or medicine. CitySearch and MusicSearch are examples of focused search engines. The motive behind using focused crawler is to search web pages that are suitable to a pre-considered arrangement of topics. It spiders only pertinent areas of the world wide web and save resources [44]. Whenever update in web page is taken place, these search engines sooner or later find these updates, and these changes affect the ranking of page. Due to dynamic nature of WWW accessing the information is difficult. Search engines depend on their crawlers to search the web pages. Building a crawler requires user's actions of browsing to be imitated [72]. Web spider collect web pages automatically by exploring the hyperlinks in the web pages. Web is growing at rapid growth so general purpose crawler faces many challenges. One of the major challenges is how to get information in precise and timely manner, to crawl all the web pages and index them. Powerful search engine such as Google has become synonym for search, it crawls and index millions of web pages every day. In 1998 when first time Google was indexed, it indexed 26 million pages. The size of World Wide Web (WWW) that is indexed is at least 4.75 billion pages. Estimated size of Google's index is more than 50 million pages. Keeping the index fresh is the major problem for search engine. Even Google being a leader of search market cannot crawl every web page. For 26 million pages to get downloaded it took around 9 days. After the system start running smoothly it works faster, in 63 hours it downloads 11 million pages [14.]

1.3. Challenges of Web Crawlers

Web crawlers are imperative elements which are utilized to accumulate the corpus of website pages indexed by the search engine. Execution of a web crawler is simple, however because of huge size of web, designing a crawler pose significant building difficulties. Based on the literature reviewed following are the challenges for web crawlers.

1. It is not possible for search engine to retrieve large collection of pages. Due to ever-increasing size of WWW, crawlers have to limit their scope. Due to resource restrictions, it is not possible to download all the pages so pruning techniques are used. Important pages that are relevant to a topic may miss due to narrow scope. Heterogeneity and inconstancy are other challenges [75, 96].

2. Search engines are automatic in nature so they can be easily fooled by any web page to make it rank higher [64].
3. Frequent data refresh to frequently response to user query with fresh data [16, 17, 64].
4. Search engines download hundreds of millions of web pages, indexing these pages is still a challenge for search engines [6].
5. In case of personalized requirements, existing search engines are not capable to meet up all user requirements. Due to scarcity of complete crawl of the retrieved results, it's hard to locate information due to a huge quantity of results [81].
6. Scrutinizing specific topic web pages from a huge number of search results is not easy task for any web user [10].
7. Query processing is one other challenge. Indexing of large number of pages is difficult; deployment cycles, compactness, and the update speed of index are the performance indicators for indexing [4, 6].
8. Decision for the starting URL's for frontier is another challenge [75].
9. In any case if changes are made web page, to decide whether an updated content is relevant to the search topic or not is another considerable challenge [82].
10. Selection of good queries to collect data record from web databases [101].
11. Due to dynamic nature and requirement of retrieving records in very less time crawlers are needed to be distributed in nature [77].

In this paper we present review on generic crawlers, focused crawlers, parallel and distributed crawlers by defining the pros and cons of each of them with special emphasis on focused web crawlers.

2. RELATED WORK

The term focused crawlers was coined by Soumen Chakrabarti. A new system for hypertext resource discovery called focused crawler whose aim was to find web pages relevant to set of topics that were pre defined. Exemplary documents were used for topic specification. The classifier evaluates the relevance of the hypertext document, and distiller identifies hypertext nodes. If a page found at the present level is not relevant, yet a few related pages are at a distance from it, the chance of including those pages will be effectively low [19].

To uncover data behind a particular HTML form, and to process outcomes of pages returned by a form submission are discussed under this technique. Error detection technique with new boundary sentence separator tag was used to adapt the copy detection process for set of records. This special tag is inserted during duplication detection. It is depth oriented crawler for content extraction. After getting data from the deep web query system hash value is calculated for every result and then duplication is removed. Inside result page detection of forms is not taken into account. Another weakness in system is computing of hash values for each sentence that leads to high resource requirements [56].

Crash tolerant crawler crawls several web pages per second. In case of crash, it recrawls by using a checkpoint. The crawling application decides which page is to be requested on the basis of current state plus pages that are crawled earlier and issue a stream of URL's to the crawling system. Then system downloads the web pages and sends the URL's for scrutiny and storage of crawling application [92].

Hidden Web Exposer is a significant technique for spidering the hidden web. Layout based information extraction technique is used. Depth orientation for content extraction and text similarity to match fields and attributes of domain is being used. This approach ignores the forms which have less than three attributes and significant human input is needed so performance is highly dependable on the quality of input data [82].

Hierarchical database sampling based algorithm receives information from the search engines to get document frequencies for some words. Information of document frequency is a part of the database description to assemble a hierarchical structure for databases. Hierarchical structure with extra information retrieved better retrieval performance but did not clearly weigh up resource selection performance. This approach does not consider flat query classification [44].

Myspiders in is a multi agent based crawler deployed on a public web server intended to browse adaptively on behalf of the user. It is appropriate as search refinement tool, which aid web user from manually browsing when the significant fresh pages might not have been listed. It is beneficial as personalized background search and notification tool [76].

A graph-oriented knowledge representation for focused crawling has incorporated a definition of the knowledge structure, its lexical representation and relevance computation strategies. This approach is a combination of focused crawling and concept of ontologies. Entity reference, background knowledge and score summarization are collectively used for computing final relevance score. Ontologies can be combined with context graphs to produce even better results [31].

In probabilistic model for intelligent web crawlers, numbers of outgoing URL's of web page are modelled by identical and independent distributed random variables. Markov chain model dynamically locate a related page on web. Defining probability as a parameter for crawler confidence deals with distribution of search query. Breadth-first search is used to find web pages that are relevant to a user query. Impact of computational capability and bandwidth is ignored [60].

Neuro dynamic programming based focused crawler named as Yet Another Focused Crawler learn function value to reach relevant web page. For this, combination of temporal difference learning algorithm with neural network is used. Neural network is fully connected with one layer of sigmoid hidden units. Convergence of td (λ) is demonstrated for linear networks and linearly independent input pattern set only. The algorithm may not even converge to locally optimal solution [85].

Focused crawler named as wHunter is based on incremental multi strategy learning. wHunter implied both support vector machines (SVM) and naïve bayes classifier. The use of both SVM and Naïve bayes leads to high efficiency using online incremental learning [43].

For topical crawling in business intelligence, crawlers are divided into breadth first search, naïve best first crawler, data object model based crawler and hub seeking crawler. Hub seeking crawler outperforms breadth first search crawler by harvesting more relevant pages than any other crawler [67].

Study has explored a design of parallel spiders as a multi-agent system to investigate redundant pages to minimize cost. With distributed environment taken into account crawler agents harmonize their information collection actions, to avoid redundancy caused by parallelization. Multi-agent platform is used where results uncovers its cooperativeness to enhance the search performance within suitable efficiency cost [61.]

Heritrix aimed to provide a good general platform for web crawling. This crawler is world's first open source, extensible, archival-quality web crawler. HTTP Get request is used to download only regular pages, excluding forms that can be accessible through Web forms. From one site to another site there is no variation in crawling process, so system face resource loss due to its blind behaviour [93].

Igloog is a grid based crawler. Information service is responsible for gathering the information about resource allocation of URL's to generate good load parity for crawlers. Igloog deals with web page downloaded on large-scale. Multiple crawlers can run on one node. The bandwidth and computing capability of node decide the number of crawlers included. The semantic vectors of URL's are computed with Latent Semantic Indexing (LSI). IglooG when scaled to the whole web can retrieve tens of millions of web documents [58].

LS crawler performs web page search on the semantic basis. Relevancy of documents is determined prior to downloading so as to generate a depository of the documents which are most relevant. Measure of relationship for keyword search in the particular domain is hypertext, by considering ontology. Priority assigned to the URL and the pair of URL's depends on the relevancy of the terms. The search is based on the matching keyword with the each term in the hyperlink [104].

A domain definition based query method in has overcome the problems of session management in conventional crawlers by using session object to restore the execution environment. HTML forms and anchor text both are used when reaching new page but systems is not fully automatic and even automatic updating is not available. This system is not scalable [4].

Web Miner use hierarchical agglomerative clustering to put URL's into clusters. Topic-wise repository is formed by taking into consideration the content of form and it help in creating ontology. Qualities of centralized and decentralized systems are combined using super peer architecture. With distributed hash table functionality nodes find files based on their key. Each node handles a portion of the hash space. Location-deterministic distributed lookup routing technique is used. There is no need of global knowledge requirement; crawler has properties of faster locating and load balancing [99].

Depth oriented crawler for content extraction is used in order to evaluate the query templates. Formativeness test is used for evaluation. This crawler efficiently navigates the search space of potential input combinations. Weakness of this system is that there is no concern to the efficiency of deep web crawling [63].

HAWK crawler is implemented using user-defined relevance formula and Shark-search to predict relevance score. Search based on content structure and link structure is combined in this architecture. Page relevance is improved by using content of page while link structure improves coverage. Relevance score is calculated by matching the crawling page with topic of search. If the calculated relevance score is higher than threshold value then entire child link is extracted. After downloading all relevant documents, similarity to topic and anchor text is computed. If similarity to topic is more than threshold value then anchor text context is computed [20].

Irobot uses both breadth first and depth first search for sitemap reconstruction (offline), and path selection for traversal and online crawling. From the list-of-threads, iRobot can find out those pages which are from one thread. Relevant pages are constantly collected in repository for indexing of web pages and data mining tasks. It intelligently skips invalid and redundant pages, to keep only informative ones also decrease the ratios of invalidation and duplication. More than one path is not allowed due to tree-like traversal path. Page structure is changed if its URL location is changed. Dealing with the frequent thread updating in forum is not taken place [15].

Ajax crawler is based on finite-state models of Ajax based applications. It use breadth first search to crawl the Ajax application. This approach is not feedback directed. Caching of the JavaScript function calls helps reducing cost of the communication [30].

Study addresses an issue of choosing webpage for vertical web crawling and based on the URL rules, two-step URL choosing strategy is proposed. Breadth-first algorithm is used to retrieve as many pages as algorithm can from every topic-specific site. Topic-specific web pages are named as 'positive pages', while others are negative pages. Page classifier classifies retrieved pages on the basis of positive and negative pages to spot a page as a topic-specific content page or a topic-specific catalogue page. URL polymerizer group the URL clusters with each URL class include similar URL's [108].

Suite of query selection techniques select good queries in order to speedily harvest data records from Web databases. Query based crawling for databases are modelled. This method is tested on controlled local servers and real web structured source [102].

A focused crawler with ontology-supported website models for information agents is aim to utilizing ontology for website models. Domain ontology is centre technology for searching web resources that are both user and domain oriented. Website models that support ontology offer a semantic level solution to provide rapid, specific, and stable query results [103].

Vertical crawler for Internet forum and two modes for vertical crawling have been discussed:

1. Crawlers that run in Directional Mode.
2. The information is structured in templates file.

Downloaded information is organized according to semantics. Post information table store crawled information of each post. This crawler downloads post pages after list pages and analysis is saved into post information tables. Re-crawling is done periodically, when there is change in number of replies and views, updates takes place in information tables. This crawler is also appropriate for news and blog sites [35].

AKSHR is depth oriented crawler for content extraction. Interface mapper is used to unify query interfaces for domain calculation of revisit frequency. Duplication is avoided using mapping knowledge base. Mapping knowledge base minimizes the mapping effort. Indexing technique is not specified and performance is defined for crawling only. Efficiency of merging procedures and schema matching over variety of query interfaces has not been quantified [12].

The Intelligent crawler solves the problem of finding relevant pages by downloading pages before crawling begins. Semantic content of the URL is based on the domain dependent ontology, which in turn support the metric that is used for prioritizing the URL queue. For estimating relevancy of the links in page, knowledge path plays important role. To filter the URL's in the frontier, semantic nature of URL is explored. This crawler doesn't need any relevance feedback. Crawling time is reduced as compared to other crawlers. Crawling rate is increased when ontology is introduced [25].

The link score of web page is based on average relevancy score of home page, and division score is calculated. Link score on comparing with threshold value decide relevancy of the web pages. Crawler fetches those links having greater value of threshold. Using porter stemming algorithm words are stemmed and top 10 words are fetched. Efficiency of crawler is combination of both maximum numbers of pages retrieved and less time to crawl. Crawler proposed has only concentrated on maximum retrieval of pages [40].

Focused parallel architecture of a web crawler has used combination of click stream analysis and text analysis approaches for collection of URL's in frontier. This architecture is not central coordinator based. Duration of all visits per web page corresponds to calculating click stream that direct to find high quality web page. C-procs of crawler correspond with each other to balance their results during finding of vital pages. One log file is accessed to compute importance of web page. Page rank and back link metrics computation can be used to download pages [89].

URL distance based focused crawler algorithm is based on an experimental crawler and a focused crawler. To calculate the relevancy, vector space model is used. Relevancy is considered between seed page and child page. Child page links are extracted using link extraction tool. Experimental crawler fetch seed page, child page and out links of the seed page. Relevancy is calculated between seed page and all of its child pages to obtain topic specific pages from internet. Distance score is calculated between visited URL's and each URL which is to be fetched [41].

Inforce crawler extract information while web page crawling. Noise removal from web pages is done using information extraction. Crawling is batch oriented and avoids duplicity of data. Topics, comments on topics and their relationships are ordered in a physical file for data analysis from clean data source. Automatic pattern mapping is used for rules for descriptions of the mapping, and extensible style sheet language transformations for rule evaluation and implement extraction. Use of learning technology can enhance mapping rules [105].

Domain Specific Hidden Web Crawler automates the process of downloading the search interfaces and semantic mappings. The process of searching, viewing, and submission is automated. Search forms are submitted as per the scrutiny of the response pages. Weakness of this system is mass storage is required for hidden web pages and no automatic updating [11].

PPSpider is highly efficient, dynamically deployed web crawler for peer-to-peer based topic-specific crawler. Candidate seed are locally browsed web pages and are used to overcome the tunnel problems. PPSpider aids combined crawling process so as to adjust in the dynamic network environment and to reduce network communication flow among peers. Entry and exit of peers should be handled correctly because it has significant influence on the crawler [57].

Genetic programming enhances the relevance measure of web pages by focused crawler. In order to apply genetic programming evidence types, functions, fitness functions, genetic operators are needed to be defined. Fitness function ranks the documents in class; Macro FI is used as fitness function. This approach has overcome the problem of choosing low threshold/high threshold that result in low efficiency of crawler and loosing effective pages respectively by using decay concept [9].

DCrawler is scalable, fully distributed, platform independent, task de-centralized, and multiple agents based crawler. Using effective assignment function DCrawler partition the domain for crawling. Effective assignment function enhances capability of collaborating with web servers to obtain recent and updated results from the search engine. Multiple agents synchronize their behaviour autonomously to scan their share of the web [51].

Learning-based focused crawling approach is based on the four attributes: URL words, its anchor text, the parent pages, and the surrounding text. For classification Naïve Bayesian classifier is used. The accuracy of relevance forecast is clearly superior to related crawling methods. Dynamic update of the training dataset can enhance the accuracy level [84].

Probabilistic models are used for focused crawling use data collection, pattern learning and focused crawling processes as three tier architecture. Importance of fresh seen URL is estimated from data acquired from pages crawled earlier. Maximum entropy Markov model utilize anchor text to distinguish useful context, local classifier models, and linear chain conditional random field. Labelled page sequences are collected for using as training data. Initially selected target pages are used in training as well as crawling procedures. The presence of noise in data can hamper the results [60].

Advanced deep web crawler based on data object model deals with both dynamic and Ajax pages. Keyword ranking strategy is combination of three policies called random, generic and adaptive for choosing appropriate queries. The forms that are extracted are stored as feature sentences. This approach has considered only the main frame page for searching [62].

A learning automata theory based decentralized random algorithm for focused Web crawling decide most favourable action from a finite set of actions. Decentralized algorithm takes benefit of learning automata to arrange crawl in optimal manner. First web document that is crawled include documents that connect the crawled document to others. This algorithm updates its configuration adaptively according to the web dynamics and chooses documents with probability close to one [96].

A technique called VIQI stands for visual interpretation of query interfaces has been developed to deal with URL's semantics that are not rich in semantic values. To face this challenge a new model for query is represented which match query elements and interface of query. This approach imitates capability of users to understand query interfaces. Euclidean distance measures the closeness between the fields [13].

In multithreading micro blog crawling architecture based on the model named multi-producers and multi-consumers model. Incremental crawling considers each user a crawling unit. With increase in number of crawling threads crawling speed and weighted coverage also increase. As the number of crawling threads exceeds 30, weighted coverage starts decreasing [91].

Focus crawler is supervised learning based crawler which detects entry URL to a forum site. Flaws in Focus crawler lies in its classifier. SVMlight is weak page classifier even if it has popular voting method for index URL detection and thread URL detection. If half of the URL's in the set are classified incorrectly, majority voting technique fails. JavaScript-based URL's cannot be detected by Focus and it affects the precision and coverage factor. While in [28] unsupervised learning helps the crawler to work in uncontrolled environment of web. In the evaluation phase; the performance of the self-adaptive model did not completely meet set requirements regarding the parameters of precision and recall [47].

A selection algorithm for focused crawlers to incorporate semantic meta data evaluates the relevance of web page for focused web crawler. Semantic information includes domain and topic of the web page. A selection criterion is used to decide whether page is relevant or not, pages with high relevance are selected. This algorithm has the ability to locate resources with exact level of desired similarity. Only pages from Wikipedia are considered while evaluating the efficiency of algorithm [100].

Virtual machines (VM) are created by dividing multi core processors into number of virtual-machines, which can simultaneously carry out diverse crawling tasks on different initial data. The speedup factor achieved by the VM-based crawler is estimated and compared with crawler without virtualization, for crawling various numbers of documents. The effect of number of virtual machines on the speedup factor is investigated [3].

Retri blog framework is used for creating blogs with an architecture centered approach with information retrieval in blogs. Techniques are used from both information retrieval and extraction to handle dynamic nature of blog. Using services provided by Retri blog developers can easily create new crawlers. Any developer who is aquitained with design patterns can easily understand the Retri blog code [34].

Focused crawling based on history of web pages is clicked by user. User's history helps in constructing link context graph and relevancy context graph. Concept context graph consist of core concept mining and helps the crawler to guide the next crawling. Results proved that this approach outperforms breadth first, cosine similarity, the link context graph and the relevancy context graph [29].

Focused crawler has made use of semantic knowledge and social information with aim of finding relevant resources related to a topic. This model crawl the web pages from the site that considered the tags tagged by the users of the site. Social and semantic information is used by crawler to retrieve web pages that are tagged. Area of crawling is chosen from bookmarks site or from portal of resources been tagged. Tagged resources can be evidence for their inside content. Human cognition based patterns are utilized to discover the pages of users who have bookmarked web resources with the semantically relevant tags [10].

Singular Value Decomposition technique is used in latent semantic indexing. This model mines the reasonable substance of a collection of content by searching for connections between the content's terms. For the most part information is retrieved by taking into account exact matching of term in query with term in record [68]. While Pseudo-relevance feedback is used to compile queries iteratively and collect seeds in the search results [98].

A framework of distributed focused crawler is based on ontologies to direct the process of crawling to those segments of the web where relevant web pages are located. To select the initial pages external page ranking and breadth first strategy is used. Page ranker of the system use ontology to focus the crawling to specific domain. This parallel architecture proves that collective use of header, anchor, and surrounding text is more appropriate for link ranking than the use of the URL text or the full page [18].

An effective parallel web crawling by means of mobile agent and incremental crawling submit good pages and lessens the traffic on the system. The crawling process is transferred to host or server to commence downloading of web page. This crawler follows the technique of sifting through those HTML pages which have not been indexed during last crawl. In this way the remote site consume very less CPU cycles. Near duplicate detection feature help crawler by reducing unwanted downloads which improve the performance. Revisit policy infer updating activity for web pages [49].

HiCrawl is a domain oriented hidden web crawler. It crawl 'Medical' domain related sites. It consists of a downloader, a web page analyzer, a form analyzer and the form processor that uses a domain specific data repository and a domain specific classification hierarchy. System starts with a set of URL's that provide links to web pages belonging to the domain of consideration of the HiCrawl [39].

A fast distributed focused web crawler use publicly available proxy servers. This focused web crawler implement multi-threading distributed system in wide area network to overcome the problems of cost of implementing distributed crawler in wide area networks. In local area network only one global internet protocol address is used due to which it is misunderstood as impolite crawler. The crawler has implemented multithreaded programming which uses only one computer to run many crawlers. Multithreaded crawler have centralized controller and easier to maintain. Downloading speed of crawler decreases by time [1].

Semantic based focused web crawler which adds precision to web crawler. It saves the bandwidth because as focused web crawler predicts similarity before downloading the page. The accuracy of framework is promising. Not any link is discarded at once by considering it is as irrelevant, as these links are further considered to check whether links present on them are relevant or not. The search is ended after checking new irrelevant links if they do not lead to any relevant links [26].

The focused crawler based on ontology dependent tags makes the search extra detailed by increasing the search topic semantically. Manually tagged and semi-automatically tagged resource relevancy is compared to evaluate the harvest rate. The method is implemented on single area of social bookmarking site however this practice can be improved to multiple social bookmarking sites to build up all the tags from various sites [36].

For focused crawling in vertical search engine, Term Frequency-Topic Unbalanced Factor term (TF-TUF) weighting schema is used to convert the webpage into item vector, then formula of prediction method based on Naïve Bayes is used. TF/TUF assumed that the document term that are occurring multiple times are more important, short documents are important than long documents for the term matching; the unstable terms have judicious control to the term and contributes to calculate the topic significance [106].

The learning part of crawler detects several URL's and forms regular expressions based on that. The crawling part does the crawling of the web forum using these learned regular expressions. Gaussian kernel gives better results in terms of accuracy and convergence time. A new feature HasReplyBtn is added to the list of features used for index/thread page classification. A freshness first strategy is used for crawling [94].

A linked data crawler fetches HTML documents in addition to resource description framework (RDF) documents. Many HTML documents don't have embedded linked data that is not pointed to any RDF documents. Linked crawler increases discovery rate of RDF documents per unit of network bandwidth and decreases computation resources on non-RDF documents [32]**.**

Focused crawler is based on vector space model and text correlation analysis and use term frequency-inverse document frequency (TF-IDF). First it evaluates the word frequency in document and then measures how relevant a word is. Text relevancy analysis using TF-IDF and vector space model is used into the generic web crawler framework, which offers an inclusive set of techniques for fast and efficient crawling. TF-IDF doesn't provide additional hints about importance of information [81].

Crawler named an enhanced form-focused crawler (E-EFC) is based on a two-step page classification strategy which can accurately crawl the pages in both specific domain and the relevant domains. Harvest rate is improved in E-FFC by utilizing features of the possible "good quality" links. A novel Domain-Specific Form Classifier is based on ontology technology. As a result, the harvest rate can be greatly improved [54].

Keyword focused web crawler made use of content based keyword driven crawling with relevancy decision mechanism. Ontology concepts ensure the best path for improving crawler's performance. Number of extracted web pages are reduced due to best path choosen thus it takes less time for crawling as it downloads relevant web pages only. It does not take into account the relevance feedback. Parameter used is depth of 2, in deep web crawlers category, SmartCrawler in [107] works on depth of 3, to dig documents deeper [2].

Competent Crawling Algorithm (CCA) for web search is used to enhance efficiency of information retrieval. Search technique used is breadth first search. Compared to existing crawling algorithms, CCA has several advantages to increase the time efficiency by dequeuing the visited URL's from buffer before the crawler encounters it. The major advantages of the CCA are scalability and robustness. In CCA, dynamic hash tables are used for scalability and the system is reliable to crawler crashes. The complexity of the searching problem will overcome by CCA [88].

SemCrawl framework crawl the ontologies and semantic web documents. SemCrawl framework is implemented as well as validated on diverse collection of web pages. The proposed framework produces more inferential results by focusing on crawling ontology annotated semantic web documents for harvesting the knowledge base [27].

A prototype linked-based search system based on Imperialist Competitive Algorithm and folksonomy strategy handle different optimization tasks. Folksonomy attach tags or labels to each web page to suffice the practice and method of categorizing contents [83].

Distributed crawling system with browser integration is a high performance, browser content equivalent, web data retrieval system aimed to retrieve and prepare textual web content for ontology learning. This crawler is a high performance, browser content equivalent [48].

SmartCrawler architecture in [107] consists of in-site exploring to find searchable forms from the web page. Link tree for balanced link prioritizing is used in first stage. Adaptive learning algorithm updates information collected during crawl. Site and link ranker both are adaptive learning based. SmartCrawler has shown wide coverage for deep web interfaces. Prequery/Postquery approaches can be combined to improve the efficiency of crawler.

Literature above discussed the problems in focused crawling that have been trounced in the past years. Focused crawling is the call of time to reduce the network bandwidth and to save time etc. On the basis of literature reviewed we have divided the crawlers into various classes. Further the focused crawlers are classified into various types.

3. TYPES OF CRAWLERS

3.1. Forum Crawlers

Web forums are repositories of information. For web forums, the task of crawler is to download all the pages [7, 37]. Web forum are popular for their open discussions. Content here is always user created and is stored in databases on receiving request from user. Response page is generated dynamically based on predefined template. The forum site is connected by very complex graph [15]. Logical structure is always defined by the following rule:

website → forum list → (forum) thread list → (thread) posts [79].

A forum is a tree like structure. Forum is partitioned into classes for the interrelated discussions. The sub-forums can even have their own sub-forums. Threads are at lowest level. Members start on their discussions or posts under threads. Forums are ordered into a fixed set of topics with one major topic, driven and updated by members, and govern by a group referred as moderators.

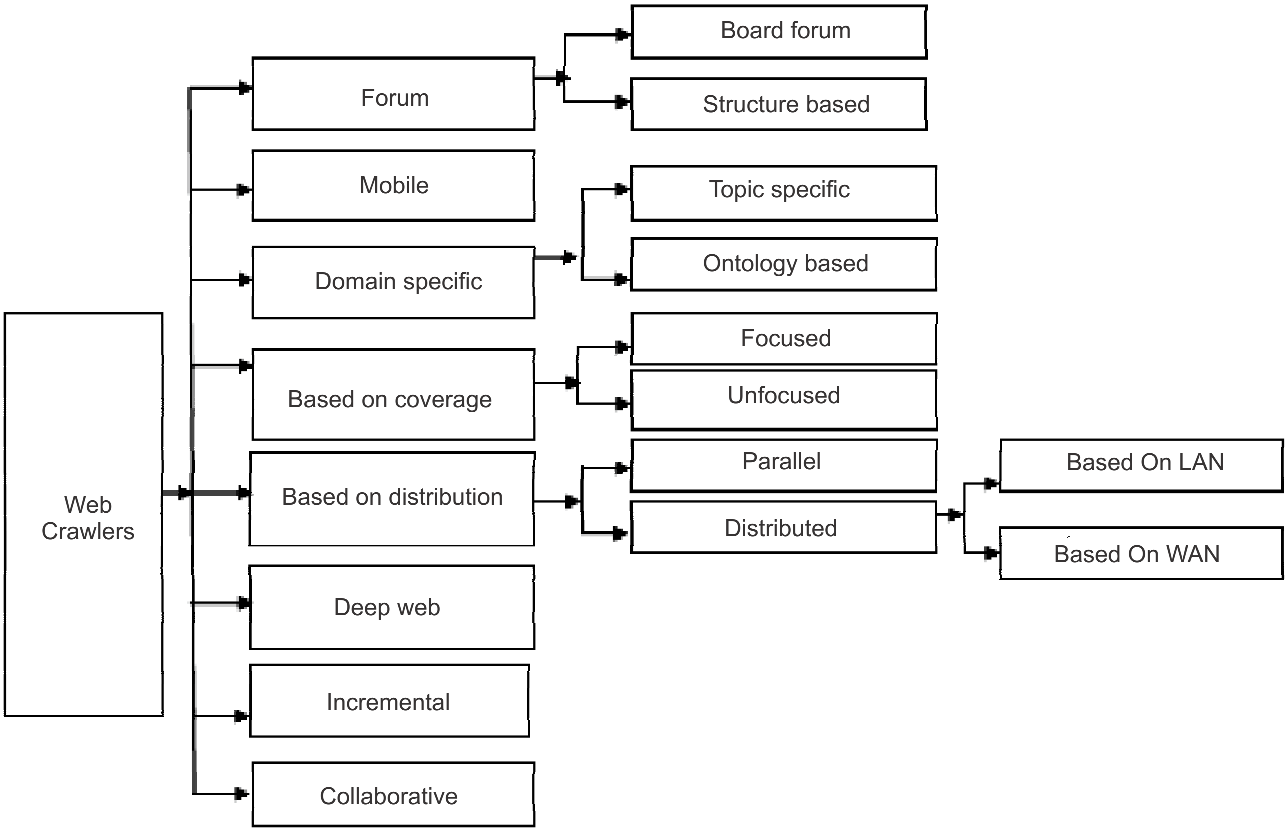

Figure 1: Types of web crawlers

3.1.1. Board Forum Crawling

This method of crawling web forum makes use of the ordered features of the web forum sites and replicate user actions of browsing web forums. This method commences crawling from the homepage, and so goes through every board of the site, all the post of the sites are crawled directly. Most meaningful data is crawled by board forum crawling efficiently and simply. Duplication detection is required in this approach [37].

3.1.2. Structure Driven Crawling

Structure-driven Crawler technique is used for learning regular expression patterns of URL's that guide a crawler to a target page. Target pages are found by comparing data object model trees of pages with a sample target page that is preselected. This technique works for the site from which sample page is drawn. The same process has to be repeated for every new site encountered. It is not suitable for crawling on large scale because for every new site process has to be repeated. Structure of the forum consists of cross links between forums, topics and threads that makes it difficult to fetch only new pages that have not been visited in one of previous crawls [37].

3.2. Mobile Crawlers

Mobility based crawlers crawl web pages with help of mobile agents. Crawlers are transferred to remote sites where they reside and filter out data and continuously study the web documents that are send to them. These crawlers reduce the load by compressing the document and transferring across the network [11].

3.3. Domain Specific Crawlers

These crawlers crawl the web pages based on a particular domain. Basically there are two types of domain specific crawlers: Topic specific crawlers and ontology based crawlers. iCrawler architecture in [55] is domain-specific crawler based on an intelligent agent technique. iCrawler outperforms the existing domain specific deep web form-focused crawlers when compared in terms of coverage rate, harvest rate, and time performance.

3.4. Based on Distribution

Load is distributed in order to drop off the bandwidth usage and boost coverage. Depending on load distribution, crawlers are of two types:

3.4.1. Parallel Crawlers

Parallel Crawlers run on the same local network and trade data through a fast interconnection. [23] recommended a universal design of a parallel crawler consist of several crawling processes named as C-Proc. Each process performs the responsibilities of which a single process crawler performs. It downloads pages from the web, stores the pages locally, extracts URL's from the downloaded pages and follows the links [49, 78]. In parallel crawling each processor has its own scheduler and bot [66]. In [65] Bulk Synchronous Parallel crawler helps in reducing the overheads with threads synchronization. In [97] domain specific intelligent parallel crawler keeps particular queue, for the diverse domains like, .edu, .org, .ac, .com etc. Crawl workers receive URL's for these queues. This leads to the load sharing by the parallel crawlers on the basis of specific domains. In [90] architecture of a parallel crawler is developed based on augmented hypertext documents. This architecture focuses on providing parallelization at document, mapper, and the crawl worker level.

3.4.2. Distributed Crawlers

Distributed crawlers run in geographically disperse localities joined through internet. Multiple crawlers crawl information concurrently on different networks [78]. In order to crawl thousands of sites within a short time frame, and to allow complex content processing, it is necessary to distribute the crawling process on a cluster [41]. For large search engine a single crawler is inadequate, search engine needs to fetch huge amount of data in very less time and without duplication [77]. In order to avoid duplication of work, each event should be assigned to event executing crawler [69]. In [109] a method for fully distributed web crawler on structured network is provided to achieve scalability and load balancing by dividing each node of system into crawling module and control module to download web pages and to manage the communication with other nodes. In [38] context based distributed focused crawler result in a list of more accurate web pages to achieve the user prerequisite intended for a particular subject of search. User gets his quick response from top layer as only local database is searched.

Writing a code for web crawler is easy but scaling the crawler is quite difficult task. Najork in [71] have specified three techniques of crawling where first is when whole web can be crawled from one data centre and pages can be replicated to other data centres to perform independent crawls at each data centre and thus serve different directory to diverse geographies. Second option is to perform a single geographically-distributed crawl, where crawlers in a given data centre crawl web servers that are close-by, and then propagate the crawled pages to their peer data centres. The third is if existing designs for distributed crawlers would scale to a geographically distributed setting.

3.5. Deep Web Crawlers

The Deep Web is defined as the content hidden behind HTML forms. To retrieve hidden content, a user has to submit form with valid input values. Generic search engines are not able to find data stored to the deep Web. Crawlers can't creep into information that requires keyword searches on a single specific web site [95]. Deep web crawlers on comparison with traditional crawlers, have execution sequence that contains additional steps for pages on which forms are detected [80]. The technical barrier for search engines is to extract information from hidden resources. A crawler cannot interact with them like human beings can. For instance private web which is a kind of hidden web consists of personal unpublished information and most of webmasters do not want search engines to index their pages [95]. One of the hidden web crawler architecture was proposed in [82]. This design automatically process, analyze, and submit forms using an internal model of forms and form submissions. In deep World Wide Web crawling identifying which HTML forms are meant for querying is challenge. Deep web crawling is divided into three parts discovering, submitting the forms and extraction of information from results of submitted queries [70].

3.6. Incremental Crawlers

An incremental crawler updates its collection of index on an incremental basis after its target build up is finally reached based on an estimate. It refreshes the existing collection by new updates on a periodical basis. It helps to save network bandwidth [78].

3.7. Collaborative Crawlers

A collaborative web crawling system employ more than one crawlers planned in a collaborative fashion for exploring web, generate, store summaries for the highest efficiency, load balancing and minimizing overhead. Coordination is achieved by partitioning the URL space in to sub-space. Each subspace is assigned to a processor, and processor build the abstract for URL's assigned within sub-space [22].

3.8. Based on Coverage

3.8.1. Focused Crawlers

Aim of focused crawler is to obtain only a subset of web. It makes the route to find the links which are more likely to lead to relevant pages by analyzing the links it has visited already. Classic focused crawler follow predefined links while the learning focused crawler are based on the training set that is updated dynamically with new links added in crawl. These crawlers filter the links that are not relevant. Efficiency of search engine directly depends on focused crawlers. Topical crawlers support decentralizing the crawling process, which is a more scalable approach [67].

3.8.1.1. Classic Focused Crawlers

The topic which classic focused crawlers search is user query. A criterion of high download of pages is based on links and their probability to prompt pages on query. The crawler continues recursively on the links contained in the downloaded pages. Similarity between topic and anchor text, query and text of page are used for computing download priorities. The advantage of focused crawler is less time spending, money and effort processing in terms of downloading only necessary web pages. [67] introduced the concept of topical crawling, where focused crawler term is coined in [19]. Classical web crawlers use hyperlinks on the web pages for crawling while in case of focused crawlers only relevant pages are downloaded.

3.8.1.2. Priority Based Crawlers

Focused crawling based on priority of web pages has been introduced in [24]. The web pages that belong to URL are downloaded from web, if there is new URL present, and then corresponding pages are downloaded. Semantic score is calculated between topic of search and pages that were downloaded and this semantic score is added into priority queue. Each time maximum similarity score is selected for crawling, priority queue will return maximum score for URL. This is the benefit of employing priority queue over simple queue.

3.8.1.3. Semantic Crawlers

Semantic crawlers assign downloaded priorities to pages by exercising semantic similarity to calculate page and topic relevance by sharing conceptual terms. Ontology defines the conceptual similarity in involved terms [86].

3.8.1.4. Learning Crawlers

As per name learning crawlers are based on a training process to lead the crawler and prioritize which web pages to visit first. Training set is used to train learning crawler and it consists of relevant and irrelevant web. A system designed in [59] has used hidden markov model for predicting the links can lead the crawler to the relevant pages. Links with the higher priorities are extracted for relevant topic classification.

3.8.1.5. Form Focused Crawlers

Aim of Form Focused Crawler (FFC) is to efficiently locate forms on public indexbale web. FFC is built on three supervised classifiers named as page classifier, link classifier and form classifier. To guide search for page and link classifiers crawler use naïve bayes textual classifiers. Page classifier classify pages that fit in to a specific domain page classifier is used. Form classifier separates non query forms from query forms. FFC focus on page content and patterns within and around hyperlinks in path to a web page [8].

3.8.1.6. Context Based Focused Crawlers

[52] Proposed a context model for focused web search and [45] has introduced a framework using combination of the link structure analysis and content similarity for focused crawler.

3.8.1.7. Focused Linked Data Crawlers

Both HTML and RDF documents are fetched by linked data crawler, while analyzing new extracted HTML links pointing to an HTML document that contains an RDF link or embedded linked data. The crawler then assigns higher priority to such links than other HTML links. This can help to harvest linked data published in both RDF and HTML documents and also improve discovery rate of RDF documents [26].

3.8.1.8. Bootstrap Crawlers

The new category called bootstrap focused crawler (BFC) is introduced in [98] which shows the importance of URL seed to frontier. Pseudo-relevance feedback is used to iteratively compile query. BFC collects seed pages, with meek overhead.

3.8.1.9. Ontology Based Crawlers

Ontology based research is divided into two parts. In first ontology in a particular area can be configured and it helps in assisting knowledge analysis. Second is to learn how to create and correspond in particular with ontology [21]. Ontology value the importance of the notion of the user query. Ontology contains aspects, explanation and various features of the concept [17].

3.8.2. Unfocused Crawlers

Unfocused crawlers are not focused towards a specific topic. They collect the entire contents of the web in a centralized location. Pages are indexed in advance to be able to respond to any possible query.

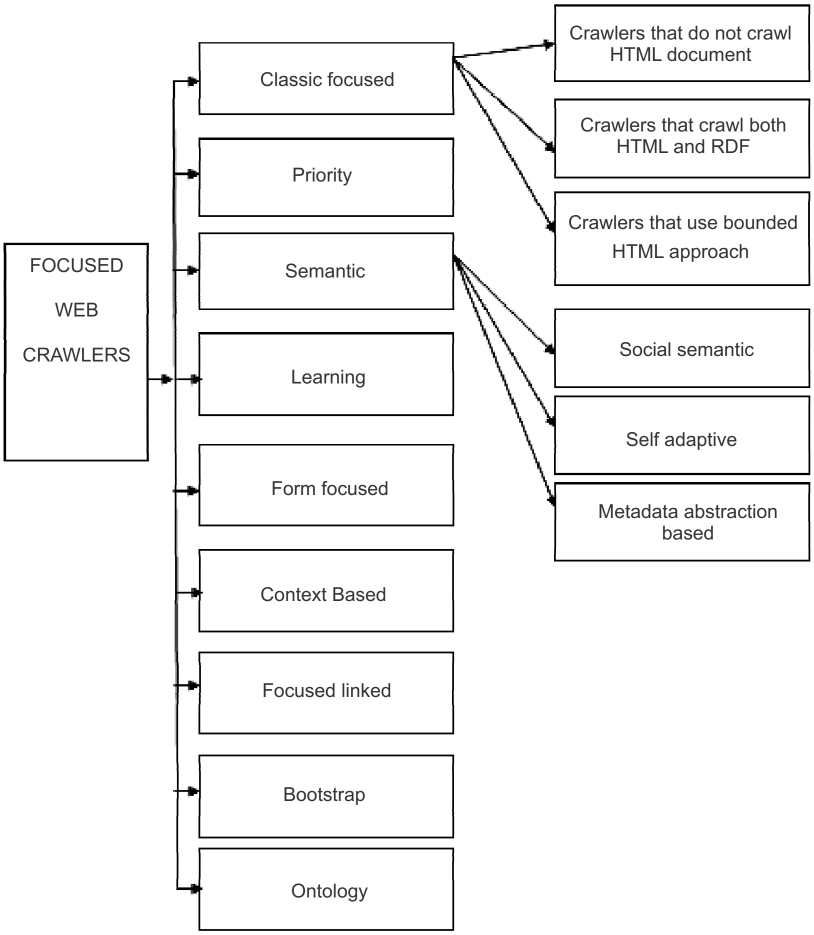
FOCUSED
WEB
CRAWLERS
Classic focused
Priority
Semantic
Learning
Form focused
Context Based
Focused linked
Bootstrap
Ontology
Crawlers that do not crawl HTML document
Crawlers that crawl both HTML and RDF
Crawlers that use bounded HTML approach
Social semantic
Self adaptive
Metadata abstraction based

Figure 2: Types of focused web crawlers

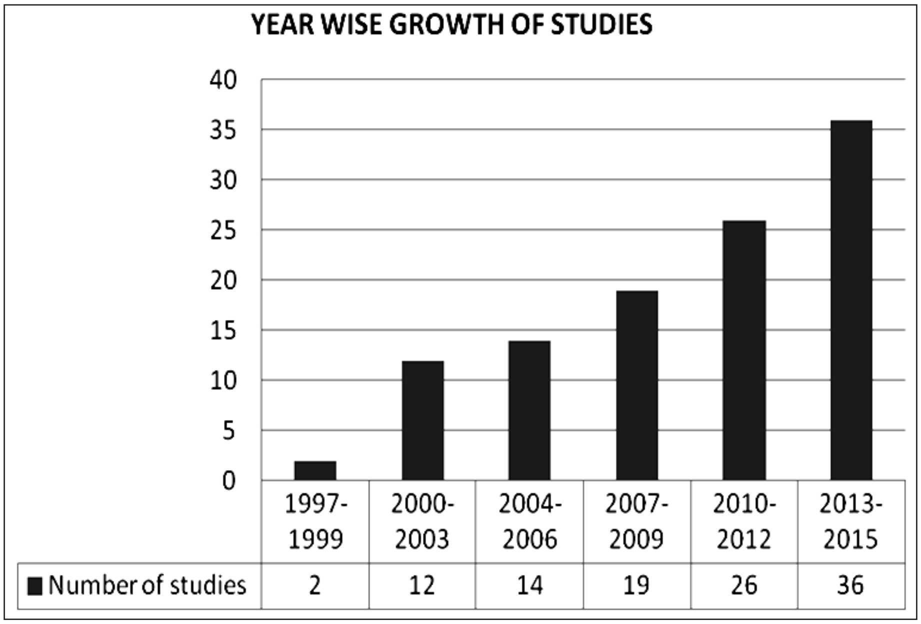

Figure 3: Scale of development in focused crawlers

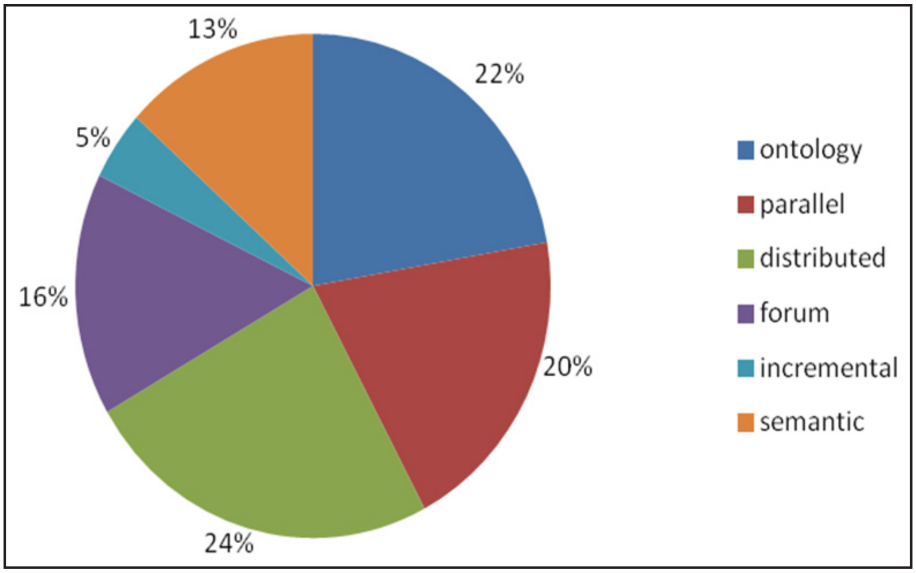

Figure 4: Distribution of various types of crawler

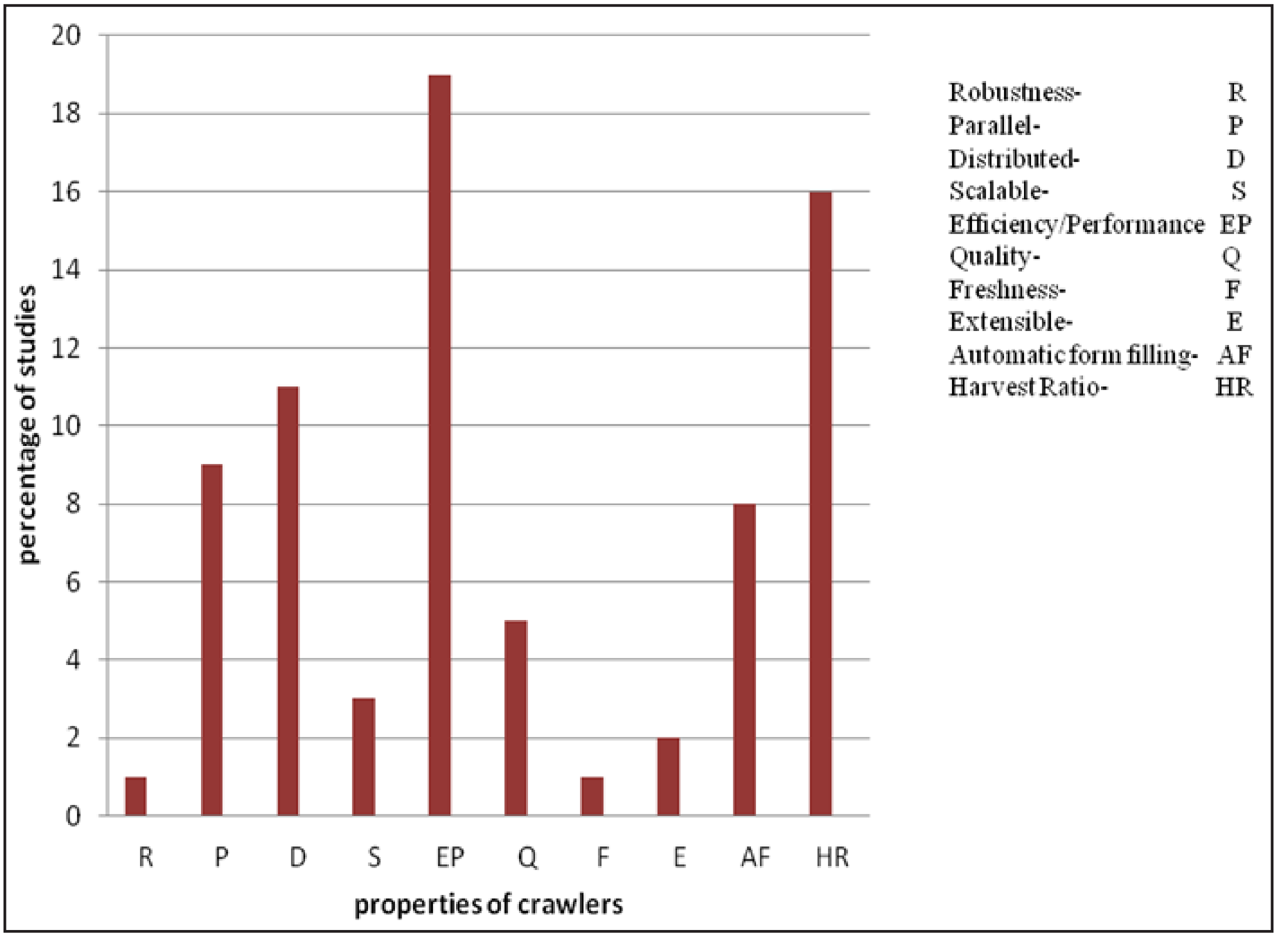

Figure 5: Expected properties from the crawlers

4. DISCUSSION

Distributed crawlers are the need of today's era due to growth of internet data and users. Distribution can be performed by assigning crawler nodes to specific sets of hosts which helps to drop the bandwidth. Figure 3 shows the year wise growth of studies in focused crawling. Figure 4 shows considerable contribution of distributed web crawlers. Figure 5 shows expected properties from crawlers in which much of the work is done on enhancing the efficiency and performance and harvest rate of the crawler

5. CONCLUSION

Size of web is ever-increasing, parallelizing the crawling process is essential for downloading the web pages in an amount of time as less as possible. For large scale search engines a single crawling process will not be sufficient. Large scale search engines search huge amounts of data rapidly even if multi threading is used. Data fetched by single centralized crawler go by a single physical link. If crawling process is distributed in several processes it can facilitate to build scalable system.

REFERENCES

[1] Achsan HT, Wibowo WC. A Fast Distributed Focused-web Crawling. Procedia Engineering. 2014 Dec 31;69:492-9.

[2] Agre GH, Mahajan NV. Keyword focused web crawler. InElectronics and Communication Systems (ICECS), 2015 2nd International Conference on 2015 Feb 26 (pp. 1089-1092). IEEE..

[3] Al-Bahadili H, Qtishat H. Application of VM-Based Computations to Speed Up the Web Crawling Process on Multi-core Processors. InDistributed Computing and Applications to Business, Engineering & Science (DCABES), 2013 12th International Symposium on 2013 Sep 2 (pp. 157-161). IEEE.

[4] Álvarez, M., Raposo, J., Pan, A., Cacheda, F., Bellas, F., & Carneiro, V. (2007, June). DeepBot: a focused crawler for accessing hidden web content. In Proceedings of the 3rd international workshop on Data engineering issues in E-commerce and services: In conjunction with ACM Conference on Electronic Commerce (EC'07) (pp. 18-25). ACM.

[5] Arora M, Kanjilal U, Varshney D. Challenges in Web Information Retrieval. In Innovations in Computing Sciences and Software Engineering 2010 (pp. 141-146). Springer Netherlands.

[6] Baeza-Yates R. Information retrieval in the web: beyond current search engines. International Journal of Approximate Reasoning. 2003 Nov 30;34(2):97-104.

[7] Bamrah NH, Satpute BS, Patil P. Web forum crawling techniques. International Journal of Computer Applications. 2014 Jan 1;85(17).

[8] Barbosa L, Freire J. Searching for Hidden-Web Databases. InWebDB 2005 Jun 16 (pp. 1-6).

[9] Bazarganigilani M, Syed A, Burki S. Focused web crawling using decay concept and genetic programming. International Journal of Data Mining & Knowledge Management Process (IJDKP). 2011 Jan;1(1):1-2.

[10] Bedi P, Thukral A, Banati H. Focused crawling of tagged web resources using ontology. Computers & Electrical Engineering. 2013 Feb 28;39(2):613-28.

[11] Bhatia KK, Sharma AK. A Framework for an Extensible Domain-specific Hidden Web Crawler (DSHWC). communicated to IEEE TKDE Journal Dec. 2008.

[12] Bhatia KK, Sharma AK, Madaan R. AKSHR: A novel framework for a Domain-specific Hidden Web Crawler. InParallel Distributed and Grid Computing (PDGC), 2010 1st International Conference on 2010 Oct 28 (pp. 307-312). IEEE

[13] Boughammoura R, Hlaoua L, Omri MN. VIQI: a new approach for visual interpretation of deep web query interfaces. InInformation Technology and e-Services (ICITeS), 2012 International Conference on 2012 Mar 24 (pp. 1-6). IEEE

[14] Brin S, Page L. Reprint of: The anatomy of a large-scale hypertextual web search engine. Computer networks. 2012 Dec 17;56(18):3825-33.

[15] Cai R, Yang JM, Lai W, Wang Y, Zhang L. iRobot: An intelligent crawler for Web forums. InProceedings of the 17th international conference on World Wide Web 2008 Apr 21 (pp. 447-456). ACM.

[16] Cambazoglu BB, Baeza-Yates R. Scalability Challenges in Web Search Engines. Synthesis Lectures on Information Concept, Retrieval, and Services. 2015 Dec 29;7(6):1-38.

[17] Cambazoglu BB, Baeza-Yates R. Scalability and efficiency challenges in large-scale web search engines. InProceedings of the Eighth ACM International Conference on Web Search and Data Mining 2015 Feb 2 (pp. 411-412). ACM.

[18] Campos R, Rojas O, Marin M, Mendoza M. Distributed Ontology-Driven Focused Crawling. InParallel, Distributed and Network-Based Processing (PDP), 2013 21st Euromicro International Conference on 2013 Feb 27 (pp. 108-115). IEEE.

[19] Chakrabarti S, Van den Berg M, Dom B. Focused crawling: a new approach to topic-specific Web resource discovery. Computer Networks. 1999 May 17;31(11):1623-40.

[20] Chen X, Zhang X. Hawk: A focused crawler with content and link analysis. Ine-Business Engineering, 2008. ICEBE'08. IEEE International Conference on 2008 Oct 22 (pp. 677-680). IEEE.

[21] Chien HC, Su TH. Study and implementation of a learning content management system search engine for special education based on semantic web (Doctoral dissertation, Master Thesis, MingHsin University of Science and Technology, HsinChu, Taiwan).

[22] Cho J, Garcia-Molina H. The evolution of the web and implications for an incremental crawler. 26th Intl. InConf. on Very Large Databases 2000 Sep.

[23] Cho J, Garcia-Molina H. Parallel crawlers. InProceedings of the 11th international conference on World Wide Web 2002 May 7 (pp. 124-135). ACM.

[24] Choudhary J, Roy D. Priority based Semantic Web Crawler. International Journal of Computer Applications. 2013 Nov;81(15):10-3.

[25] Dahiwale P, Mokhade A, Raghuwanshi MM. Intelligent web crawler. InProceedings of the International Conference and Workshop on Emerging Trends in Technology 2010 Feb 26 (pp. 613-617). ACM.

[26] Dahiwale P, Raghuwanshi MM, Malik L. Design of improved focused web crawler by analyzing semantic nature of URL and anchor text. InIndustrial and Information Systems (ICIIS), 2014 9th International Conference on 2014 Dec 15 (pp. 1-6). IEEE.

[27] Dhingra V, Bhatia KK. SemCrawl: Framework for crawling ontology annotated web documents for intelligent information retrieval. InIntelligent Distributed Computing 2015 (pp. 213-223). Springer International Publishing.

[28] Dong H, Hussain FK. Self-adaptive semantic focused crawler for mining services information discovery. Industrial Informatics, IEEE Transactions on. 2014 May;10(2):1616-26.

[29] Du Y, Pen Q, Gao Z. A topic-specific crawling strategy based on semantics similarity. Data & Knowledge Engineering. 2013 Nov 30;88:75-93.

[30] Duda C, Frey G, Kossmann D, Matter R, Zhou C. Ajax crawl: Making ajax applications searchable. InData Engineering, 2009. ICDE'09. IEEE 25th International Conference on 2009 Mar 29 (pp. 78-89). IEEE.

[31] Ehrig M, Maedche A. Ontology-focused crawling of Web documents. InProceedings of the 2003 ACM symposium on Applied computing 2003 Mar 9 (pp. 1174-1178). ACM.

[32] Emamdadi R, Kahani M, Zarrinkalam F. A focused linked data crawler based on HTML link analysis. InComputer and Knowledge Engineering (ICCKE), 2014 4th International eConference on 2014 Oct 29 (pp. 74-79). IEEE.

[33] Ferrara E, De Meo P, Fiumara G, Baumgartner R. Web data extraction, applications and techniques: A survey. Knowledge-based systems. 2014 Nov 30;70:301-23.

[34] Ferreira, R., Freitas, F., Brito, P., Melo, J., Lima, R., & Costa, E. (2013). RetriBlog: an architecture-centered framework for developing blog crawlers. Expert Systems with Applications, 40(4), 1177-1195.

[35] Gao Q, Xiao B, Lin Z, Chen X, Zhou B. A high-precision forum crawler based on vertical crawling. InNetwork Infrastructure and Digital Content, 2009. IC-NIDC 2009. IEEE International Conference on 2009 Nov 6 (pp. 362-367). IEEE.

[36] Gaur R, Sharma DK. Focused crawling with ontology using semi-automatic tagging for relevancy. InContemporary Computing (IC3), 2014 Seventh International Conference on 2014 Aug 7 (pp. 501-506). IEEE.

[37] Guo Y, Li K, Zhang K, Zhang G. Board forum crawling: a Web crawling method for Web forum. InProceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence 2006 Dec 18 (pp. 745-748). IEEE Computer Society.

[38] Gupta P, Sharma A, Gupta JP, Bhatia K. A Novel Framework for Context Based Distributed Focused Crawler (CBDFC). Int. J. Computer and Communication Technology. 2009;1(1):14.

[39] Gupta S, Bhatia KK. HiCrawl: A Hidden Web Crawler for Medical Domain. InComputational and Business Intelligence (ISCBI), 2013 International Symposium on 2013 Aug 24 (pp. 152-157). IEEE.

[40] Hati D, Kumar A. An approach for identifying URLs based on division score and link score in focused crawler. International Journal of Computer Applications. 2010 May;2(3):48-53.

[41] Hati D, Kumar A. UDBFC: An effective focused crawling approach based on URL Distance calculation. InComputer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on 2010 Jul 9 (Vol. 2, pp. 59-63). IEEE.

[42] Hu K, Wong WS. A probabilistic model for intelligent Web crawlers. InComputer Software and Applications Conference, 2003. COMPSAC 2003. Proceedings. 27th Annual International 2003 Nov 3 (pp. 278-282). IEEE.

[43] Huang Y, Ye Y. wHunter: a focused web crawler–a tool for digital library. InDigital Libraries: International Collaboration and Cross-Fertilization 2004 Dec 13 (pp. 519-522). Springer Berlin Heidelberg.

[44] Ipeirotis PG, Gravano L. Distributed search over the hidden web: Hierarchical database sampling and selection. InProceedings of the 28th international conference on Very Large Data Bases 2002 Aug 20 (pp. 394-405). VLDB Endowment.

[45] Jamali M, Sayyadi H, Hariri BB, Abolhassani H. A method for focused crawling using combination of link structure and content similarity. InProceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence 2006 Dec 18 (pp. 753-756). IEEE Computer Society.

[46] Ledford JL. Search Engine Optimization Bible. John Wiley & Sons; 2009 Apr 29.

[47] Jiang J, Song X, Yu N, Lin CY. Focus: learning to crawl web forums. Knowledge and Data Engineering, IEEE Transactions on. 2013 Jun;25(6):1293-306.

[48] Juffinger A, Neidha T, Weichselbraun A, Wohlgenannt G, Granitzer M, Kern R, Scharl A. Distributed Web2. 0 crawling for ontology evolution. InDigital Information Management, 2007. ICDIM'07. 2nd International Conference on 2007 Oct 28 (Vol. 2, pp. 615-620). IEEE.

[49] Kausar MA, Dhaka VS, Singh SK. An Effective Parallel Web Crawler based on Mobile Agent and Incremental Crawling. Journal of Industrial and Intelligent Information. 2013 Mar;1(1).

[50] Nasar M, Kausar MA, Singh SK. Application of Soft Computing In Information Retrieval-A Review of Literature. International Journal of Computer Science. 2013;3(03):10-7.

[51] Kumar MS, Neelima P. Design and implementation of scalable, fully distributed web crawler for a web search engine. International Journal of Computer Applications (0975-8887). 2011 Feb.

[52] Kumar S, Chauhan N. A Context Model For Focused Web Search. Published in International Journal of Computers and Technology. 2012 Jun 30;2(3).

[53] Le Breton G, Bergeron N, Hallé S. A Reference Framework for the Automated Exploration of Web Applications. InEngineering of Complex Computer Systems (ICECCS), 2014 19th International Conference on 2014 Aug 4 (pp. 81-90). IEEE.

[54] Li Y, Wang Y, Du J. E-FFC: an enhanced form-focused crawler for domain-specific deep web databases. Journal of Intelligent Information Systems. 2013 Feb 1;40(1):159-84.

[55] Li Y, Wang Y, Tian E. A New Architecture of an Intelligent Agent-Based Crawler for Domain-Specific Deep Web Databases. InWeb Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on 2012 Dec 4 (Vol. 1, pp. 656-663). IEEE.

[56] Liddle SW, Embley DW, Scott DT, Yau SH. Extracting data behind web forms. InAdvanced Conceptual Modeling Techniques 2002 Oct 7 (pp. 402-413). Springer Berlin Heidelberg.

[57] Liu D, Liu J. PPSpider: Towards an Efficient and Robust Topic-Specific Crawler Based on Peer-to-Peer Network. InComputer Science and Engineering, 2009. WCSE'09. Second International Workshop on 2009 Oct 28 (Vol. 1, pp. 101-105). IEEE.

[58] Liu F, Ma FY, Ye YM, Li ML. IGLOOG: A distributed Web crawler based on grid service. InWeb Technologies Research and Development-APWeb 2005 2005 Mar 29 (pp. 207-216). Springer Berlin Heidelberg.

[59] Liu H, Janssen J, Milios E. Using HMM to learn user browsing patterns for focused web crawling. Data & Knowledge Engineering. 2006 Nov 30;59(2):270-91.

[60] Liu H, Milios E, Janssen J. Probabilistic models for focused web crawling. InProceedings of the 6th annual ACM international workshop on Web information and data management 2004 Nov 12 (pp. 16-22). ACM.

[61] Luo J, Shi Z, Wang M, Wang W. Parallel web spiders for cooperative information gathering. InGrid and Cooperative Computing-GCC 2005 2005 Nov 30 (pp. 1192-1197). Springer Berlin Heidelberg.

[62] Luo J, Shi Z, Wang M, Wang W. Parallel web spiders for cooperative information gathering. InGrid and Cooperative Computing-GCC 2005 2005 Nov 30 (pp. 1192-1197). Springer Berlin Heidelberg.

[63] Madhavan J, Ko D, Kot Ł, Ganapathy V, Rasmussen A, Halevy A. Google's deep web crawl. Proceedings of the VLDB Endowment. 2008 Aug 1;1(2):1241-52.

[64] Marchiori M. Security of World Wide Web Search Engines. Reliability, Quality and Safety of Software-Intensive Systems.

[65] Marin M, Bonacic C. Bulk-Synchronous On-Line Crawling on Clusters of Computers. InParallel, Distributed and Network-Based Processing, 2008. PDP 2008. 16th Euromicro Conference on 2008 Feb 13 (pp. 414-421). IEEE.

[66] Marin M, Paredes R, Bonacic C. High-performance priority queues for parallel crawlers. InProceedings of the 10th ACM workshop on Web information and data management 2008 Oct 30 (pp. 47-54). ACM.

[67] Menczer F, Pant G, Srinivasan P. Topical web crawlers: Evaluating adaptive algorithms. ACM Transactions on Internet Technology (TOIT). 2004 Nov 1;4(4):378-419.

[68] Minhas G, Kumar M. LSI based relevance computation for topical web crawler. Journal of Emerging Technologies in Web Intelligence. 2013 Jan 11;5(4):401-6.

[69] Mirtaheri SM, Zou D, Bochmann GV, Jourdan GV, Onut IV. Dist-ria crawler: A distributed crawler for rich internet applications. InP2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2013 Eighth International Conference on 2013 Oct 28 (pp. 105-112). IEEE.

[70] Moraes MC, Heuser CA, Moreira VP, Barbosa D. Prequery discovery of domain-specific query forms: A survey. Knowledge and Data Engineering, IEEE Transactions on. 2013 Aug;25(8):1830-48.

[71] Najork M. Web crawler architecture. InEncyclopedia of Database Systems 2009 (pp. 3462-3465). Springer US.

[72] Nemeslaki A, Pocsarovszky K. Web crawler research methodology.

[73] Novak B. A survey of focused web crawling algorithms. Proceedings of SIKDD. 2004 Oct 12;5558.

[74] Olston C, Najork M. Web crawling. Foundations and Trends in Information Retrieval. 2010 Mar 1;4(3):175-246.

[75] Oppenheim C, Morris A, McKnight C, Lowley S. The evaluation of WWW search engines. Journal of documentation. 2000 Apr 1;56(2):190-211.

[76] Pant G, Menczer F. MySpiders: Evolve your own intelligent Web crawlers. Autonomous agents and multi-agent systems. 2002 Jun 1;5(2):221-9.

[77] Patel P, Hasan M, Tanawala B. DISTRIBUTED HIGH PERFORMANCE WEB CRAWLER. IJITR. 2013 May 28;1(3):236-9.

[78] Pavalam SM, Raja SK, Jawahar M, Akorli FK. Web Crawler in Mobile Systems. International Journal of Machine Learning and Computing. 2012 Aug 1;2(4):531.

[79] Pavkovic M, Protic J. Intelligent crawler for web forums based on improved regular expressions. InTelecommunications Forum (TELFOR), 2013 21st 2013 Nov 26 (pp. 817-820). IEEE.

[80] Peisu X, Ke T, Qinzhen H. A framework of deep Web crawler. InControl Conference, 2008. CCC 2008. 27th Chinese 2008 Jul 16 (pp. 582-586). IEEE.

[81] Qin Q, Peng X. A Focused Crawler Based on Correlation Analysis. International Journal of Future Generation Communication and Networking. 2014;7(6):13-20.

[82] Garcia-Molina H, Raghavan S. Crawling the Hidden Web. In27th International Conference on Very Large Data Bases 2001.

[83] Rasekh I. A new competitive intelligence-based strategy for web page search. InSemantic Computing (ICSC), 2015 IEEE International Conference on 2015 Feb 7 (pp. 120-126). IEEE.

[84] Safran MS, Althagafi A, Che D. Improving relevance prediction for focused Web crawlers. InComputer and Information Science (ICIS), 2012 IEEE/ACIS 11th International Conference on 2012 May 30 (pp. 161-166). IEEE.

[85] Sakkis G. YAFC: Yet Another Focused Crawler.

[86] Sampat J, Jain A, Mistry D. Focused Web Crawler and its Approaches.

[87] Santhosh KD, Kamath M. Design and implementation of competent web crawler and indexer using web services. InAdvanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on 2014 May 8 (pp. 1672-1677). IEEE.

[88] Saranya S, Zoraida BS, Paul PV. A Study on Competent Crawling Algorithm (CCA) for Web Search to Enhance Efficiency of Information Retrieval. InArtificial Intelligence and Evolutionary Algorithms in Engineering Systems 2015 (pp. 9-16). Springer India.

[89] Selamat A, Ahmadi-Abkenari F. Application of clickstream analysis as web page importance metric in parallel crawlers. InInformation Technology (ITSim), 2010 International Symposium in 2010 Jun 15 (Vol. 1, pp. 1-6). IEEE.

[90] Sharma AK, Gupta JP, Agarwal DP. Parcahyd: an architecture of a parallel crawler based on augmented hypertext documents. International Journal of Advancements in Technology. 2010 Oct;1(2):270-83.

[91] Shen D, Wang H, Cao J, Li P, Jiang Z. The design and implement of high efficient incremental microblogging crawler. InMultimedia Information Networking and Security (MINES), 2012 Fourth International Conference on 2012 Nov 2 (pp. 537-540). IEEE.

[92] Shkapenyuk V, Suel T. Design and implementation of a high-performance distributed web crawler. InData Engineering, 2002. Proceedings. 18th International Conference on 2002 (pp. 357-368). IEEE.

[93] Sigurðsson K. *Adaptive Revisiting with Heritrix* (Doctoral dissertation, Master Thesis. Reykjavik: University of Iceland, Faculty of Engineering, 2005. 96 p. Supervisors Helgi Þorbergsson, PhD, Þorsteinn Hallgrímsson).

[94] Sreeja SR, Chaudhari S. An Effective Forum Crawler. InCircuits, Systems, Communication and Information Technology Applications (CSCITA), 2014 International Conference on 2014 Apr 4 (pp. 230-234). IEEE.

[95] Tjin-Kam-Jet KT, Trieschnigg D, Hiemstra D. Deep web search: an overview and roadmap.

[96] Torkestani JA. An adaptive focused web crawling algorithm based on learning automata. Applied Intelligence. 2012 Dec 1;37(4):586-601.

[97] Tyagi N, Gupta D. A novel architecture for domain specific parallel crawler. Department of Computer Engineering, Shobhit University. Meerut, India. 2010.

[98] Vieira K, Barbosa L, da Silva AS, Freire J, Moura E. Finding seeds to bootstrap focused crawlers. World Wide Web. 2015 Feb 26:1-26.

[99] Vikas O, Chiluka NJ, Ray PK, Meena G, Meshram AK, Gupta A, Sisodia A. WebMiner--Anatomy of Super Peer Based Incremental Topic-Specific Web Crawler. Innull 2007 Apr 22 (p. 32). IEEE.

[100] Wadwekar S, Mukhopadhyay D. A Selection Algorithm for Focused Crawlers Incorporating Semantic Metadata. InDistributed Computing and Internet Technology 2013 Feb 5 (pp. 561-572). Springer Berlin Heidelberg.

[101] Wu P, Wen JR, Liu H, Ma WY. Query selection techniques for efficient crawling of structured web sources. InData Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on 2006 Apr 3 (pp. 47-47). IEEE.

[102] Yang SY. OntoCrawler: A focused crawler with ontology-supported website models for information agents. Expert Systems with Applications. 2010 Jul 31;37(7):5381-9.

[103] Yang SY, Hsu CL. Ontology-supported focused-crawler for specified scholar’s webpages. InIntelligent Systems Design and Applications, 2008. ISDA’08. Eighth International Conference on 2008 Nov 26 (Vol. 2, pp. 409-414). IEEE.

[104] Yuvarani M, Iyengar NC, Kannan A. LSCrawler: a framework for an enhanced focused web crawler based on link semantics. InWeb Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on 2006 Dec (pp. 794-800). IEEE.

[105] Zhang C, Zhang J. Inforce: Forum data crawling with information extraction. InUniversal Communication Symposium (IUCS), 2010 4th International 2010 Oct 18 (pp. 367-373). IEEE.

[106] Zhang W, Chen Y. Bayes topic prediction model for focused crawling of vertical search engine. InComputing, Communications and IT Applications Conference (ComComAp), 2014 IEEE 2014 Oct 20 (pp. 294-299). IEEE.

[107] Zhao F, Zhou J, Nie C, Huang H, Jin H. SmartCrawler: A Two-stage Crawler for Efficiently Harvesting Deep-Web Interfaces.

[108] Zheng X, Zhou T, Yu Z, Chen D. URL Rule based focused crawler. Ine-Business Engineering, 2008. ICEBE’08. IEEE International Conference on 2008 Oct 22 (pp. 147-154). IEEE.

[109] Zhu K, Xu Z, Wang X, Zhao Y. A full distributed web crawler based on structured network. InInformation Retrieval Technology 2008 Jan 15 (pp. 478-483). Springer Berlin Heidelberg.