Overview

A survey of Web crawlers for information retrieval

Manish Kumar,[1]* Rajesh Bhatia[1] and Dhavleesh Rattan[2]

Performance of any search engine relies heavily on its Web crawler. Web crawlers are the programs that get webpages from the Web by following hyperlinks. These webpages are indexed by a search engine and can be retrieved by a user query. In the area of Web crawling, we still lack an exhaustive study that covers all crawling techniques. This study follows the guidelines of systematic literature review and applies it to the field of Web crawling. We used the standard procedure of carrying out a systematic literature review on 248 studies from a total of 1488 articles published in 12 leading journals and other premier conferences and workshops. Existing literature about the Web crawler is classified into different key subareas. Each subarea is further divided according to the techniques being used. We analyzed the distribution of various articles using multiple criteria and depicted conclusions. Various studies that use open source Web crawlers are also reported. We have highlighted future areas of research. We call for an increased awareness in various fields of the Web crawler and identify how techniques from other domains can be used for crawling the Web. Limitations and recommendations for future are also discussed.

How to cite this article:
WIREs Data Mining Knowl Discov 2017, e1218. doi: 10.1002/widm.1218

INTRODUCTION AND MOTIVATION

The search engine has become a vital part of our digital life. In the ocean of Web, finding information is like finding a needle in the haystack. The search engine is used to find information on the Web. Search engines can be of two types—crawler based and human powered. The human powered search engine indexes a collection of high-quality user submitted or handpicked websites. The webmaster or reviewer of a website submits a short description of his site that builds the search base. In a human powered search engine, results or at least position of results is affected by human intervention. On the other hand, crawler-based search engines do not have this problem. Crawler-based search engine have three main components: crawler, indexer, and searching-ranking algorithm. A Web crawler is the heart of any crawler-based search engine. It keeps on traversing webpages on the Web to gather information that can be indexed by an indexer to handle any user query efficiently. Searching ranking algorithm returns those webpages that are best matched and ordered in response to respective user query. Mostly a user analyzes first few results in response to the query he has submitted. So, it becomes necessary to order the results efficiently.

The main focus of our paper is Web crawler. A Web crawler is a part of search engine that gathers information from the Web so that indexer can create an index of the data. Web crawler starts the process of crawling from a single uniform resource locator (URL) or a set of seed URLs. As a crawler visits a URL, it adds all hyperlinks in the webpage to a list of URLs to be visited further. The objective of crawling is to collect as many useful webpages as possible in the least possible time. Crawler prioritizes the order in which the URLs' are visited due to large collection of data on the web. Huge size and a variety of the Web make it difficult for any crawler to

*Correspondence to: manishkamboj3@gmail.com

[1]Department of Computer Science and Engineering, PEC University of Technology, Chandigarh, India

[2]Department of Computer Engineering, Punjabi University, Patiala, India

Conflict of interest: The authors have declared no conflicts of interest for this article.

retrieve all relevant data from the Web. Thus, various variants of Web crawler have emerged as an active research area.

In general, crawl means to move in one direction slowly. Technically, Web crawlers are the tools for data acquisition in the search engines. These are also called as spiders or robots or wanderers. A mere basic Web crawler is a function with a set of seed URLs as input and a set of crawled webpages as output. This simple function takes URL one by one, gets the webpage, and adds URLs found on this webpage to the list of URLs to be visited further. Brin and Page[1] is the most frequently cited in the literature, discuss the general basic architecture of the Web crawler and the various data structures that can be used. As shown in Figure 1, URL to visit and URL visited so far are maintained to keep track of the webpages visited so far. Single URL is chosen from the list of URLs and the corresponding webpage is downloaded at the local site. From the downloaded webpage, URLs are extracted and added to the list of *URLs to visit*. This basic architecture can be modified according to the application needs of crawling. In the architecture, more components can be added, or existing components can be combined as per the requirements.

A bare minimum crawler needs at least these components: A set of seed URLs for starting the crawling process, downloader to download webpages from the Web to the local repository, URL extractor to get the URLs from webpages. This process is repeated recursively until the list of URLs to be visited is empty.

When we started working on the Web crawler, we found that the information is scattered over in various sources. There exists no single source with comprehensive information till date. A summarized report will help the researchers to identify studies and carry forward the research in a particular direction. To carry out the process, we followed the guidelines of Kitchenham and Charters,[2] Budgen and Brereton[3] and Brereton et al.[4] These studies define systematic literature review as a tool for identifying, evaluating, and interpreting all available research relevant to a particular topic or phenomenon. The systematic review study carried out by Rattan et al.[5] is used as a general reference. This research article can act as the base for any researcher who is interested in the area of Web crawler or related area.

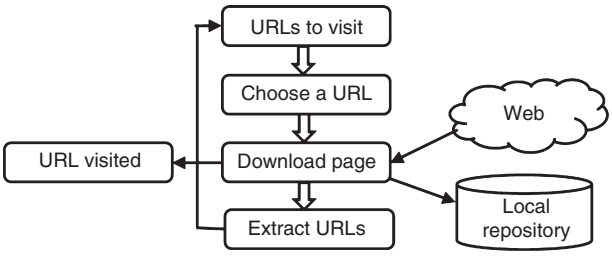

FIGURE 1 | Architecture of a Web crawler.

Motivation for Work

- A Web crawler is the main part of any search engine that finds data that can be indexed by a search engine. Our study detects various shortcomings and strategies for crawling the Web.
- Our study tries to explore various technologies used for Web crawling and will demonstrate their comparative analysis. Furthermore, we will discuss various subject systems and performance metrics used by different studies.
- When we started working on the Web crawler, a lack of a complete systematic literature review was a motivating factors. We analyzed the entire existing database for Web crawler and summarized it to report research gaps for further investigation.

The remainder of this paper is organized as follows: *Background* section gives background, general terms, policies, and challenges in the area of a Web crawler. *Current Status of Web Crawler* section discusses the results of our systematic review. In next section, various performance metrics for Web crawler are presented. Fifth section discusses several research gaps that are identified from existing work, it also includes various future avenues for carrying out research in the area of a Web crawler. Final section concludes and provides recommendations for future work.

BACKGROUND

In this section, we will discuss various terminology related to the Web crawler. We present a taxonomy of Web crawlers and summarize different types of crawler.

Types of Web Crawler

There is no standard taxonomy for the Web crawlers available in the literature. Various researchers have categorized Web crawler in their own way. Figure 2 gives a broad taxonomy of Web crawler.

We list here basic types of Web crawler:

Universal or Broad crawler: These type of Web crawlers are not limited to webpages of a particular

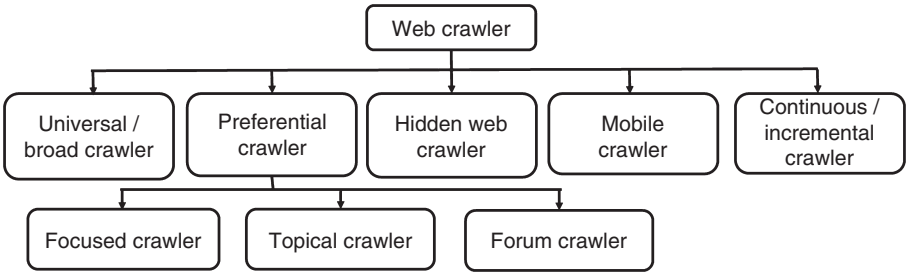

FIGURE 2 | A taxonomy of Web crawler.

topic or domain. They keep on following links endlessly and get all webpages they encounter.

Preferential crawler: This category of Web crawler does not crawl all links they encounter rather the user submits a condition or topic of interest that guides the preferential crawler. Furthermore, the preferential crawler can be categorized as focused and topical crawler. Chakrabarti et al.[6] proposed one of the first focused crawler that selectively seeks out webpages that are relevant to a predefined set of topics.

Topical crawler or topic-specific crawler is used for searching information related to some specific topic from the Web. Topical crawling assumes that only the topic of interest is specified while focused crawling assumes that some labeled examples of relevant and nonrelevant webpages are also available.[7] The other category of forum crawler only deals with crawling of online forum content.

Hidden Web crawler: A significant amount of information on the Web cannot be accessed directly by following the hyperlinks on webpages. This information is hidden behind search or query interface, this part of the Web is called hidden Web or deep Web.[8] A special category of crawlers called hidden Web crawler deals with crawling this section of the Web.

Mobile crawler: It is a method of crawling in which selection and filtration of webpages are performed on server side rather than on the search engine side. Moving code to data in mobile crawling reduces network load caused by traditional Web crawler.[9]

Continuous or Incremental crawler: The Web is dynamic and data on the webpages keep on changing frequently. These crawlers are used to maintain the index database of the search engine up-to-date.[10] However, there is a trade-off between managing freshness and resources consumption.

Web Crawling Policies

Any Web crawler has the potential to disrupt the services of a server. Koster[11] gives a set of policies every Web crawler must follow.

Politeness policy: A crawler should not hamper any website with the requests. Every crawler is expected to respect Robots.txt and should crawl only allowed webpages. Crawler should act as a 'good citizen' of the Web world.

Parallelization policy: Multiple threads of a crawler are used to maximize download rate of webpages. A policy is required for assigning new URLs discovered during crawling process to different threads running in parallel.

Revisit policy: To keep an index of a search engine up-to-date, a Web crawler needs to revisit webpages. Either a uniform revisit policy or a proportional revisit policy can be used for deciding when to revisit a webpage.

Robustness policy: The Web is hosted by servers that may mislead a crawler and get it stuck into fetching a huge number of webpages of a particular domain. However, not all the traps are malicious some are due to side effects of faulty website designs. A crawler must be immune to malicious behavior of any Web server.

Challenges in Crawling the Web

Regardless of the category of a Web crawler, the researchers when dealing with Web crawlers face some challenges. Some of the challenges are listed here.

- *Nonuniform structures*: The Web is dynamic and uses inconsistent data structures, as there is no universal norm to build a website. Due to lack of uniformity, collecting data becomes difficult. The problem is amplified when crawler has to deal with semistructured and unstructured data.[12]
- *Scale and revisit*: Size of the Web cannot be measured. Furthermore, there is a trade-off between coverage and maintaining freshness of a search engine database. The goal of any Web crawler must be to ensure coverage of all

reasonable content while bypassing low quality and irrelevant content.[13]

- *Crawling multimedia*: A crawler can easily analyze text but analyzing multimedia is an open challenge. Analyzing multimedia content on webpages to detect criminal activities is one of the prominent applications these days.[14]
- *Crawling deep Web*: A large part of the Web is hidden behind search interfaces and forms. This part of the Web that cannot be reached directly comprises hidden Web or deep Web. Hidden Web is accessible by querying the database, but query selection is another challenge.[15]

CURRENT STATUS OF WEB CRAWLER

Some papers in the literature give general information regarding a Web crawler. A generic search engine architecture including indexing, link analysis, and webpage storage is discussed in Refs 1,16. These papers are milestones in this field with maximum citations. Ref 16 discusses the general design and implementation of each component of a Web crawler in detail. In this section, we will discuss various categories of a Web crawler, based on the taxonomy shown in Figure 2. Various studies have been categorized into focused crawler, topical crawler, hidden Web crawler, mobile crawler, and forum crawler. There are a large number of techniques and tools for each category that can be used to classify the web crawlers further. Table 1 shows the categorization along with the count of studies in each category. No thick line can be drawn between different categories. Few papers hybridize more than one technique so fall under more than one category. Forum crawler is different from focused crawler as it specializes in retrieving information about user and user generated contents.[254]

Focused Crawler Techniques

Focused crawler technique gives priority to those URLs in the process of crawling, in which probability of finding information of user's interest is high. It is evident from Table 1 that most of the studies present in literature are related to the focused crawler. Main categories of crawler given in Table 1 are further classified according to technique or tool they have used. This section discusses and categorizes various studies of a focused crawler. The subcategories as shown in Table 2 are not stringent and some papers may fall into more than one subcategories. Some of the papers are general studies of focused crawler and do not fit into any of the subcategories of Table 2. Techniques used by the focused crawler, topical crawler, and forum crawler are somewhat same, and hence a common table is constructed for them.

Focused Web Crawler Using Soft Computing Techniques

Focused Web crawler using soft computing techniques uses the real-world phenomena such as memberships, grouping, classification of factors, and so forth for deciding relevancy of a webpage. These techniques do not use strict mathematical definitions and are tolerant to imprecision, uncertainty, and partial truth to achieve a solution. In the starting paragraph of each category, we will discuss common characteristics of that technique and their subsequent

TABLE 1 | Number of Studies Referring to Different Categories of Web Crawler

Sr. No.	Category of Web Crawler	#	Citations
1.	Focused crawler	149	6, 7, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163
2.	Hidden Web crawler	48	8, 15, 84, 85, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207
3.	Topical crawler	35	129, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241
4.	Mobile crawler	13	9, 10, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252
5.	Forum crawler	3	76, 253, 254

TABLE 2 | Subcategories of Focused, Forum, and Topical Crawler

Sr. No.	Category	Code	#	Citation
1.	Focused crawler using soft computing techniques	C1	30	29, 42, 46, 47, 49, 50, 53, 54, 59, 66, 72, 77, 78, 86, 87, 89, 90, 100, 101, 104, 121, 122, 125, 145, 145, 146, 152, 153, 156, 160
2.	Application-based focused crawler	C2	23	18, 20, 46, 50, 61, 65, 69, 77, 78, 81, 85, 98, 102, 104, 111, 117, 137, 140, 141, 151, 157, 158, 173
3.	Focused crawler based on link, text and URL	C3	22	6, 27, 38, 40, 41, 43, 72, 76, 83, 88, 93, 126, 130, 154, 210, 217, 222, 223, 228, 231, 241, 254
4.	Focused crawler based on context, graph, decision tree, and DOM	C4	19	27, 37, 39, 52, 56, 59, 62, 82, 105, 106, 114, 120, 123, 129, 135, 142, 144, 159, 233
5.	Semantic crawling-based focused crawler	C5	13	30, 32, 36, 48, 64, 68, 74, 113, 125, 128, 147, 152, 235
6.	Learnable focused crawler	C6	11	60, 71, 79, 103, 115, 139, 150, 209, 226, 227, 234
7.	Topic specific focused crawler	C7	9	212, 213, 215, 220, 221, 230, 236, 238, 239
8.	Parallel and distributed focused crawler	C8	8	19, 22, 33, 35, 63, 118, 119, 155
9.	Language classification-based focused crawler	C9	7	17, 28, 31, 44, 99, 109, 112
10.	Tf-idf- and rule-based focused crawler	C10	7	24, 64, 80, 162, 106, 107, 134
11.	Vertical search engine based on focused crawler	C11	6	23, 51, 75, 136, 155, 253
12.	Query, keyword, and metadata-based focused crawler	C12	4	25, 108, 127, 148
13.	Location- and geographical-based focused crawler	C13	2	21, 58
14.	Incremental and revisit policy-based focused crawler	C14	2	92, 138

use in various studies. Various features of a webpage and their interrelationship are used by various ontology-based focused Web crawlers. Classification of webpages is an important aspect of focused crawling for deciding the relevance of a webpage to be crawled. Artificial neural network (ANN) can be trained to learn characteristics of a webpage to decide its relevancy for a focused Web crawler. Many techniques in literature have used ANN as a filter to classify webpages as relevant or irrelevant. Hidden Markov model (HMM) learns from previously crawled webpage sequences to decide the relevancy of a newly visited webpage. Formal concept analysis (FCA) is used to describe a relationship between the webpage relevancy and a set of attributes from those webpages. FCA forms clusters of concepts such as 'page related to social networking' and attributes that are to be analyzed for this class of webpages. In next paragraphs, we will discuss how these characteristics are used by various studies in literature.

Ontologies are used for evaluating the importance of a Web document of interest using network structure which is updated periodically.[42] As the connected ontologies are updated, the crawling process becomes more focused. Semi-supervised ontology learning-based focused (SOF) crawler[47] is proposed that determine the text similarity between concept description of a concept and description of a webpage. It uses metadata in a webpage to classify webpages as relevant or irrelevant in accordance to the topic of interest specified to Web crawler. Dong et al.[46,49,50] used semantic focused crawling for classifying industrial and digital health ecosystems. Crawler discovers service information from the Web and classifies it based on specific service domain knowledge. Barros et al.[29] developed foxset using fuzzy theory for collecting and evaluating documents of a dataset. The documents in the dataset are collected by using a crawler that uses metadata of webpages to decide its relevance through a breadth-first search.

The ontology-based approaches require manually assigned concept weight that remains constant during crawling and results in a poor harvest ratio.[6] Zheng et al.[152] proposed the use of ANN to determine the relativeness between a webpage and a given topic. They provided supervised training based on ANN that uses labeled webpages. Unsupervised training techniques have also been proposed in the literature. Su et al.[125] state that ontologies can evolve automatically during crawling.

A novel approach based on HMM is suggested by Liu et al.[87] for predicting links leading to relevant

webpages. Data from a user browser session is used to form a link structure among the webpages to learn the sequences that can be followed to reach webpages of interest. Gcrawler based on genetic algorithm is proposed by Shokouhi et al.,[122] which keeps on expanding its initial keywords during the crawling process. Refs 66,86 uses similar HMM-based focused crawling techniques. More emphasis is given on three categories of crawlers, that is, best first crawlers, semantic crawlers, and learning crawlers in these studies.

Du et al.[53] suggested FCA for constructing user interest ontology. A cosine similarity feature vector is used to calculate the similarity between webpages. FCA is combined with concept context graph (CCG) by Gao et al.[59] They do not analyze the links on a webpage, only the content on a webpage is taken into consideration. Yang et al.[121] proposed a Java-based ontocrawler for scholar domain. The proposed technique uses users query to decide the ontology. Therefore, a large number of unrelated webpages are also returned in the result. Ref 73 deals with a system having data in a tabular form using ontology-based system. Luong et al.[89] presented an ontology-learning framework for the biological domain. DMatch ontology matcher[90] is proposed which can match available semantic Web documents to the topic ontology. Yang[145] stressed the use of ontology and website model for webpage comparison.

Some genetic focused crawlers are also proposed in literature. Ning et al.[100] optimized the crossover, selection, and mutation operator for a new focused crawler analysis model. Ozel[101] combined genetic algorithms along with HTML tags for improving performance. The system performance is improved upto 95% by combining various features. Zheng[153] uses ant algorithm and Ref 54 uses niche genetic algorithm to improve performance of focused crawler.

Application-Based Focused Crawler

Crawlers that target a particular group of users come under this category. Some focused crawlers in the literature are proposed for doing a particular task or application. These application-based focused crawlers cover a broad domain ranging from crawling medical data for sentiments analysis to crawling education resources. Most of the work in this category deals with crawling the educational and scholarly data from the Web. Only open corpus sources can be targeted for crawling data present behind the login page comes under the category of hidden Web. Amongst various applications, we will first discuss the application of crawler for security. Agarwal and Sureka[20] gave a method for identification of malicious videos on YouTube. The best-fit crawling strategy is used in which if a node is relevant and has the highest priority amongst all nodes then that node will be crawled first. Specialized crawler on the topic of security (SCS) is developed in Ref 102 that uses ANN. SCS uses IP and domain name tracking for crawling. The proposed crawler indexes and follows the updates of webpages that can be a threat to security.

Cho et al.[173] discussed the design and performance of WebBase, a tool for the Web research. It allows researchers to retrieve webpages from the WebBase and stream them on Internet. Authors discuss various trade-offs and basic design for the Web crawler. A similar focused crawler to build a digital library for the scientific community is developed in Ref 111. They used meta-search enhanced focused crawling to avoid the problem of traditional focused crawler being trapped within a limited subgraph. Meta-searching keeps drawing queries from a domain-specific lexicon, retrieving URLs by querying multiple search engines and combining the top results.

The approaches discussed so far do not take into consideration the big data. A map reduce-based crawl-extraction-ingestion (CEI) workflow-based crawler is proposed in Ref 137. They discussed various challenges, lessons, and opportunities in building a scholarly big data platform. They give architecture for data crawling, information extraction, and analytics. NetSifter[61] uses a combination of webpage level analytics and heuristics to take advantage of both focused crawler and full Web crawling. A Web crawler for face stock information of financial field is proposed in Ref 104. Abbasi et al.[18] carried out focused crawling for online medical sentiments from Web 2.0. The proposed approach is validated based on precision and recall values.

Hijazi and Itmazi[65] proposed a crawler for open educational resources (OER). The administrator feeds the server with seed URLs and keywords based on the courses of user interest. The crawler visits the website and downloads webpages and material having keywords related to the course. A similar open corpus content service (OCCS)[81] enables the dynamic discovery, harvesting, and delivery of educational content from open corpus sources. It discovers, classifies, and indexes the data. Huang et al.[69] gave a novel approach for building an intelligent focused crawler for retrieving E-commerce information. For the users who are interested in keeping track of a developing story, a beehive model is suggested in Ref 97. The story can even be reconstructed backward in

time. The technique uses bee model, in which bees are either scouts or recruits bee. A novel profile-based focused crawling system for social profiles is proposed in Ref 151 that does not require any privileged access to internal private databases of such websites. The user's profile is treated as additional ranking criteria for guiding the crawling. The experimental result shows that the proposed crawler is better in terms of harvest ratio and robustness.

In today's world, a large amount of data are available online. Technology intelligence can play a vital role in handling this continuously increasing data. Technology intelligence can be considered as a component of technology management that can monitor data and respond to new developments just in time. Ref 117 proposes a crawler which can make the process of technology intelligence efficient. A subject-oriented Web crawler that is distributed and modular is proposed in Ref 140. It also allows construction of subject-oriented search engine. Xu et al.[141] gave design and implementation of a focused crawler for software components. Zhuang et al.[157] proposed focused crawling for missing documents in digital libraries for any given publication venue. A home aggregator is used that saves homepage of the author and uses it as seed URL. A crawler keeping in mind the requirements of law enforcement agencies and intelligence analyst is developed in Ref 158. The crawler engine refines progressively according to the user binary feedback, i.e., yes/no.

Focused Crawler Based on Link, Text, and URL

The webpage corresponding to a seed URL provides the links that can be used as the gateway to the topic of interest. This category of focused crawler uses text on the webpage and in the URL for deciding the relevancy of webpage. The user specifies the topic of interest not by mentioning some keywords but using some exemplary webpages. URL-based crawling is preferable because of two reasons as (1) it decides whether a webpage is of interest or not before downloading it and (2) it is beneficial when the resources are crucial.

Chakrabarti et al.[6] introduced the concept of focused crawling using classifier and distiller in the year 1999. Classifier decides the relevancy of webpage with respect to a given topic, and distiller identifies the URLs that need to be further explored to reach the webpage of interest. After this, the field of focused crawler evolved at a fast rate. Webpages that are one link away are semantically highly related to the topic of interest than those webpages that are many links away.[27] They use link distance instead of probability or relevance score to rank webpages.

Concordia Indexing and DIscovering system (CINDI)[37] uses revised context graph and multilevel inspection scheme to discover relevant webpages. It explores relevant resources that are many links away from seed URLs. This paper also falls in the next subcategory of focused crawler based on context graph. CINDI is used to find computer science and software engineering academic documents. To increase overall recall of the crawler, proposed technique also takes into consideration irrelevant webpages that may lead to relevant webpages.

Traditional focused crawlers consider full webpage for guiding the crawling, but instead of considering all the data on a webpage, it can be partitioned into different blocks. Li et al.[83] use relevance prediction of candidate URLs based on block partition of the webpage, anchor text, and block text. A method for partitioning the webpage based on data of the webpage is discussed in Ref 88. Intelligent crawler (Icrawler) developed by Uzun et al.[130] uses content extractor in which webpages are first divided into blocks and then various contents such as links, text, menus that are extracted from each block. Similar crawlers that partition webpages into blocks are proposed in Refs 88,127 Pant et al.[222] use context of the links i.e., text that appears around a hyperlink within a webpage for guiding the crawler. They compared the proposed crawler with the text window-based crawler using harvest ratio and target recall.

Focused Crawler Based on Context Graph, Decision Tree, and Document Object Model

This category of focused crawler visualizes Web as a graph or decision tree in which each webpage represents a node. Furthermore, nodes of similar interest are connected to each other. A taxonomy of topics can be constructed in which webpages are arranged in a layered fashion according to the topic they belong. The context graph is very helpful for any crawler as it gives a clear picture of the categories that are directly or indirectly related to the topic of interest.[159] A separate context graph is built for every seed URL given by the user using backward crawling. After constructing context graph for all the seed URLs, corresponding layers from different context graphs are combined yielding a merged context graph. Du et al.[52] introduced a novel method for topic-specific Web crawling approach using context graph. New concepts are added and old concepts are deleted from the CCG. The proposed technique includes three main steps: mine the core concept, construct context graph, and find the appropriate

position. The proposed method is highly dependent on the word units that are chosen and topic of crawling. A word unit is a collection of words that are semantically similar.

Chen et al.[39] introduced an algorithm for mapping keywords and documents of a language to a hierarchical topic taxonomy. They proposed relevance prediction based on hierarchal context information (RPHCI) of the taxonomy. Fu et al.[56] proposed a novel graph-based sentiment (GBS) crawler that combines topic and sentiment information about a particular topic. It also takes into consideration the tunneling strategy that allows focused crawler to traverse irrelevant webpages to reach to the relevant one. Gao et al.[59] suggested incrementally updating CCG based on FCA. An unvisited webpage is taken as a concept and based on attributes of the concept, it is inserted at the suitable layer.

Li et al.[82] discussed the construction of a decision tree based on anchor text of hyperlinks. The decision tree constructed is a boolean function. A heuristic-based approach content block partition-selective link context (CBP-SLC) is presented in Ref 105. In this technique, a webpage is partitioned into smaller content blocks based on document object model (DOM). A service class description (SCD)-based topic-specific crawler DynaBot is presented in Ref 114. SCD uses a triplet < T, G, P > where T denotes a set of type definitions, G denotes control flow graph, and P denotes a set of probing templates. A Web crawler to discover the Web services relevant to a service class of interest uses SCD. Xcrawl[120] combines the searching and crawling procedure for focused crawling. It uses query probing that depends on a lexical database. The proposed system has a significant increase in recall while maintaining precision.

Automatic topical crawler (AuTo Crawler)[129] takes into consideration the user interest specification to identify target examples. A user can specify the topic either using a Web directory or by specifying keywords. The disadvantage of the crawler is that it can only handle pure HTML text. Yang et al.[144] discussed the problem of tunneling when an off-topic document is linked to a highly relevant document. This issue is due to the lack of structured information in a document. They use document segmentation to identify useful segments present around a hyperlink. The classifier treats the document as small subdocuments instead of a single document. Training webpages and parent URL are arranged in a layered graph structure that is backtracked to extract features to be used by the Web crawler. Yang et al.[233] used DOM tree in which relationship is maintained between different paragraphs of a webpage. Vector space model (VSM) is used to calculate the similarity between webpages.

Semantic Crawling-Based Focused Crawler

As the Web has evolved from traditional HTML to modern Web, i.e., Web 2.0, the future of the Web is semantic Web.[32] Semantic Web can be considered as an extension of today's Web in which information has a well-defined meaning expressed using resource description framework (RDF). Focused crawlers in this category exploit the semantics of the Web content and use some ontology heuristics. A semantic focused crawler (SFC) computes the relevancy of a webpage by using domain knowledge related to the search topic. Semantic Web arranges all data in the form of logically linked data instead of traditional hyperlinked Web. It is highly unlikely that the linking scheme that is used traditionally will make any sense in semantic Web.[30] In the section *Focused Web Crawler Using Soft Computing Techniques*, some of the semantic crawling-based papers were also discussed as they were more into soft computing domain of focused crawling.

Biocrawler[30] is the starting point in the field of semantic Web crawler that uses a combination of semantically enhanced content and focused crawling. In this technique, each crawler starts from a random webpage and sends the acquired content back to search engine and its peer crawlers. A multithreaded SFC[32] for educational learning content is presented that starts with some basic concepts and expand them based on domain ontologies. For expanding, it computes dynamic semantic relevance by fetching top priority webpages. Q-learning-based link prediction (QBLP)[36] algorithm has the capability of guessing the best candidate webpages to be followed for crawling. It also links the webpages to the relevant topic and linking topics of relevance. Crawler uses the experience gained during crawling to adjust crawling strategy to find best relevant webpages. QBLP also uses Q-learner and Bayes classifier.

A novel self-adaptive semantic focused crawler (SASF)[48] introduced by Dong and Hussain uses unsupervised ontology-based learning. The Web crawler downloads a few webpages and uses the statistical data of these webpages to determine semantic relevance between a service description and concept description of a concept. They address three main issues of mining service information: heterogeneity, ubiquity, and ambiguity. Su et al.[125] discussed another unsupervised ontology learning-based focused crawler to compute relevance score between topics and Web documents. The weight of each

concept C_k is $W_{C_k}^0 = 1.00 \times n^{d(C_K,\ T)}$, where n is fixed discount factor and $d(C_K,\ T)$ denotes the distance between the topic concept T and C_K.

Hao et al.[64] proposed a crawler based on latent semantic index (LSI) combined with term frequency-inverse document frequency (Tf-idf) for hyperlinked topics. Tf-idf scheme gives the weight for the term i in a document as $W_i = tf_i * \log\left({}^{N}/_{df_i}\right)$, where tf_i is the frequency of term i in the document, N is the total number of document and df_i is the number of document containing term i. They compare Tf-idf- and LSI-based Web crawler and fuses their advantages to propose a combined algorithm. The proposed technique also uses webpage relevance topic prediction. Jung[74] uses local knowledge indexing in which multiple crawlers collaborate for semantic instances from the knowledge encoded information system. Thukral et al.[128] proposed focused crawling using human cognition (FCHC) for social bookmarking websites. The crawler uses tags and bookmarks tagged by the users with semantically relevant tags for guiding the crawling process. Ying et al.[147] uses breadcrumb navigation that divides the Web into the different semantic forest and then searches the forests to find the subtrees relevant to a given topic. Webpages are analyzed by a distiller to remove errors and advertisement webpages.

Learnable Focused Crawler

The major task of a focused crawler is to decide whether a webpage is relevant or not. A set of continuously updating knowledge data needs to be maintained for learnable focused crawler. The crawler may use online catalogues like Dmoz.org to acquire knowledge from online taxonomies. Training process can be supervised, unsupervised, or any other learning paradigm. The classifier can be trained in a supervised way that requires a set of labeled documents for its training. Naïve Bayes, SVM are some of the popular classifiers used in literature. Gautam and Padmini[103] claim that Naïve Bayes is a weak choice as compared to neural network or SVM for a learnable focused crawler. They used supervised learning to train the crawling classifier. Rungsawang and Angkawattanawit[227] proposed an automatic learnable topic crawler. They used the concept of hub and authority webpages where authority webpages are considered as a good set of seed URLs. The proposed crawler is trained in the first stage to guide the crawling process further.

Aggarwal et al.[209] proposed a self-learning crawler that uses in-linking between webpages. The statistical model is used to maintain a dynamically updated set of statistical information that is learned during crawling process. This paper discusses content-based learning, URL token-based learning, link-based learning, sibling-based learning, and combining these features. Authors showed that the crawler using learning information reached to a harvest ratio of 42% as compared to a harvest ratio of 28% of a normal crawler.

Bayesian object-based crawler (BOB crawler)[60] uses supervised learning based on keywords present in the URL and title of the webpage. These keywords are used for comparing the similarity between the webpage and focused crawler topic. Webpages are treated as objects having a set of features that can be used for predicting the relevancy. Zhang and Lu[234] proposed a semi-supervised topical web crawler (SCTWC), using a Q-function that gives a mapping from a set of words to a reward value. The output values of different Q-functions are mapped to separate classes. Automatic selection of seed URLs and time costing problems need to be tackled before SCTWC can be a real success.

An incremental learning strategy based on SVM and Naïve Bayes is the basis for Whunter focused crawler.[71] The proposed crawler starts with a few sample webpages and more knowledge is integrated about the topic with time. Qian et al.[226] introduced reinforcement learning to find relevance between a focused topic and a candidate webpage. Safran et al.[115] use URL text, anchor text, parent webpage, and text around the link for predicting the relevancy of a webpage. They used a Naïve Bayes learnable crawler.

Topic Specific Focused Crawler

Topic-specific crawling is a method that crawls webpages according to the user interest. Crawlers in this category are used for building the repository of webpages according to given topic. A topic similarity is calculated to determine the best-fit category for a webpage. The webpages are crawled according to their relevance probability. Begmark[239] uses Mercator crawler for building a document collection on various topics in science, mathematics, technology, and engineering for a digital library.

Chung and Clarke[212] partitioned the Web into general subject areas with a crawler assigned to each. Collaboration within crawler is required so that the URLs to be visited can be ordered. However, the proposed approach has the disadvantage that multiple crawlers may independently encounter the same URL. Davison[213] states that webpages with the same content are typically linked in terms of textual content with each other. Hence, this concept is useful for

indexer and search engines. Peng et al.[224] proposed to use CCG based on FCA to give the order in which webpages should be visited according to the user interest. The CCG is constructed based on the relationship between webpages using links.

Mukherjea[220] presented web topic management system (WTMS) for collection and analysis of information on the Web related to a particular topic. The proposed approach groups webpages according to their physical domains. The information can be visualized at various levels i.e., abstraction-site level view and webpage level view. Noh et al.[221] use term frequency/document frequency, entropy, and compile rules for computing relevance of a webpage to a topic. The degree of relevance of a webpage R_i is defined as $R_i = (1 - \rho)\ \lambda_i / |K| + \rho\ R_j$ where, R_j is the degree of relevance of a webpage j having URL of webpage i which is already crawled, ρ is a constant, λ_i is the number of keywords in the webpage i, $|K|$ is the cardinality of predefined key phrases. The proposed technique has an accuracy of 97.8%.

Zong et al.[238] discussed a new probability hyperlink-induced topic search (P-HITS) algorithm for topic-specific Web crawler. To determine similarity to the topic, a similarity score is computed using vector space approach. HITS algorithm uses hubs and authority webpages. In P-HITS, the probability is introduced to select URLs based on HITS.

Parallel and Distributed Focused Crawler

Due to large size of the Web, a single process centralized crawler may slow down the crawling process. A parallel crawler is a multiprocessor crawler that divides the Web into segments and each segment is assigned to one of the parallel agents. Scalability, network-load dispersion and network-load reduction are some of the advantages of parallel and distributed crawling approach. Selecting appropriate strategy to divide the Web is the main issue in parallel crawlers.[22] Achsan and Wibowo[19] proposed to run hundreds of thread from a crawler and distribute those using publically available proxy servers. This approach will save the crawler from being banned by server, as crawling may be confused with a cyber-attack. Out of transparent, anonymous and elite, a Web crawler can use any type of proxy server. Bosnjak et al. proposed TwitterEcho,[33] a crawler to continuously collect data from a particular user community like Twitter. The proposed crawler claims to be continuous, modular, and fault tolerant.

A master slave parallel architecture for a Web crawler, based on ontology and DOM is proposed in Ref 35. Master coordinates each slave using a shared memory and each slave is concerned about a concept extracted from the ontology. Selamat and Abkenari[118] raised the issue that existing parallel crawlers use link dependent metrics e.g., backlink, pagerank, and so on for the ordering of URLs which causes an overhead in exchange of information. A general parallel Web crawler is discussed in Ref 163. Overlap, coverage, communication overhead, and quality are the metrics that can be used for the evaluation of parallel crawlers. URL hash based-, site hash based-, and hierarchical-based classification method can be used for partitioning the Web. This paper presents some guidelines for implementing any parallel crawling strategy.

Language Classification-Based Focused Crawler

Nowadays, we are interested in speech recognition, word recognition, and textual content on the Web to create a language model of a particular language. The crawlers in this category are used for collecting language- and topic-specific corpora using focused crawling strategy. A country-level domain can be used to identify the language of a particular URL. Barbosa and Bangalore[28] proposed a crawling strategy to build a corpus from which diverse language models are generated. Crawler is incremental in the sense that it tries to fill the gaps present in current language model that is constructed from previous cycles. Baykan et al.[31] gives a relevant answer to an interesting question i.e., if only the URL of a webpage is given, can we identify its language? They use a machine-learning approach that classifies URL according to the country code top-level domain (ccTLD). They check the official language of ccTLD's country and assign the corresponding language to the URL. They also compared the performance of various URL-based language classifiers.

A lightweight Bangla stemmer is used for Bangla News classification using Naïve Bayes classifier.[44] It uses News code taxonomies published by International Press Telecommunication Council (IPTC). In this project, SVM classifier is not used as it gives excellent precision with the poor recall. The News websites are crawled on breadth first search method to collect the News webpages. The News article contents are extracted from News-related webpages to generate full-text-rich site summary (RSS) file. Nhan et al.[99] proposed a similar method based on genetic algorithm to crawl Vietnamese webpages. Putra and Akbar[109] proposed a modification of WebSPHIX for Javanese and Sundanese corpus construction. They studied dictionary breadth first, dictionary by page link, n-gram depth first, and n-gram random. They concluded that combination of dictionary algorithm

and breadth first methods delivers the highest performance as compared to other combinations.

Radu and Rebedea[112] gave architecture of a system for Romanian vocabulary. Their focus is on discovering new potential words that have entered the Romanian lexicon. They also used Scrappy framework for extracting Web content based on several user-defined rules. Rungsawang[17] proposed a crawler to collect webpages that are related to Thailand and written in any foreign language say English. Instead of considering full website or webpage, they considered only Web segments that may have a high probability of linking to relevant webpages. They have not built a single dictionary for all the topics but have one specific dictionary for each Thai-related topical crawler.

Tf-idf and Rule-Based Focused Crawler

Tf-idf is a measure to reflect the importance of a word in a document for a collection of documents. Before using tf-idf, stop words are removed and word stemming is performed on all the documents. Rule-based focused crawler uses linkage statistics among topics to guide the crawling process. Crawler is trained with some predefined topic taxonomy to generate the rules with a probability score. Kumar and Vig[80] explain the importance of Tf-idf, based on which term frequency definition semantic (TFDS) score table is constructed that is used for guiding the crawler. Tf-idf combined with LSI is proposed in Ref 64. The technique proposed in Ref 106 partitions multitopic webpage into several single topic context webpages. It uses DOM for structuring webpages. Tf-idf scheme is used to calculate the frequency of words in any content block of a webpage to decide its importance and guide the crawling process. Pesaranghader et al.[107] highlighted a method of improving the multiterm topics focused crawling using term frequency-information content (TF-IC). Information content has a taxonomy of concepts and relationships among them. They discuss various disadvantages of Tf-idf and LSI in the multiterm topic and keyword prioritization. Also, IDF may have inaccurate values due to a small number of windows and overlooking hierarchy of concepts.

Interclass rule-based focused crawling[24] uses linkage statistics among topics for focused crawling. The system uses canonical classes and hierarchy of topics along with a set of document examples. Interclass rules at first sight look similar to the topic-citation matrix. Pappas et al.[162] proposed an agent-based focused crawling in which agents are used for weighing the topics and for calculating genre relevance score of unvisited Web. Wang et al.[134] used Naïve Bayes classifier and an improved tf-idf algorithm to extract content of the webpage and compute its rank. The proposed crawler is compared with BFS and page rank crawler.

Vertical Search Engine Based on Focused Crawler

As data on the Web is increasing, a specialized search engine called as vertical search engines is getting popular. Unlike a normal search engine, vertical search engine results are selected from a smaller group of websites. The vertical search engine present results as products or services. Almpanidis et al.[23] combined text and link analysis for both classifier and distiller to construct a vertical search engine. The crawler of a vertical search engine automatically classifies the crawled webpages into existing categories.

Exsearch,[75] a novel vertical search engine gathers related information from various websites for online barter business. Exsearch has five modules: focused crawler, information extraction module, machine-learning module, indexing module, and retrieval module. It also provides cross language service and personalized search. A Time Sensitive Vertical Search (TSVS) engine[136] is proposed that uses query triggered crawling (QTC) that coordinate crawling by real time queries. The system has four major modules: adaptive queue and cache module, history retrieval module, semantic expand module, and merge/sort module. QTC improves only the latency of the first result; however, the latency of overall system becomes a bit high. Zhou et al.[155] use a crawling template-based periodic strategy for a distributed vertical crawler. This technique is most suitable for template customized vertical crawling. The vertical crawler uses a regular expression to extract structured information. The proposed crawler is meant to extract information from Internet forums.

Query, Keyword, and Metadata-Based Focused Crawler

This category of focused crawler uses metadata of the webpage to guide crawling. Queries keywords from the user are also used for training and relevance feedback. Sandhan[108] is an Indian project for tourism and health domains across 10 major Indian languages. For training set, they prepared three sets of queries i.e., regional queries, nonregional queries, and health queries. To identify the language of a webpage, they used metadata of the webpage and N-gram method. It uses a language and domain-specific focused crawler. Altingovde et al.[25] devised to construct a domain-specific Web portal that uses an information extractor for extracting the data and a

query engine that allows keyword and advanced query on the extracted data.

Tang et al.[127] proposed a relevance feedback focused crawler based on the query by example for medical information. Yuan et al.[148] made an improvement of PageRank for focused crawler to predict the quality of a webpage before downloading them. They used a learned Information Retrieval (IR) style query of weighted single words and words pair. They used T-PageRank that is based on the topical random surfer. Two webpages are said to be equivalent if they have an angle of zero degrees and their cosine value becomes 1. In a random walk, the surfer can take four types of action: jumping to a random webpage with the same topic, jumping to a random webpage and change the topic, following hyperlink and staying on the same topic, and following hyperlink and jumping to any new topic. Every action can be modeled according to the conditional probabilities.

Location and Geographical-Based Focused Crawler

Sometimes, a query submitted by a user is highly dependent on the geographical location of the user. That is the reason geo-aware focused crawler came into the picture. The geospatial information of the webpages is mostly hidden in unstructured data. Given a list of city–state pairs, the goal of geographical focused crawling is to collect the webpages that are relevant to the targeted city–state pair. A collaborative crawler based on open source crawler labrin is proposed in Ref 58. It uses distributed crawling in which every crawler is responsible for crawling a part of the Web. The technique takes into consideration the geographical sensitive nodes. It has been shown that out of various strategies URL based and extended anchor text based have the overall best performances.

Ahlers and Boll[21] gave the design of an adaptive geospatial focused crawler that can identify and retrieve location relevant documents on the Web. They used Bayesian classifiers to order the links for a faster coverage of precise geospatial webpages.

Incremental and Revisit Policy-Based Focused Crawler

Due to dynamic nature of the Web, data collected by a crawler become stale after a time. Therefore, a crawler needs to revisit webpages after a regular interval of time to keep the index updated. Mali and Meshram[92] proposed a novel selection and revisit policy. The proposed technique first checks for the change in the structure of the webpage and after that, the change in the text content and images are checked. To check changes in the text content, code is assigned to all text content and comparison of the code is made with the last saved local copy of webpage. The webpage is recrawled and refreshed if some changes are detected. Initial words of the tags are compared to check any changes in the structure of webpage. To check change in images rescaling-based method is used.

A webpage consists of a template and content. The template needs to be stripped off for getting the content. However, the proposed technique does not work for webpages with multiple content bodies. Xi et al.[138] analyzed that it is not possible for a traditional Web crawler to decide whether a webpage has been modified or newly added, without doing full crawling. They proposed a method of the structured pattern for deciding the updation probability of a webpage.

Figure 3 shows some techniques used by various categories of a focused crawler. Table 2 shows categories of focused crawler that are referred in Figure 3 as CX, where X is a number from 1 to 14. The reference number of the corresponding study is also mentioned in Figure 3. Some studies that belong to more than one category are also shown. Table 3 gives a comparative analysis of various studies related to focused Web crawler along with their strengths and limitations. Categories in Table 3 are picked from Table 2 and performance metrics used by corresponding studies are also presented. Each study tries to answer an issue of focused Web crawler. Those problems are also presented in Table 3.

Hidden Web Crawler Techniques

The part of Web that past login form, search, or query interfaces is called hidden Web. The information in the hidden Web may not be accessible by following the hyperlinks present on a Webpage. The webpages in the hidden Web are assembled as a response to queries submitted through query interfaces on a database. A large amount of information on the Web is hidden behind search interfaces and forms that cannot be indexed by a traditional Web crawler. Crawling hidden Web is an open challenge due to its heterogeneous and dynamic nature.[196] This section discusses the various techniques that are proposed in the literature for crawling the hidden Web. The various studies of hidden Web crawler can be further categorized

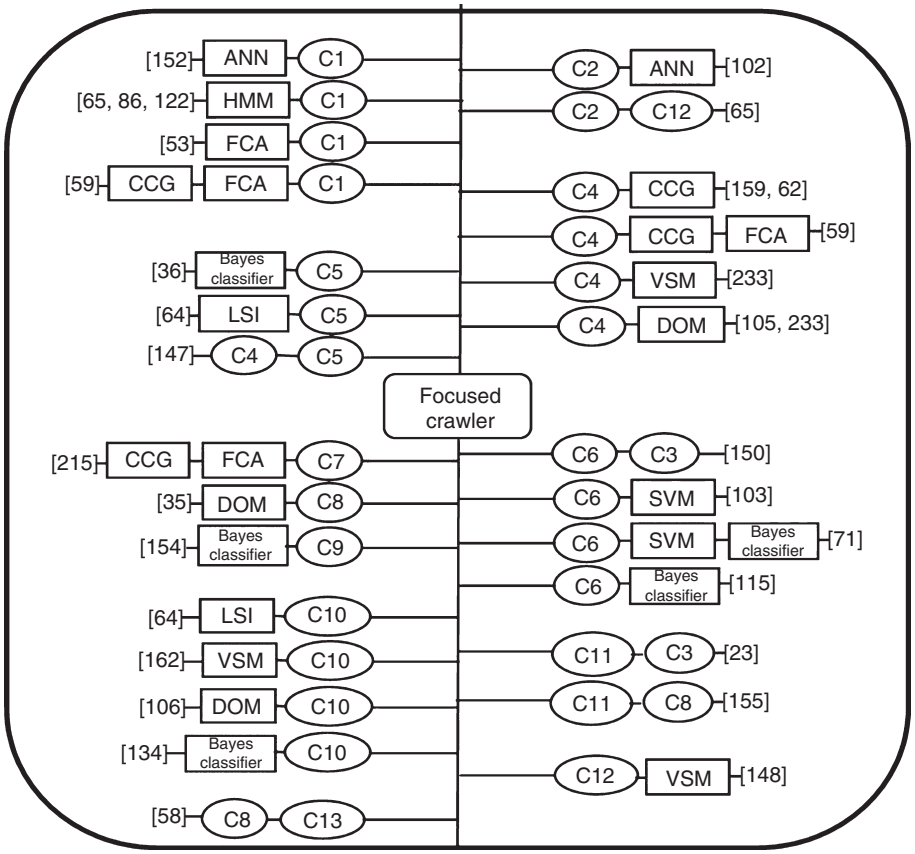

FIGURE 3 | Different techniques used by various categories of focused crawler.

on the basis of techniques and tools used as shown in Table 4. Some studies are general studies of hidden Web and few studies fall into multiple categories due to their hybridized nature.

Keyword Query-Based Approach

Entry point in the hidden Web for a crawler is a query interface. However, after finding query interface, the crawler needs to generate meaningful queries to get data. To build queries, crawler should have a set of keyword that can be combined recursively to build more efficient queries. Some of the techniques focus only on single attribute keyword queries, while some other on multi attributes keyword queries. Ntoulas et al.[204] consider query forms as an entry point to the hidden Web but the challenge is to generate meaningful queries automatically. Their focus is on single attribute keyword queries. Greedy approach is used for making the query efficient in each step. El-Desouky et al.[175] handle the problem of crawling only single valued attribute forms. In the first phase, webpages are collected and then relevant forms are determined and analyzed, to extract the labels. The label matcher is used for generating queries.

Chandramouli and Gauch[172] discussed a cooperative system that explores the information in weblogs and file system. The system determines a potential set of keywords and an efficient query be built to collect webpages. The amount of information gathered by the crawler is reduced in this approach, as information about webpage creation, modification, and deletion is fetched first from the weblogs. Hence, the approach is bandwidth efficient. SmartCrawl[177] is a search engine in which crawler automatically generate queries for getting the webpages. Values to be filled are chosen either while indexing or when the user performs a search. The method used by SmartCrawl consists of finding forms, generating queries, going to results, and searching the index created.

Many websites like Yahoo! directory manually organize the Web accessible databases. QProber[8] does this process automatically using query probes. They have taken into consideration all the searchable Web databases of text documents. They train the classifier using a set of predefined documents and

TABLE 3 | Focused Web Crawler Studies and Comparative Analysis

	Category	Strengths	Limitations	Tool Proposed	Metrics Used	Problem Handled
Chakrabarti et al.[6]	Focused crawler based on link, text and URL	Compatible with Javascript sources	The assumption of Linkage locality and sibling locality is not always true. No support for session mechanism	No	Harvest ratio (H), robustness, relevance	Instead of crawling the full Web, targeting only a portion of Web
Cho et al.[43]	Focused crawler based on link, text and URL	Ordering the URLs to visit important webpages first	The technique is validated only on Stanford website	No	Percentage of webpages crawled	How should a crawler select URLs from its queue of known URLs?
Pant et al.[223]	Focused crawler based on link, text and URL	Lexical and linkage analysis for improving the performance	No support for acronyms in cluster labeling technique	Yes (Panorama)	Harvest ratio	How to bridge the gap between information available on digital libraries and the Web?
Pant et al.[222]	Focused crawler based on link, text and URL	It uses word both near a hyperlink as well as on the entire webpage	The performance may be improved by using phrases and words synonyms	No	Harvest ratio and recall	How can multiple features be combined to form a multilevel inspection crawler?
Diligenti et al.[159]	Focused crawler based on context, graph, decision tree, and DOM	Improved efficiency as compared to traditional crawler	The requirement for reverse links	No	On-topic webpages	Off-topic documents often lead reliably to highly relevant documents
Shchekotykhin[120]	Focused crawler based on context, graph, decision tree, and DOM	Uses combined techniques to identify and exploit navigational structures of website	It does not work with AJAX applications	Yes (Xcrawl)	Recall	How hub and authorities can be used for graph-based focused crawling?
Tsay et al.[129]	Focused crawler based on context, graph, decision tree, and DOM	Taxonomy-based and keyword-based approaches are used to specify user interest. Ordering strategy uses context graph along with several other predictors	It works only with pure HTML text	Yes (AuTo Crawler)	Precision	How to get out of irrelevant webpages?
Chen and Desai[37]	Focused crawler based on context, graph, decision tree, and DOM	Uses Revised context graph to improve recall. Proposed strategy gets rid of the strict link distance requirement	The proposed study is not validated	Yes (CINDI)	-	Constructing a digital library having a collection of online academic and scientific documents

(*continued overleaf*)

TABLE 3 | Continued

	Category	Strengths	Limitations	Tool Proposed	Metrics Used	Problem Handled
Aggarwal et al.[209]	Learnable focused crawler	Intelligent crawling that learns the characteristics of linkage structure of WWW	Linkage characteristics used for WWW are less	No	Harvest ratio	Intelligent crawling that learns the characteristics of linkage structure of WWW
Huang and Ye[71]	Learnable focused crawler	Initially, it uses SVM classifier and once enough web pages are obtained the classifier is switched to naive Bayes	The initial speed of the crawler is slow	Yes (Whunter)	Harvest ratio	Achieving good performance efficiently in limited resources
Chung and Clarke[212]	Topic-specific focused crawler	Uses hash-based technique on the URL to determine the topic of page and assigning it to a particular crawler	Multiple crawlers may independently encounter the same URL	Yes (X4)	Accuracy	How to use collaborative crawling for increasing speed of crawling?
Noh et al.[221]	Topic-specific focused crawler	It computes the degree of relevance of webpages using Tf-idf entropy and compiled rules	Subject system used consist of only 400 webpages	No	Accuracy, efficiency and consistency	Collecting webpages related to a particular topic
Qin et al.[111]	Application-based focused crawler	Uses meta search enhanced focused crawling and handles the tunneling problem effectively	Validation of the proposed technique is domain specific	No	Precision	How to avoid the problem of focused crawler being trapped within a limited subgraph
Abbasi et al.[18]	Application-based focused crawler	It also takes into accounts webpages that are not related directly but by cocitation	-	No	Precision and recall	How to crawl sentiments in medical domain?
Zhang et al.[151]	Application-based focused crawler	Crawling system for social networking websites that uses profiles ranking	The assumption of privileged access to the internal private database is not real	No	Harvest ratio	How to use focused crawling in the case of social media websites?
Liu et al.[86]	Focused crawler using soft computing techniques	User's personalized interests are used for guiding the crawling	The feature set used for HMM is small	no	Precision	Focused crawling based on user behavior

then extract classification rules from the classifier to transform them into queries. The queries are issued to the database and depending on number of matches, database is classified. Ipeirotis et al. in Ref 179 proposed a similar query optimizing text centric approach. To classify a database, they do not inspect webpages or documents in the database, rather they exploit the number of matches for each query probe. Global as view (GAV) and local as view (LAV) are the two approaches discussed for source modeling in Ref 181. Source modeling involves parsing the query interfaces, extracting attributes, and semantic

TABLE 4 | Categories of Hidden Web Crawler

Sr. No.	Category	#	Citation
1.	Keyword query-based approach	10	8, 172, 175, 177, 178, 179, 181, 182, 202, 204
2.	Form-based approach	6	84, 164, 170, 192, 193, 199
3.	Revisit policy and incremental approach	5	172, 184, 186, 205, 206
4.	Attribute and label extraction approach	4	167, 176, 183, 191
5.	Labeled value set-based approach	3	165, 194, 207
6.	Domain- or topic-specific approach	3	188, 195, 201

relationships. In GAV, mapping associates a query to each element of global schema whereas in LAV the mapping associates a query to each element of the source schema. LAV is more flexible than GAV as describing information sources does not involve information sources and the semantic relationship of other queries.

Liang et al.[182] proposed an approach for extracting query attributes from query forms and translating the source queries into target queries. The query translation has valid attribute extraction and automatic form filling. Some preprocessing is done before extracting valid attributes. Yan et al.[202] used a house domain of Chinese environment. The frame used for integrating query interface is called integrating query interfaces based on ontology (IQIBO). They also use universal query interface (UQI) that is built using domain ontology that needs not to be changed when the query interfaces of the Web database change.

Form-Based Approach

This section discusses the studies that focus more on finding forms than issuing queries and keywords to those forms. As there can be different type of forms on a webpage, the crawler should be able to distinguish between searchable forms i.e., query forms and nonsearchable forms. Subscription forms, login forms, message posting forms, and polling forms are some of the examples of nonsearchable forms. Akilandeswari and Gopalan[164] presented the design of a novel Web crawler using reinforcement learning-based agent. First, the crawler automatically finds out websites having hidden information. Second, by filling out forms with relevant keywords, the information from these websites can be gathered. The learning module uses webpage classifier and link classifier. Form classifier can differentiate between searchable and nonsearchable forms. Nonsearchable forms are the interfaces for login, subscription, and registration. Furche et al.[199] proposed an approach for understanding the forms. Ontology-based Web pattern analysis with logic (OPAL) divides the form understating into labeling and form interpretation. Form labeling identifies the form and its fields, arranges them into a tree, labels the found fields, and segments forms from the webpage. Based on the labels form interpretation aligns a form labeling with a domain.

Barbosa and Freire proposed form focused crawler (FFC)[170] that can gather topic-specific forms effectively while avoiding unproductive searches. FFC uses a form classifier, domain-specific form classifier, and an adaptive link learner in which features are extracted from the path learned. The main disadvantage of FFC is that training link classifier is a time consuming task and the forms retrieved are highly heterogeneous. An enhanced form focused crawler (E-FFC) is an enhanced framework proposed by Li et al.[84] It is an improvement of FFC proposed earlier. It crawls webpages and classifies them according to taxonomy. Link classifier learns the feature of the links leading to target path that contains webpages having searchable forms. Adaptive domain feature learner automatically learns pattern from form databases samples to filter searchable forms from nonsearchable forms. E-FFC is better than FFC in terms of harvest ratio and coverage rate.

Nguyen et al.[192] gave a method to extract the form labels. They used learning classifier for element label mapping. In extraction phase, the mappings are generated and form features are extracted. The proposed approach does not require human input to define rules in which human intervention is required only for the training of the classifier. The same team of authors proposed PruSM: a prudent schema matching approach for Web forms[193] in which matches are determined for a form element with the goal of minimizing error propagation. They group the form elements into attributes set and frequent attributes are used in the matching discovery module. An attribute is considered frequent if its frequency is above a threshold. The similarity between two attributes is computed based on label similarity, domain value similarity, and correlation.

Revisit and Incremental Approach

This section discusses the studies that incrementally collect data from hidden databases. Efficient revisit policy for a crawler can be devised so that data at

the local server is always updated. Revisit policy can be decided based on the change probability of a webpage. After randomly constructing the first query, subsequent queries use its result to build queries that are more efficient. Huang et al.[206] discussed incremental harvest model for incrementally crawling the deep Web. The model is constructed using records in the database at a different time interval from the local database. It also uses incremental records between neighboring versions of Web databases. The first query is selected from the randomly selected initial query list and result of which is saved in the local database. The query is rebuilt using the harvest model. Chandramouli and Gauch[172] highlighted a cooperative Web service using information present in Web logs and file system. The proposed method decreases the amount of data collected by a crawler and to keep the search engine collection up-to-date. Using weblogs, speculative queries are generated and webpages are collected as viewed by the user in response to the actual queries. When a query is submitted the URL and the query is also recorded in the weblog.

Liu et al. developed DP9,[184] an open source gateway service that allows search engines to index open archives initiative (OAI). They created HTML webpage for an OAI collection, having HTML links that can generate OAI request for a specific identifier. It has three main modules: URL wrapper, OAI handler, and XLST processor. Madaan et al.[186] devised an incremental hidden Web crawler for domain-specific Web. Proposed architecture has following modules: domain-specific hidden web crawler (DSHWC), URL extractor, revisits frequency calculator, update module, and dispatcher. Based on the probability of updation of a webpage, the time period between two successive revisits can be adjusted. Zhang et al.[205] proposed a framework based on URL classification for an incremental deep Web crawler. The framework not only crawls listed webpages in deep Web but also crawls leaf webpages that change often.

Attribute and Label Extraction Approach

Form labels and attributes are the prime requirements for many techniques that aim at retrieving data from hidden databases. Studies that focus mainly on extracting attributes and labels from forms are included in this section. The attributes and labels of a Web form play a major role in deciding the values that are to be used to extract data from hidden Web. The domain of an attribute in a form can be finite or infinite depending upon its type like a dropdown list has a fixed domain while a textbox has an infinite domain. Furthermore, the label of the attribute present in a form can restrict the infinite domain of a textbox. An et al.[167] provided a method to measure the semantic similarity of query interface attribute with used keywords. They considered attributes in two possible ways: programmer viewpoint attributes (PVA) and user viewpoint attributes (UVA). To automatically extract attributes firstly, PVAs are obtained from inner identifier of the sources and UVA are obtained from the free text within the query interface. Final attributes are determined by taking the intersection of both.

While doing attribute or label extraction, it is necessary to distinguish between single attribute and multi attribute search form fields. Label extraction technology for domain-specific Hidden Web crawling (LEHW) is proposed in Ref 176. The proposed technique can deal with both single and multiattribute forms without ignoring any search form field. First, the proposed algorithm parses the hidden Web forms to check if they are single attribute or multi attribute. Second, labels are extracted from the multiattribute forms. For single attribute form, the user can type list of keywords in single search box. Labels are extracted to deal with multiattribute forms. To make an efficient query, the valid attributes are extracted and semantic relationship between attributes is determined by using WordNet. Nguyen et al.[191] discussed a machine-learning-based approach for extracting the form labels. Three phases are used for label extraction: candidate mapping generation, classification, and reconciliation. Mapping is done in two stages, at first or initial step, a high-recall classifier is used to eliminate incorrect mapping. In the next step, a high-precision classifier is used so that mapping can be as precise as possible.

Labeled Value Set-Based Approach

Labels from a form are matched to determine the values that can be used for building queries to get data behind the form. These values are extracted from a knowledge database having label-value pairs. In 2001, Raghavan and Molina[207] proposed a hidden web expose crawler (HiWe) that uses label value set (LVS) table in which labels along with their corresponding values are stored. Layout-based information extraction technique (LITE) is used for extracting the values and populating in LVS table. Textual attributes labels of the form are matched with the strings present in knowledge base. Main contribution of the proposed approach is to retrieve significant data before submitting the queries. The HiWe has the capability of rendering a webpage to extract HTML forms from it. HiWe can recognize different dependencies present between the elements

of a form, but it does not support partially filled forms.

DeepBot[165] uses pointers to the documents called as routes. The proposed Web crawler accesses webpages in the same way as a human does, by using a mini Web browser. Crawler start with a list of routes and then some documents are downloaded and examined. From the anchors present in a document, new routes are obtained and added to the list. When no routes are left, then the process stops. Form analyzer associates the form fields with some domain attributes. They claim that the proposed approach is different from HiWe as DeepBot can use a form, even if it has some fields that do not match any attribute of the domain values of LVS. Also, DeepBot fully supports JavaScript sources.

Peisu et al.[194] proposed a framework of a deep Web crawler in which a user has to key in a set of keywords to access webpages from a certain website. Here, a form is modeled as a set of <element, domain> pair. The following sequence of action is performed on any form—first, the form is analyzed, then values are assigned and submitted, response is analyzed, and if the response webpage contains hypertext links, these links are followed immediately.

Domain or Topic-Specific Approach

After identifying the forms, another challenge is to determine the domain of data source behind the forms. This section discusses the studies that focus only on finding domain-specific forms. Moraes et al.[188] presented an up-to-date review of the methods for the discovery of domain-specific forms that do not involve form submission. The objective of their survey is to discuss methods that can locate HTML forms on the Web and out of those filtering the query forms. The focus of their survey is on pre-query techniques. Various techniques are organized into a comprehensive classification depending upon the main goal of the technique. Wang et al.[201] gave a traffic advisory system based on deep Web. The user enters the origin and destination, departure date, and other information. Query module analyzes the information and if requirements are met, results are given back to the client. Otherwise, the system crawls the related websites depending on the user requirements. Domain ontology is the decision-making component of this deep Web crawler.

Kosmix[195] developed by Rajaraman is an intersection of topic exploration and deep Web. The author claims Kosmix to be the first general purpose topic exploration engine to harness the deep Web using a federated search approach. Federated search uses APIs to access the deep Web sources at query time and constructs resultant webpages based on the responses. Kosmix categorization service (KCS) determines the nodes in the taxonomy of topics that are most closely related to a given query.

Table 5 gives a comparative analysis of various studies related to hidden Web crawler along with their strength and weakness. Precision and recall are the two important metrics that are used by this category of Web crawlers and hence various studies are compared on these two paremeters.

Mobile Crawler Techniques

It is reported in the literature that around 40% of the Internet traffic and consumption of bandwidth is due to Web crawlers.[10] Centralized data access, centralized webpage filtering, and centralized index are some of the limitations of traditional Web crawler that can be handled efficiently by the mobile crawler. A mobile crawler is a program that can transfer itself to the Web server to download information and contents available on the server. The mobile crawler uses the approach of bringing 'code to data' instead of traditional 'data to code' approach. Hammer and Fiedler[9] in the year 2000 suggested a breakthrough detailed architecture of mobile crawler. Various issues such as scaling issue, efficiency issue, and outdated index quality issue are discussed in the study. They gave a mobile approach to Web crawling with the advantages such as localized data access, remote webpage selection, remote webpage filtering, and compression that are highly efficient regarding resources. To execute the mobile crawler at a remote position, an application framework architecture that is responsible for the creation and management of mobile crawler is proposed. The virtual environment at the server manages various policies, so that only particular code can be executed. Query search engine is provided that allows the user to issue queries to the crawler.

There are few studies about the mobile crawlers that are available in the literature that are categorized as shown in Table 6 and each category is discussed in detail next.

Freshness and Revisit Policy-Based Mobile Crawler

Due to dynamic nature of the Web, there is a continuous need to revisit a website to maintain the index of a search engine up-to-date. Traditional crawler needs to have a series of communication with a server for downloading webpage of interest at local site. The mobile crawler can significantly reduce this overhead by moving itself to the server. The concept

TABLE 5 | Hidden Web Crawler Studies and Comparative Analysis

	Category	Support for Focused Crawling	Strengths	Limitations	Tools Used	Precision	Recall
Alvarez et al.[165]	LVS	Yes	Compatible with Javascript sources	No support for session mechanism	-	High	High
Barbosa et al.[170]	Form based	Yes	Resources utilization is good	Slow learning process	ACHE	High	High
Chandramouli et al.[172]	Keyword query-based retrieval	No	Low bandwidth requirement	All server may not allow to store information	-	Average	Average
El-Desouky et al.[175]	Keyword query-based approach	Yes	Can handle both single and multivalued attributes	No support for scheduled queries	-	High	Very high
Furche et al.[199]	Form based	No	First comprehensive approach for forms understanding and integration	No support for dynamic forms	OPAL	High	High
Gravano et al.[8]	Keyword query-based retrieval	Not specified	Efficient, scalable, and accurate crawling method	Dependencies on classifier are not always correct	Qprober	They use $F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$ F1 value is average	
Ipeirotis et al.[179]	Keyword query-based retrieval	No	Rule-based approach is used and hence other language methods can be applied directly	Proposed algorithm does not inspect any document it just query probe the database	RIPPER	F1 value is average	
Li et al.[84]	Form based	Yes	High harvest ratio and coverage rate	Retrieval of redundant information	EFFC	High	High
Mesbah et al.[187]	Document object model	Yes	Can be used for crawling highly dynamic websites and tool is open source	No support for client site Java scripts	AJAX and CRAWLJAX	High	High
Nguyen et al.[192]	Form based	Not specified	No human intervention required	Each element mapping is considered in isolation	LABELEX	High	High
Ntoulas et al.[204]	Keyword query based retrieval	Yes	Proposed approach is adaptive and optimal	Only supports single attribute keywords queries	-	Coverage is used as metrics whose value is high	
Yu et al.[203]	Document object model	Not specified	Quick, efficient and accurate	Duplicate data extracted, nonversatility	-	High	High

of mobile crawler was proposed in Ref 9. Mobile crawler outshines the centralized architecture of the current Web crawling system. They perform crawling at the server end and send the compressed data back to the side of origination of the crawler. Distributed web crawling and indexing system (DWCIS) is introduced in Ref 10 that works on the concept of master slave design. master agent (MA) resides at the search engine side and manages slave agents (SA). SA is dispatched and once it reaches to server, it requests data and generates an index, which is sent back to MA. A similar technique of distributed Web crawling based on mobile crawling is proposed in Ref 249.

According to Nath and Bal,[250,251] no search engine has succeeded in covering more than 16% of estimated size of the Web due to slow crawling process. The proposed technique uses webpage size as one of the criteria to decide whether the webpage has been

TABLE 6 | Categories of Mobile Crawler

Sr. No.	Category	#	Citation
1.	Freshness and revisit policy-based mobile crawler	5	10, 244, 249, 250, 251
2.	Security issues in mobile crawler	3	242, 247, 252
3.	User feedback-based mobile crawler	2	243, 245
4.	Agent-based mobile crawler	1	246
5.	Ontology-based mobile crawler	1	248

modified or not since the last visit. They used mobile crawlers that can identify the modified webpages at the remote site without downloading them. Hence, the crawler helps considerably in reducing Internet traffic and load on remote server. However, they do not take into consideration of the distributed indexing. They also discuss a mobile agent-based crawler that retrieves webpages, process them, and compare their data.[251] This comparison is used for determining which webpages have been modified since the last crawl. The proposed model consists of a crawler manager, frequency change estimator module, statistics database, old database file module, a comparator module, analyzer module, and remote server. Change frequency of every webpage is saved and depending upon the statistics revisit frequency is decided.

Kausar et al.[244] presented an approach based on Java Aglets for mobile crawling. The proposed approach reduces network load and traffic. The mobile agent systems consist of two components: mobile agent and aglet platform. The home platform is responsible for creating, initializing, dispatching, receiving, and eliminating a mobile agent. Aglet agent is responsible for creating, managing, and dispatching it to the remote host. A security manager is used to protect the aglet platform and aglets from malicious entities. Proposed approach does not take into consideration the changes in the structure of a webpage.

Security Issues in Mobile Crawler

For making mobile crawler a real success, some security issues need to be tackled. The security issues in the mobile agent can fall in any of the four categories as discussed in Ref 242. These are host to migrant attack, migrant to host attack, migrant to migrant attack, and third party attack. Authors proposed to encrypt information at migrant level. The method uses a combination of invertible function and multiplicative inverse for the text encryption.

Upadhyay et al.[247] highlighted a similar encryption-based approach for the mobile agent. As the mobile agent has to visit different nodes in the network, security should be applied to the platform. Encryption/decryption is used to ensure the security of agents and host resources. Pahal et al.[252] proposed a security solution for the mobile crawling. The data are encrypted before passing it to mobile agent and decrypted back on visiting the host website.

User Feedback-Based Mobile Crawling

Whenever a query is submitted, enormous results are given to the user, but those results lack refinement. This category discusses the mobile crawler that uses feedback submitted by a user for ranking and indexing of results collected by a crawler. Gupta et al.[243] discussed an architecture that continuously changes retrieval process to work according to the latest trends. It not only uses synonyms words but also semantically related keywords for ranking and indexing. For ordering of URLs, it extracts top N terms from the topic model repository. The term frequency of each top N terms in the topic model repository is calculated. Weight is assigned according to the place of appearance of a keyword in the webpage i.e., header, summary, or body.

Site-oriented Processors for HTML INformation eXtraction (SPHINX)[245] is proposed which is Java-based toolkit and support crawlers that are site specific, personal, and relocatable. SPHINX provides a facility for encapsulation of knowledge base in reusable objects called classifier. They provide a graph visualization of the website that is beneficial for the programmer. Category ranking and language modeling are two interesting crawlers written with the help of SPHINX. Category ranking uses priority driven crawling of SPHINX. Language modeling crawler uses just in time language modeling technique. The relocatable crawler can be implemented using SPHINIX and that is the reason this study is included in this category.

Agent-Based Mobile Crawler

A focused crawling approach using multiagent Web search system I-spider is used in Ref 246. User submits a query and agents collaborate with each other to decide the order in which URLs should be downloaded. It converts the format of HTML webpages to XML before indexing them. The frameworks used consist of multiagents, local database model, mode base system, and knowledge base module.

Ontology-Based Mobile Crawler

A novel method to measure the understanding between two agents in the related domain is proposed in Ref 248. Whenever a crawler goes into a certain domain, it is first compared with the agents in existing domain. They measure concept–concept similarity, concept–ontology similarity, and ontology–ontology similarity.

In Table 7, a comparative analysis of various mobile crawler studies along with their strengths and limitations is presented. As mobile crawler were proposed to handle the different limitations of traditional crawler so Table 7 also presents the parameters based on which that particular study is better than a traditional crawler.

Open Source Web Crawler Used by Various Studies

There are many open source Web crawlers available on the Internet. These crawlers are based on different programming languages. Any researcher can easily extend them to cater their needs.

As shown in Table 8, a few studies have mentioned the open source crawler they have used. Nutch is used by six studies, as it is one of the most advanced and active projects in Web crawling field. Its support for distributed computing makes it dear of many researchers. Babouk[45] is based on Apache Nutch and they have used its distributed crawling feature. Most of the studies do not mention the reason for choosing a particular crawler from the pool of open source crawlers available. OCCS[81] is not directly using Nutch, but uses Nutchwax for indexing ARC files. Nutchwax gives the feature of adding extra fields to the index. Nutchwax can be used with a servlet like Tomcat to provide the free text searches. Ranking feature of Nutch is also used by OCCS. Neunerdt et al.[98] extended Nutch to examine the performance of their proposed algorithm.

Nutch is flexible as it gives parameter settings like choosing the depth and maximum number of webpages fetched from one layer. Uzun et al.[130] use depth setting parameter of Nutch so that the content extracting process can be optimized. Zheng et al.[154] developed a Nutch-based focused crawler for hardwaretoday.com. They used Nutch to crawl according to the breadth first strategy to generate a list of subwebpages. They modified the crawling list generator of the Nutch to get filtration algorithm for URL rule-Based Focused Crawler (UBFC) and BaseLine Focused Crawler (BLFC). Zhuang et al.[157] compared the proposed algorithm with Nutch on the parameter of 'number of papers harvested on the ACL conference website.'

Another Java-based open source crawler, Crawler4j is used in Ref 32. It is customized to crawl based on FCFS strategy. They named it as 'classic focused crawler' and used it as a baseline crawler to compare the results with their proposed crawler. In Ref 34, crawler4j is used for implementing the proposed method, they have used Jsoup library to parse HTML documents. The limitation of crawler4j is that it does not provide support for robot.txt.

Mercator Web crawler is an exception, as it was a commercial Web crawler used by AltaVista replacing older scooter crawler. Most of the commercial Web search services consider crawler code as trade secrets. However, later Mercator code was made public.Mercator is a Java-based crawler having both scalability and extensibility. The multitext crawler that uses the design goals of Mercator is extended by X4 crawler in Ref 212. Mercator is claimed to be the fastest crawler, which can gather webpages at the rate of 112 webpages per second.[255] Many studies refer Mercator but a few mention where they have actually used it.

OCCS[81] used another Java-based open source crawler Heritrix which is extensible, scalable, and of archival quality. Heritrix can perform broad, focus, continuous, and experimental crawling. The OCCS integrates Hertirix with the language guesser Java tool command language (JTCL) and text classifier rainbow. Gao et al.[58] extended another open source crawler Larbin that is written in C++.

Scrapy is a Python-based Web crawler proposed in 2008. Radu and Rebedea[112] designed RWscrapper using Scrapy framework. Scrapy provides basic infrastructure necessary to extract Web content using user-defined rules. The architecture of Scrapy consists of three key elements: items, spiders, and processing pipelines.

Language of Implementation Used by Various Studies

Various studies have used different languages for the implementation of their proposed techniques. We have classified them as shown in Table 9 according to their language of implementation. Seventy-two percent of the studies as shown in Figure 4 use Java as the implementation language. It also includes studies[32,34,45,81,130,154,157] which have extended Java-based open source crawlers.

Java is the most widely used language for the implementation of crawler and Figure 4 shows its clear dominance. Miller et al.[245] devised SPHINX a Java-

TABLE 7 | Mobile Crawler Studies and Comparative Analysis

	Category	Strengths	Limitations	Tools or Language or Crawler Used	Parameter on Which Mobile Crawler Is Better Than Traditional Crawler
Badawi et al.[10]	Revisit policy	Use fewer resources than traditional crawler	Server side execution of code is not always possible	DWCIS	Amount of data transferred and network load
Dixit and Sharma[242]	Security issues in mobile crawling	Security of migrant crawlers is handled effectively	Encryption may be high on resources and time	Invertible function and multiplicative inverses	Security of migrating code
Gupta et al.[243]	User feedback-based ranking	Semantically related keyword used for ranking and indexing that results in improved precision	With time size of prospective table may become unmanageable	-	Relevance rate
Hammer and Fiedler[9]	Architecture of mobile crawler	Detailed architecture along with advantages and disadvantages is discussed	Security problem in code migration and integration of mobile crawler virtual machine with Web server	Java, CLIPS rule, and SQL	Network and total load in terms of data transferred
Kausar et al.[244]	Revisit policy	Efficiently handle URLs generated by JavaScript functions	Only content change is taken into consideration not structural change	Java aglet	Number of bytes retrieved by crawler and freshness of search engine
Miller et al.[245]	User feedback-based ranking	Personal crawling without programming and site specific	Loose integration between browsing and crawling	SPHINX/Java	-
Nath and Bal[251]	Freshness maintenance	Revisit is decided based on past change statistics	No provision of distributed indexing	PMCS	Low bandwidth, load on network, webpage change behavior
Pahal et al.[252]	Security in mobile crawling	Encryption/decryption-based technique for securing the mobile crawler	No option of self-securing support for mobile crawler	Java applet, servlet	Security of mobile crawler
Pandey and Mishra[246]	Agent based	Focused crawling based on multiagent collaboration technique	More elaboration of model is required	Ispider	Speedup of processing and reliability

based toolkit for Web crawling. They provide an interface that can be used by the programmer for writing crawler directly in Java. They also state that most of the state-of-art crawlers are coded in Java, C++/C, or Perl.

OCCS[81] used Java-based crawler as well as tools such as JTCL and Rainbow that are also implemented in Java. JTCL is a Java implementation of TextCat, a text categorization library that is used to create a written language identification tool.

An et al.[167] used C++ with Borland Delphi to implement their proposed algorithm. Delphi is used to download Web data sources and extracting PVAs and UVAs automatically. Chakrabarti et al.[6] also used C++ for the proposed approach of focused crawling. Some studies[57] used C++ for implementing a prototype of crawling and also used Java for indexing and searching. Ref 62 is included because it uses open source larbin that is written in C++. Ozel[101] used Microsoft Visual C++ compiler for the implementation of various proposed experiments. SLAIR text-mining framework that is a C++-based suite is discussed in Ref 158. A multiplatform framework integrates optimized clustering with text-mining technologies.

Distributed focused crawling is implemented using C# in Ref 19. C# provides libraries to access

TABLE 8 | Open Source Web Crawlers Used by Various Studies

Name of Crawler	Language Used	Citation
Nutch	Java	45, 81, 98, 130, 154, 157
Crawler4J	Java	32, 34
Mercator	Java	212, 239
Larbin	C++	58
Heritrix	Java	81
Scrappy	Python	112

data via computer networks and to develop multi-threading architecture. To crawl Vietnamese webpages[99] used word segmentation, and a genetic algorithm implemented in C#. A Linux-based learnable topic-specific Web crawler is implemented in C.[227] For hidden Web crawling, a DotNet technology-based crawler using a tool prototype is proposed in Ref 174.

Subject Systems

This section discusses various subject systems used in the literature for the validation of proposed technique. These subject systems are used to evaluate the crawling process and for comparative analysis. Various subject systems are presented in Table 10. We are hopeful that this table may help researchers in choosing an appropriate subject system for their study. We have identified 18 subject systems, all are open source in the way as they are freely accessible for the crawling purpose.

Open Web, that is, Internet is the most commonly used subject system, as any crawler has in its access the free Web. Any website in the open Web can be crawled except those webpages listed in the robot.txt of the website. Most of the studies have used webpages as a subject system without applying any restriction. The number of webpages crawled may vary from a few to up to 50 million webpages. Number of webpages used by different studies ranges from 400 webpages[221] to 50,000 webpages,[218] depending upon their requirements,

TABLE 9 | Language of Implementation Used by Various Studies

Sr. No.	Category	#	Citation
1.	Java	44	9, 10, 32, 34, 45, 46, 47, 48, 49, 50, 51, 54, 57, 60, 71, 74, 80, 81, 98, 102, 103, 105, 106, 109, 114, 121, 124, 128, 130, 141, 146, 148, 149, 154, 157, 160, 187, 190, 211, 220, 238, 239, 244, 245,
2.	C++	8	6, 57, 58, 62, 101, 104, 158, 167
3.	C#	5	19, 42, 99, 134, 200
4.	C	1	227
5.	DotNet	1	174
6.	Perl	1	33
7.	JADE	1	30

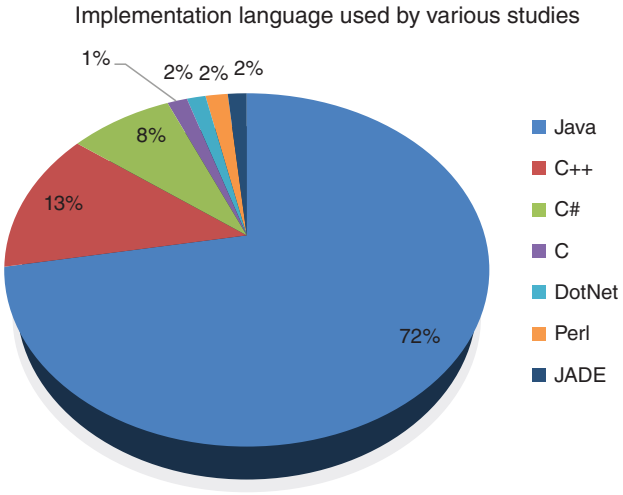

FIGURE 4 | Implementation language used by various studies.

Seed URLs from the open Web is the second category in Table 10. The number of URLs may vary from 120 URLs to 100,000 URLs.[213] Some studies have mentioned the size of data they have used for a particular topic, for example, Ref 9 have mentioned 9 MB as the size of data crawled. Most of the studies of topic-oriented crawler use this category as a subject system. Ref 99 uses diverse topics as world cup soccer, information visualization, and Titanic. Studies using more than one topics are also listed in Table 10.

To access deep Web resources, open Web is also used as subject system as shown in above table. Open directory project (ODP) by Mozilla.org is the second most widely used open source subject system. ODP provides a categorical collection of URLs that are edited manually and are not biased by commercial consideration. ODP is used in three ways in literature-crawling documents available in ODP, crawling some topics of the directory and taking some seed URLs from any category of ODP. The number of topics can vary from one topic as in Ref 161 to any number like 100 topics used in Ref 235.

In some cases, the proposed study is meant only for a particular website or domain. In such cases, the researchers are supposed to choose that particular website as the subject system. As in twitterEcho,[33]

TABLE 10 | Subject System Used by Various Studies

Sr. No.	Subject System Used	Category	Amount of Information in Number	Citations	#
1.	Open Web	Webpages	200 to 10,000	10, 17, 32, 35, 72, 75, 80, 84, 86, 88, 95, 96, 99, 104, 115, 129, 130, 142, 160, 164, 209, 218, 221, 225, 231, 232, 254,	27
		URLs	10 to 8523	18, 31, 40, 41, 52, 60, 77, 98, 114, 143, 146, 149, 155, 202, 213, 227, 253	17
		Topics	4 to 100	55, 58, 71, 94, 106, 144, 148, 152, 156, 238	10
		Single topic	-	9, 34, 44, 54, 93, 97, 220, 241	8
		Queries keyword	23 to 1 million	89, 91, 139, 162, 168, 180, 211	7
		Deep Web resources	8 to 2884	181, 183, 189, 191, 192, 194	6
		Websites	21 to 4874	30, 108, 138, 154, 172, 207	6
		Web documents	1051 to 100,000	21, 23, 42, 101, 109, 236	6
		Others	-	28, 76, 112, 128, 170	5
		Particular domain	-	85, 120, 167	3
2.	Open directory project (ODP) or Directory Mozilla (DMOZ)	Documents	500 to 40 million	24, 37, 68, 79, 105, 107, 113, 122, 163, 178, 204, 212, 219, 228, 240,	15
		Topics	2 to 100	39, 66, 73, 87, 103, 107, 161, 210, 214, 234, 235	11
		Seed URLs	20 to 450	123, 127, 150, 222	4
3.	Particular website	-	-	*Website name (reference number)*: OCLC website,[19] YouTube,[20] Twitter,[33] Wikipedia,[48] compass.com,[50] netriu.fr,[62] Virgilio.it,[63] Netease military channel,[64] dangdang.com,[69] 163.com,[134] Chinese travel site,[136] chemicaljournal,[137] hardware-related site,[145] sports.sina.com,[147] flicker,[151] acm.org,[169] turbo.com,[174,175] sina.cn, cmea.co.uk,[199] Amazon,[204] 51job.com, ganji.com,[205] sbgl.jdzj.com,[216] blog.sina.com.cn,[230] mathforum.org[239]	24
4.	University website	-	-	22, 25, 82, 215, 237	5
5.	Australian yellow pages	-	200 to 2000 webpages	46, 47, 49	3
6.	Yahoo! directory	-	5000 to 260,000 documents	6, 83, 224	3
7.	CiteSeer	-	150 to 593	157, 223	2
8.	Dummy databases	-	10 records	15, 174	2
9.	Google Web directory	-	14 categories	171, 198	2
10.	Newsgroup articles	-	419,000 articles	8, 179	2
11.	Nano Science and Engg. (NSE)	-	20,000 to 996,028 webpages	110, 111	2

(*continued overleaf*)

TABLE 10 | Continued

Sr. No.	Subject System Used	Category	Amount of Information in Number	Citations	#
12.	UIUC website (University of Illinois Urbana Champaign)	-	447 to 15,600 webpages	166, 197	2
13.	Reuter corpus volume	-	-	105	1
14.	SQUID Proxy traces	-	-	208	1
15.	California housing dataset	-	20,640 data points	27	1
16.	Seedfinder	-	15 websites	256	1
17.	MetaQuerier	-	3 domains	182	1
18.	Two public form collection	TEL8 and FFC	-	193	1

only Twitter can be the subject system, in a similar way for videos crawling YouTube is used as a subject system.[20] We have found 24 studies in total that belongs to this category as shown in Table 10.

Another category is the universities website as the subject system. We have determined five such studies that use Japanese universities,[82] UTM University,[22] Turkish universities,[25] and other. Dong et al. have used Australian yellow pages for their proposed studies.[46,47,49] New technologies and amended tools can be compared easily using these open source system.

PERFORMANCE METRICS FOR WEB CRAWLER

A Web crawler performance metric is a standard measure of a degree to which a crawler possesses some property. As per the best of our knowledge, Cleverdon in the year 1966[257] was the first one to use precision, recall, coverage, effort, and so forth in the area of IR system for his technical report. Precision and recall are the most commonly used and fairly understood quantities in the field of IR.[258] They define precision as the proportion of retrieved material that is actually relevant and recall as the proportion of relevant material retrieved in answer to a search request. Effectiveness is defined as a measure of the ability of the system to satisfy the user in terms of the relevance of documents retrieved. Evaluation of many studies in IR is done by using precision and recall for two or more crawler on the same set of data.[29] The next subsections discuss the performance metrics for each category of a Web crawler.

Performance Metrics for Focused Web Crawler

Brin and Page[1] mentioned the importance of precision and recall, as the size of the Web grows we need tools to fetch documents having very high precision and recall. However, because of the large size of Web, it is extremely difficult to measure recall for a focused crawler.[6] The most critical evaluation of the focused crawling is to measure the rate at which relevant webpages are acquired and irrelevant webpages are filtered off from the crawling process. This is also called harvest ratio. Robustness is also discussed in the paper as an important metric for a focused crawler. It is the indicator of the ability of a Web crawler to be on-topic webpages, without being too sensitive on seed URLs provided to it. According to Ref 13, use of precision and recall make sense when the relevant webpages are from a finite set. The measurement of execution time that is calculated as the time passed from starting of execution until the agents reach the predefined threshold of crawled webpages.[162] If there is no way to measure the actual number of webpages available, the total number of webpages collected by each crawler is used as metric.[172] Precision (Relevance) is judged by the human inspection which is biased and inconsistent.[6] For calculating recall, it is not always possible to get the exact total number of relevant webpages on the Web, so the recall cannot be measured directly. Therefore, Ref 65 proposes maximum average similarity that is the accumulated similarity over the number of crawled webpages and maximum average similarity as $\max_{d\in T}\frac{\sum_{p\in s}\mathrm{Cos}(p,d)}{|s|}$ where T is the set of target webpages, S is the set of crawled webpages, and cos (p, d) is the standard cosine similarity function.[65]

Precision and recall do not take into consideration the metadata of the webpage. Precision and recall can be condensed into one number using F_1-measure as[8] $F1 = \frac{2*\mathrm{precision}*\mathrm{recall}}{\mathrm{precision}+\mathrm{recall}}$. F_1-measure that is the harmonic mean of precision and recall is referred by the name of harmonic mean in some studies. When the value of harmonic mean reaches the highest, it

signify the values of precision and recall reaches to highest at the same time.[46] According to Ref 116, the primary metric in evaluating crawler performance is harvest ratio. It is always preferred to have the high values of harvest ratio. F-measure tries to eliminate any bias between precision and recall. In the absence of a known relevant set, we treat the recall of the target set i.e., target recall as an estimate of the actual recall.[103] In addition, if targets are random sample of the relevant webpages of the Web, then target recall gives a good estimate of the actual recall. As per our observation, harvest ratio is used widely in the area of Web crawler. Harvest ratio is defined as the fraction of webpages crawled that satisfy the relevance criteria among all crawled webpages.[32] URL overlap ratio between two crawlers is the measure of same set of URLs fetched by both the crawlers. Exclusive harvest ratio gives the exclusively crawled webpages by the crawler under consideration.[24]

For a dynamic environment like Web, it is difficult to have an exhaustive knowledge to determine a topic's entire relevant set. For the assessment of crawler, they used mean similarity between topics and crawled webpages.[94] Accuracy ratio used in Ref 100 is defined as the ratio of detected webpages that are related to the topic to the sum of detected webpages by the crawler. Ref 221 defines the crawling efficiency, accuracy, and consistency for a topic-specific crawler. Efficiency is defined as a ratio of number of URLs related to the specific topic to the total number of URLs crawled. Accuracy is measured how correctly the crawler can classify the webpages based on the terms. Ref 232 compares on topic retrieved documents to the total number of downloads.

Performance Metrics for Hidden Web Crawler

According to Ref 207, the metrics used for traditional crawlers such as scalability, freshness, and so on to measure the effectiveness of crawling activity, are equally applicable for hidden Web crawler. However, these metrics do not measure the effectiveness of automatic form processing and submission techniques. They proposed to use coverage to measure the ratio of relevant webpages extracted by a crawler to the total number of relevant webpages present in the target hidden Web database. However, this metric has two main disadvantages. First, it is hard to determine the relevancy of webpages from the hidden database. Second, this metrics is dependent on the content of database. Authors in Ref 207 suggested using submission efficiency as the ratio of submission count that results in a response webpage containing one or more search result to the total number of forms that crawler submits during the crawler activity. In case of hidden Web, if P_c denotes the total number of webpages crawled by the crawler, N_f is the total number of domain-specific forms of deep Web database collected, N_{cf} is the total number of domain-specific searchable forms in the crawler search space, then harvest ratio is $\frac{Ncf}{Pc}$ and coverage rate is $\frac{Ncf}{Nf}$.[85] In Ref 170, harvest ratio is used to measure the number of relevant forms retrieved per webpage crawled. Ref 174 uses recall (coverage) i.e., the ability to extract as much content from hidden databases as possible. It gives the probability that in the result of the query, documents retrieved are relevant. Precision (relevancy) gives an idea of whether the extracted content from the hidden database is relevant to the query. Ref 179 defines the coverage and specificity for hidden Web as $\text{Coverage}(D, C_i) = \sum_{q \text{ is query probe for } c_i} f(q)$, coverage for a database D for a category C_i is the total number of matches for the C_i query probe. $f(q)$ is the total number of matches from D for query q. Specificity of D for C_i is $\text{Specficity}(D, C_i) = \frac{\text{Coverage}(D, C_i)}{\sum_{q \text{ is query probe for } c_i} f(q)}$. Refs 182,183 measure how precisely the correct translation of forms is done by the crawler out of all deep Web query forms found. Ref 204 defines coverage as the fraction of documents in the hidden Web that can be downloaded using continuously updating query keywords.

Performance Metrics for Incremental Crawler

For incremental crawler, Ref 138 uses three performance metrics. First is the strict miss rate to measure how many actually modified webpages are not included in the final chosen set. Second is the general miss rate that measures how many webpages have been modified more than once in the calculated interval. Last is the coverage rate that is the number of actually modified webpages in the calculated interval. Ref 204 uses total coverage, incremental coverage rate, and effectiveness to measure the performance of an incremental crawler. Consider a database at a particular instance of time (t_i). N_f denotes the number of records in the database at that time. N_a denotes the number of records inserted, deleted, or updated from t_i to t_{i+1} in the database. N_c represents the number of records that the incremental crawler can crawl. N_s denotes the number of records that belongs to above three parts. Therefore, these metrics can be defined as

Total coverage rate $= {}^{N_f + N_s}/_{N_f + N_a}$, incremental coverage rate $= {}^{N_s}/_{N_a}$, and effectiveness $= {}^{N_s}/_{N_c}$.

Performance Metrics for Collaborative and Mobile Crawler

Overlap metric was first introduced in Ref 163 for collaborative crawling. Different crawling nodes may download a single webpage. Overlap is defined as $\frac{N-I}{N}$, where N is the total number of webpages downloaded by the overall crawler and I denotes the number of unique downloaded webpages by the overall crawler. Coverage is defined for collaborative crawler as $\frac{|A_N \cap P_N|}{|P_N|}$, where N is the most important webpages in total. P_N is the set of webpages downloaded by the hypothetical crawler and A_N represents the set of webpages downloaded by actual crawler and this needs not be same as P_N. Another metric for collaborative crawling is diversity,[58] which is $\frac{S}{N}$, where S denotes the number of unique domain names of the downloaded webpages by the overall crawler and N is the total number of downloaded webpages by the overall crawler. It gives an idea if the crawler is biased toward a certain domain. As communication is required by crawler nodes to exchange the crawling information, the communication overhead in collaborative crawling is also used to compare various crawlers. It is defined in terms of exchanged URLs per downloaded webpage.

Coverage and focus are also used as the evaluation parameter for the distributed focused crawler that uses Twitter data.[33] The coverage measures the tweet loss for the monitored users. The focus metric aims to evaluate if the crawler is focused on target Twitter community or not. Sum of information represent the results regarding all collected webpages having relevancy more than a threshold. Fallout rate is the proportion of nonrelevant concept associated with the concept in the whole collection of nonrelevant metadata for that concept in all metadata.[46] Fallout rate is also used to measure the error rate of the crawler.[47]

To measure the performance of Web crawler in terms of bandwidth saving, $\frac{TC-WS}{TC} * 100$ is measured, where TC is number of bytes transferred using traditional crawler and WS is the number of bytes transferred using Web services.

Performance Metrics for Forum Crawler

For a forum crawler, Ref 76 has used two metrics—effectiveness and coverage. Effectiveness measures the percentage of thread webpages among all the webpages crawled of the forum. Coverage measures the percentage of crawled thread webpages to all retrievable thread webpages of the forum. Satisfaction[77] gives a measure of how much the end directory formed after crawling is useful for the user. Discard ratio used in Ref 80 is computed as $\frac{\text{discarded}}{\text{discarded + selected}} * 100$, where discarded denotes the total number of webpages discarded by a crawler upto a certain time and selected denotes the total number of webpages present in the crawler repository at a certain time. Coverage and freshness are two metrics to measure a search engine quality; these metrics unlike running time cannot be improved by technical approaches.[15]

Apart from these categories, next we will discuss some other metrics used in the literature. These metrics are important, but they do not fall in any of the above mentioned categories. Biocrawler,[30] a semantic-based crawler, uses S-throughput. It is defined as the amount of energy gained per unit of bandwidth. The energy of each crawler increases by a fixed amount when the crawler consumes some semantic content and decreases each time the crawler moves. A unit of bandwidth is defined as the opening of an HTTP connection by the crawler.

The Geo-focused crawler uses an entirely different set of metrics. Geo-coverage is the number of retrieved webpages with at least one geographic entity, out of a total number of retrieved webpages. Geo-focus is the number of retrieved webpages with at least one geographic entity. Geo-centrality uses the geodesic paths from an arbitrary node to geographically aware node. It measures how many links are to be followed to reach to a geographical aware webpage.[58]

Ref 125 uses performance to cost ratio to compare several crawling strategies. The quality of different crawler can be compared using metrics like quality score using all crawled webpages, quality score using relevance-feedback relevant webpages.[127] Websites are arranged in the categories of above average quality, average quality, and below average quality. Crawlers are compared based on webpages crawled from these categories. Ref 128 uses a plot between relevant resources to the crawled resources to compare the proposed crawler. The distance of crawling path is a measure that gives a sequence of webpages and links going from a seed URL to a downloaded webpage.[150]

Ref 105 uses error rate (ERR) defined as $\frac{|R_n(P_t)|}{|P_t|} * 100$, where $R_n(P_t)$ is the set of positive documents in the reliable negative data set and P_t is the

set of positive documents in the unlabeled document set. The performance metrics used for tunneling performance is target length.[106] Target length is the distance from seed URL to the target relevant webpage. Running time is a measure of time taken to crawl the same crawl sequence by different crawlers.[62] To measure the number of URLs visited and processed from a database by the Web crawler, URLs to be seen and processed URLs are used in Ref 63. Time cost is the time required if all crawlers under consideration are called for crawling equal number of webpages.[69] Matching ratio[74] is defined as the ratio of matched and unmatched classes collected by the Web crawler.

Scalability is measured by using the time needed to crawl as well as the number of characteristics that will affect the time required.[187] Average precision is used to measure how quick and precise a crawler works. Average precision for the single concept is the average of precision values at each logically linked and relevant metadata for that concept.[49]

It is to be noted that search engine and crawler metrics are different from webpage importance metric that defines the importance of a webpage like page rank, backlink count, location metrics, etc. The Web crawler takes the decision of crawling a webpage and order the URLs based on these importance metrics.[16,22,43]

As it is evident from Table 11, precision is the most widely used metric for comparing various Web crawlers followed by recall, harvest ratio and F-measure. Apart from these metrics, some metrics with their citations are given in Appendix A. Most of the studies have used a combination of metrics to compare the proposed crawlers. The most frequently used combination is precision and recall. To carry out the quality evaluation of IR, the most appropriate set of metrics depends on the user application and implementation of the technique.[29]

TABLE 11 | Performances Metrics Used by Various Studies

Metric	Citations	#
Precision (relevance)	6, 18, 22, 23, 28, 31, 40, 41, 44, 46, 47, 48, 49, 50, 52, 53, 56, 57, 64, 65, 73, 75, 77, 79, 80, 86, 87, 91, 95, 99, 102, 104, 105, 108, 111, 115, 121, 122, 129, 130, 138, 144, 150, 162, 165, 169, 172, 174, 175, 176, 181, 182, 183, 192, 193, 197, 199, 203, 211, 217, 227, 230, 231, 235, 238, 239, 240, 243, 256	69
Recall (coverage)	6, 18, 22, 23, 28, 31, 33, 36, 40, 44, 46, 47, 48, 52, 53, 56, 57, 73, 75, 76, 77, 84, 85, 87, 91, 100, 102, 103, 104, 105, 106, 108, 120, 121, 130, 138, 144, 153, 154, 157, 161, 163, 165, 169, 174, 175, 178, 179, 181, 192, 193, 197, 199, 203, 204, 218, 219, 222, 228, 230, 231, 240, 256	63
Harvest ratio	6, 15, 20, 21, 23, 24, 32, 38, 47, 48, 49, 50, 52, 60, 68, 69, 71, 72, 83, 84, 85, 88, 98, 103, 106, 107, 109, 116, 125, 128, 134, 142, 147, 148, 149, 151, 152, 154, 156, 158, 170, 198, 208, 209, 211, 214, 222, 223, 224, 225, 228, 234	52
F_1-measure	8, 18, 23, 31, 40, 46, 47, 48, 52, 56, 73, 77, 89, 100, 101, 105, 130, 153, 165, 174, 179, 192, 193, 199, 203	25
Accuracy	100, 101, 130, 145, 153, 176, 187, 212, 221, 226	10

DISCUSSION

We surveyed 248 articles out of a collection of 1488 and categorized them in different areas. There is no systematic literature review on the Web crawler available in literature. Ref 188 represents a survey of domain-specific query forms that is related only to hidden Web. Ref 259 also represents a review of crawler but it is short. To carry out this research, we framed four research questions to determine the current status of Web crawlers. We also tried to find the research status of the focused crawler, hidden Web crawler, and mobile crawler and how the subareas have evolved over the time. We have also determined the subject system used for the validation of the proposed technique in the literature. Various performance metrics for each category of the crawler are also discussed. Our focus is very broad than earlier surveys and includes latest research work related to Web crawler up to mid-2014 using systematic literature review guidelines of Kitchenham.[2] We explored all the subareas in detail along with the techniques used.

In the next section, we will try to discuss our main findings, strengths, and weakness of our study. We will start with the key subareas of Web crawler followed by implication for researchers and limitations of the review.

Current Status of Key Areas

Broadly, we divided the literature into three key areas based on their importance from a researcher point of view. Although the areas in Web crawler are not

independent and there are some studies that are highly interrelated.

There is a lot of work being carried out after the year 2003 in the area of focused crawler as compared to hidden Web crawler. The mobile crawler is still in its infant stage with a few studies available. Focused crawler widely uses soft computing techniques. Many applications, based on focused crawler, have been reported in the literature and they can be extended to other domains. The flexible architecture of the crawler as in Ref 158 allows integration with advanced technologies for a specific task like link prediction, semantic analysis, etc. Link analysis approach of focused crawling is combined with different features in Refs 41,42,73,94,155,218,229,232,242,255. Zhang et al.[150] proposed a basic learnable crawler that combines link and text analysis to predict whether the next webpage should be downloaded or not. Also, the problem of indexing is still not discussed in detail by any of the application-based Web crawler articles. Formalizing the webpages as nodes will reduce the crawling problem to a huge graph and this will end up in formalizing it as a set covering problem which is NP hard. After studying all the literature of focused crawler, we believe that there is no universal rule for deciding which classifier will work best in a particular situation. Therefore, every classifier has its own strengths and weaknesses.

Coming to hidden Web and mobile crawler, there is a lot of scope in these areas. In hidden Web crawler, the query-based approach is used extensively. We noticed here that any approach that can handle dynamically generated Web content would be a breakthrough. An et al.[167] propose to create a semantic deep Web by adding an ontology layer to the deep Web.

The area of the mobile crawler is widely open and can become a real success if security-related issues of mobile crawling could be handled effectively. Future lies in the mobile crawler as it can handle the exponentially increasing size of the Web in a scalable manner.

Very few studies extended the available open source crawlers for the implementation of their proposed crawling strategy. More emphasis should be given on the extensibility of the existing open source crawler by the researchers. We have observed that there is a vital requirement of a study to compare various open source Web crawlers based on different parameters.

Implications for Research and Practice

This work can benefit the researchers who are looking for open research issues in the area of a Web crawler and for the practitioner to determine the best technique as per their needs. The crawler technique to be used depends on various factors like the resources available to the Web crawler, the amount of relevant information about the topic of interest on the Web, the total number of websites to be crawled, the rate of change of the webpages, and amount of malicious content on the webpages. The various findings and observations are discussed in this section.

The field of Web crawler is not very old and there is no perfect standard by the research community for the same. Search-based IR is used for managing database of every company. Choosing the appropriate technique according to the application can affect the performance drastically. Traffic on the Internet is increasing rapidly and major contributors to this traffic are Web crawlers, so the efficient crawling techniques can control this bottleneck. Researchers and practitioner need to come together to bring out with a more stringent security protocol than robot.txt. Nowadays it is left on the crawler to follow robot.txt or not. This can be a security breach for the servers. To keep the database of the search engine up-to-date, crawler needs to revisit the server after an interval for recrawling. A comprehensive method is also required for determining the revisit policy so that any crawler will not hamper the server with requests. Batsakis et al.[160] address various issues any researcher may face during design and implementation of a focused crawler.

We consider the following avenues as the most promising for the future research. These are discussed according to the category of the Web crawlers.

Focused crawler: The area of a focused crawler is evolving at a rapid speed. The following can be some active areas of research.

Crawler as a service: After crawling, only a small subset of the crawled data is used. Focused crawling becomes crucial when an organization requires crawling a large portion of the Web. Also, crawler contributes massively to the Internet traffic. Many organization or parties that require same crawled data can collaborate for crawling and later can access the required data. This collaboration can reduce the overall traffic on the Internet and save the resources that are wasted in crawling a single website repeatedly by different crawlers.

Lack of crawling standards: Website administrators sometimes find the crawlers activities suspicious because of security concern and bandwidth usage. Repeated access to a webpage by the crawler trigger alarms for the website administrators. To avoid such scenarios a crawler must provide the contact details of its owner. A standard must be

developed to decide which resources of the server are accessible to the crawler. It will be beneficial for both website administrator and owner of the crawler. Also, rules can be devised to force the Web crawlers to follow robots.txt.

Lack of sentimental search engine: As we are rapidly advancing in sentiment-focused Web crawling, future can be a sentimental search engine. This type of search engine can be used in product reviews, social media applications, marketing, etc. Apart from the keyword match-based search results, the user can be provided with sentiment- and demographic-based results. Existing techniques for sentiment analysis use text analysis and natural language processing to extract subjective information of the webpage. However, we need to consider factors like different context, cultural factors, and linguistic refinements to extract sentiments from the webpage. The sentimental search engine can make the existing search engines more effective for the users.

Crawling multimedia information: To our best knowledge, no existing crawler downloads and processes large size objects like videos, images, and audio files. These objects account for a large portion of the Web content. Multimedia crawling requires techniques to analyze high-level semantics from the low-level features of multimedia objects. Crawling and indexing multimedia data can be a futuristic area of research. Website developers should use rich multimedia only when it is required and should provide a textual description of the content present on the webpage. This can be an active research area: first, because of the large size of multimedia data on the Web and second, due to the valuable content it represents in today world. Also, website developers must be provided with a standard to give a textual description of multimedia data on a webpage that can be used by the crawler to categorize the data.

Understanding website structure and seed URL: A good website structure not only improves the user experience but also helps in fast crawling. Structure of the website makes it easy for a crawler to determine which part of the website to crawl for the information of interest. Also, choosing the correct seed URL for crawling is important as crawler move forward from the given seed webpage. Understanding website structure and best seed URL also paves for crawler success.

Social network crawlers: Most of the social media website allows to access controlled data using free Web-based APIs. For the more complex needs, there is always need to crawl social media websites. Even, LinkedIn publically states that crawling is illegal. As social media has data that can be used for decision-making, it becomes very important to devise a method for crawling it. A generalized API can be devised for crawling a particular social networking website. This can be helpful in keeping track of various stories, trending issues, and group activities on the social networking.

Hidden Web: Large size of data and its importance for the user makes the hidden Web crawlers an important topic of research.

Handling client side scripts: Most of the work in deep Web deals with webpages having HTML code. No technique handles the webpages that are generated dynamically at the client side. The support of handling client side-generated dynamic webpages is urgent as it is used extensively these days. The techniques in the literature do not handle JavaScript and AJAX webpages that are generated dynamically.

Dealing with non-HTML search interfaces: Today many search interfaces are implemented in flash or using Java applets. Traditional hidden web crawlers ignore such interfaces. These interfaces need to be handled effectively to reach a sizeable portion of the hidden Web. The technique in the literature effectively handles HTML search interfaces but ignore non-HTML search interfaces.

Search interface detection and extraction: Search interfaces act as the entry point for any hidden Web database. In the literature, there are many methods to determine whether a webpage contains a form or not. However, it becomes tough to determine which form is search form and can be used to access the hidden database. Various soft computing techniques can be used to determine relevant forms. Moreover, we still know very less about the structure of deep Web and features that can distinguish deep Web from the surface Web.

Generalization of labeled value set: LVS table has labels and corresponding values, this type of tables are domain specific. In the literature, LVS table is either populated manually or the values are added from the query results. There is a need of a technique that can be used to pass values to the search interfaces. We can think of some common database that can be used for querying the web databases and creating a classification of datasets obtained into multiple domains. A big database of the labels and corresponding values pair can be constructed. The domain-specific LVS table for any proposed technique can be populated from such database.

Indexing technique for hidden data: Once the data are extracted by submitting the forms the resulted webpages are indexed. Traditional indexing approach stores the URLs of the webpage against the query that collected it from the Web. Later, query

processor matches the user query with the indexed keywords and returns the resulting URLs. This technique is not appropriate for indexing hidden Web because same resultant URLs may be returned in response to the different queries. In such cases, same set of URLs is to be stored against a different combination of queries. Indexing the hidden Web data can be a future area of research.

Mobile crawler: This area is still growing and can be the future of crawling. Very less work is done in this area and there are many possibilities in the field of mobile crawler.

Server side push methods: It is very difficult for a crawler to determine how often a webpage of interest changes. It is a resource intensive task even if a mobile crawler is used to maintain the repository of the search engine up-to-date. However, if a server agrees to push subscribed changes to the Web crawler it can save resources and minimize the traffic on the server. Minimizing the overhead on the crawler and devising an effective push-based method can be a future research area. Keeping the database of search engine up-to-date is an open research issue. As there is a trade-off between coverage and freshness of the result in a search engine.

Security issues in mobile crawler: As the code of the mobile crawler travels to the server and uses its resources to crawl the data. The server administrator may consider it as a security breach. Providing virtual environment at the server side can be one of the solutions. As the mobile crawler will use resources of the server for crawling and indexing so, a time when load on the server is low should be chosen to do the specified tasks of mobile crawling. This field needs to be explored further to make mobile crawler a real success.

The data on the Web are not restricted to structured data, a huge amount of unstructured and semi-structured data are being created continuously on the Web. This large data from one point of view can be considered as 'big data.' Crawling and indexing big data can also be an interesting research area.

Some search engines like 'AltaVista' started using Mercator crawler. AltaVista was closed officially on July 8, 2013. The working and design of Mercator are available online. As more and more researchers will join the open source community, it will facilitate the evaluation and testing of the crawlers in various domains. Researchers can combine the functionality of various open source Web crawlers to make the crawling process scalable and efficient.

Moreover, there is an immense difference in the crawling techniques used by the academicians, researchers and those used in the industries. Algorithm and technique used by the commercial search engines are not known to outside world. In addition, these crawlers algorithms deal with tons of billions of webpages. However, very few academic crawlers deal with such huge amount of webpages. Therefore, industry and academicians need to work together to further exploring the area of Web crawlers.

Limitations of the Review

One of the limitations of this study is multiple meaning associated with the keyword 'crawler.' We have been extremely cautious in data extraction step. Manual search for including the missing articles was done. However, a manual search may miss some relevant articles.

Among the authors to discard or to include a study, disagreements arose. The experience of one author of carrying out similar systematic review was used extensively at each stage. Any conflict was resolved with discussions, until a common consensus was reached. Subcategories were decided based on available literature and studies collected.

Conclusions and Future Work

In this paper, we extracted studies from various e-sources related to the Web crawler. A total of 248 studies were involved in the literature review out of 1,488 articles published in leading journals, conferences, and workshops. Extracted studies were categorized into different areas of a Web crawler. A detailed description of each category is presented. We have also arranged and analyzed the studies from various points of view like open source crawler used by different studies, the language of implementation used by different studies, and subject systems used by them.

Our study shows that researchers are working intensively in the field of Web crawler. We have noticed that field of Web crawler is used extensively for applications that target a particular group of users. A Web crawler that deals with getting data from hidden Web is also of interest for the researchers around the globe. To handle the exponential growth of the Web, mobile crawler will get popularity in the coming days. We believe that this work will act as a research ground for the key areas of a Web crawler. This study could be used not only by researchers but also by practitioner and developers to get an insight of Web crawlers.

In the future, we would like to design a crawler for the area of 'Web crawler' which can get all the

studies on the Web related to the field of Web crawler automatically. This can be done by using a keyword based focused Web crawler. Also, a collaboration of researchers in this area will be required to keep such a database up-to-date.

REFERENCES

1. Brin S, Page L. Reprint of: the anatomy of a large-scale hypertextual Web search engine. *Comput Networks* 2012, 56:3825–3833. https://doi.org/10.1016/j.comnet.2012.10.007.
2. Kitchenham B. Systematic literature reviews in software engineering—a systematic literature review. *Inf Softw Technol* 2009, 51:7–15. https://doi.org/10.1016/j.infsof.2008.09.009.
3. Budgen D, Brereton P. Performing systematic literature reviews in software engineering. In: *ICSE '06 Proceedings of the 28th International Conference on Software Engineering*, Shanghai, China, May 20 – 28, 2006, 1051–1052. https://doi.org/10.1145/1134285.1134500.
4. Brereton P, Kitchenham B, Budgen D, Turner M, Khalil M. Lessons from applying the systematic literature review process within the software engineering domain. *J Syst Softw* 2007, 80:571–583. https://doi.org/10.1016/j.jss.2006.07.009.
5. Rattan D, Bhatia R, Singh M. Software clone detection: a systematic review. Elsevier; 2013, 1165–1199. https://doi.org/10.1016/j.infsof.2013.01.008.
6. Chakrabarti S, Van Den Berg M, Dom B. Focused crawling: a new approach to topic-specific Web resource discovery. *Comput Networks* 1999, 31:1623–1640. https://doi.org/10.1016/S1389-1286(99)00052-3.
7. Yu HL, Bingwu L, Fang Y. Similarity computation of web pages of focused crawler. In: *2010 International Forum on Information Technology and Applications*, 2010, 2, 70–72. https://doi.org/10.1109/IFITA.2010.308.
8. Gravano L, Ipeirotis PG, Sahami M. QProber: a system for automatic classification of hidden-web databases. *ACM Trans Inf Syst* 2003, 21:1–41. https://doi.org/10.1145/635484.635485.
9. Hammer J, Fiedler J. Using mobile crawlers to search the Web efficiently. *Int J Comput Inf Sci* 2000, 1:36–58.
10. Badawi M, Mohamed A, Hussein A, Gheith M. Maintaining the search engine freshness using mobile agent. *Egypt Informatics J* 2013, 14:27–36. https://doi.org/10.1016/j.eij.2012.11.001.
11. Koster M. *A Standard for Robot Exclusion*. 1994. Available at: http://ftp.nada.kth.se/pub/hacks/src3/linkchecker/norobots-rfc.html.
12. Abiteboul S. Querying semi-structured data. In: *International Conference on Database Theory*. Berlin and Heidelberg: Springer; 1997, 1–18.
13. Olston C, Najork M. Web crawling. *Found Trends Inf Retr* 2010, 4:175–246. https://doi.org/10.1561/1500000017.
14. Turek W, Opalinski A, Kisiel-Dorohinicki M. Extensible Web crawler—towards multimedia material analysis. In: *Multimedia Communications Services and Security*. Dziech A, Czyżewski A, eds. Berlin and Heidelberg: Springer; 2011, 183–190. https://doi.org/10.1007/978-3-642-21512-4_22.
15. Zheng Q, Wu Z, Cheng X, Jiang L, Liu J. Learning to crawl deep Web. *Inf Syst* 2013, 38:801–819. https://doi.org/10.1016/j.is.2013.02.001.
16. Arasu A, Cho J, Garcia-Molina H, Paepcke A, Raghavan S. Searching the Web. *ACM Trans Internet Technol* 2001, 1:2–43. https://doi.org/10.1145/383034.383035.
17. Rungsawang A, Suebchua T, Manaskasemsak B. Thai related foreign language-specific website segment crawler. In: *2014—The 28th IEEE International Conference on Advanced Information Networking and Applications*, 2014, 293–298. https://doi.org/10.1109/WAINA.2014.56.
18. Abbasi A, Fu T, Zeng D, Adjeroh D. Crawling credible online medical sentiments for social intelligence. In: *2013 International Conference on IEEE Computer Society*, 2013, 254–263. https://doi.org/10.1109/SocialCom.2013.43.
19. Achsan HTY, Wibowo WC. A fast distributed focused-Web crawling. *Procedia Eng* 2014, 69:492–499. https://doi.org/10.1016/j.proeng.2014.03.017.
20. Agarwal S, Sureka A. A focused crawler for mining hate and extremism promoting videos on YouTube. In: *Proceedings of the 25th ACM Conference on Hypertext and social media—HT '14*, New York, NY, ACM Press, 2014, 294–296. https://doi.org/10.1145/2631775.2631776.
21. Ahlers D, Boll S. Adaptive geospatially focused crawling. In: *Proceedings of the 18th ACM Conference on Information Knowledge Management—CIKM '09*, New York, NY, ACM Press, 2009: 445. https://doi.org/10.1145/1645953.1646011.
22. Ahmadi-Abkenari F, Selamat A. An architecture for a focused trend parallel Web crawler with the application of clickstream analysis. *Inf Sci* 2012, 184:266–281. https://doi.org/10.1016/j.ins.2011.08.022.

23. Almpanidis G, Kotropoulos C, Pitas I. Combining text and link analysis for focused crawling—an application for vertical search engines. *Inf Syst* 2007, 32:886–908. https://doi.org/10.1016/j.is.2006.09.004.
24. Altingovde IS, Ulusoy O. Exploiting interclass rules for focused crawling. *IEEE Intell Syst* 2004, 19:66–73. https://doi.org/10.1109/MIS.2004.62.
25. Altingovde IS, Ozcan R, Cetintas S, Yilmaz H, Ulusoy Ö. An automatic approach to construct domain-specific Web portals. In: *Proceedings of the Sixth ACM Conference on Information Knowledge Management—CIKM '07*, New York, NY, ACM Press, 2007, 849. https://doi.org/10.1145/1321440.1321558.
26. Avraam I, Anagnostopoulos I. A comparison over focused Web crawling strategies. In: *15th Panhellenic Conference on Informatics*, 2011, 245–249. https://doi.org/10.1109/PCI.2011.53.
27. Babaria R, Nath JS, Krishnan S, Sivaramakrishnan KR, Bhattacharyya C, Murty MN. Focused crawling with scalable ordinal regression solvers. In: *24th International Conference on Machine Learning—ICML '07*, New York, NY, ACM Press, 2007, 57–64. https://doi.org/10.1145/1273496.1273504.
28. Barbosa L, Bangalore S. Focusing on novelty a crawling strategy to build diverse language models. In: *Proceedings of the 20th ACM International Conference on Information Knowledge Management—CIKM '11*, New York, NY, ACM Press, 2011, 755. https://doi.org/10.1145/2063576.2063687.
29. Barros R, Rodrigues Nt JA, Xexéo GB, de Souza JM. A collaborative approach to build evaluated web page datasets. *Futur Gener Comput Syst* 2011, 27:119–126. https://doi.org/10.1016/j.future.2010.06.007.
30. Batzios A, Dimou C, Symeonidis AL, Mitkas PA. BioCrawler: an intelligent crawler for the semantic Web. *Expert Syst Appl* 2008, 35:524–530. https://doi.org/10.1016/j.eswa.2007.07.054.
31. Baykan E, Henzinger M, Weber I. A comprehensive study of techniques for URL-based web page language classification. *ACM Trans Web* 2013, 7:1–37. https://doi.org/10.1145/2435215.2435218.
32. Bedi P, Thukral A, Banati H, Behl A, Mendiratta V. A multi-threaded semantic focused crawler. *J Comput Sci Technol* 2012, 27:1233–1242. https://doi.org/10.1007/s11390-012-1299-8.
33. Boanjak M, Oliveira E, Martins J, Mendes Rodrigues E, Sarmento L. TwitterEcho—a distributed focused crawler to support open research with twitter data. In: *Proceedings of the 21st International Conference Companion on World Wide Web—WWW '12 Companion*. New York, NY, ACM Press, 2012, 1233. https://doi.org/10.1145/2187980.2188266.
34. Caliskan K, Ozcan R. Comparing classification methods for link context based focused crawlers. In: *2013 International Conference on Electronics, Computer and Computation (ICECCO)*, IEEE, 2013, 143–146. https://doi.org/10.1109/ICECCO.2013.6718249.
35. Campos R, Rojas O, Marin M, Mendoza M. Distributed ontology-driven focused crawling. In: *Proceedings of the 2013 21st Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP 2013)*, IEEE, 2013, 108–115. https://doi.org/10.1109/PDP.2013.23.
36. Chen D, Liying F, Jianzhuo Y, Shi B. Semantic focused crawler based on Q-learning and Bayes classifier. In: *2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2010)*, IEEE, 2010, 420–423. https://doi.org/10.1109/ICCSIT.2010.5563878.
37. Chen R, Desai BC. An enhanced Web robot for the CINDI system. In: *Proceedings of the C3S2E '08 Canadian Conference on Computer Science & Software Engineering*, Montreal, QC, Canada, May 12 – 13, 2008. New York, NY, ACM Press, 2008, 133. https://doi.org/10.1145/1370256.1370278.
38. Chen X, Zhang X. HAWK: a focused crawler with content and link analysis. In: *2008 I.E. International Conference on e-Business Engineering*, IEEE, 2008, 677–680. https://doi.org/10.1109/ICEBE.2008.46.
39. Chen Z, Ma J, Han X, Zhang D. An effective relevance prediction algorithm based on hierarchical taxonomy for focused crawling. In: *Information Retrieval Technology*, Berlin and Heidelberg, Springer, 2008, 613–619. https://doi.org/10.1007/978-3-540-68636-1_72.
40. Chen Z, Liu J, Zhai H, Jiang L, Cao B. Web Page Recognition Algorithm Based on Link Analysis in Theme Search Engine. In: *2012 Second International Conference on Cloud and Green Computing*, IEEE, 2012, 405–409. https://doi.org/10.1109/CGC.2012.42.
41. Cheng Q, Beizhan W, Pianpian W. Efficient focused crawling strategy using combination of link structure and content similarity. In: *2008 I.E. International Symposium on IT Medical Education*, 2008, 1045–1048. https://doi.org/10.1109/ITME.2008.4744029.
42. Wu C, Hou W, Shi Y, Liu T. A Web search contextual crawler using ontology relation mining. In: *2009 International Conference on Computational Intelligence and Software Engineering*, IEEE, 2009, 1–4. https://doi.org/10.1109/CISE.2009.5365842.
43. Cho J. Efficient crawling through URL ordering. *Comput Networks ISDN Syst* 1998, 30:161–172. https://doi.org/10.1016/S0169-7552(98)00108-1.
44. Chy AN. Bangla news classification using Naive Bayes classifier. In: *16th International Conference on Computing and Information Technology*, 2014, 8–10. https://doi.org/10.1109/ICCITechn.2014.6997369.
45. De Groc C. Babouk: focused Web crawling for corpus compilation and automatic terminology extraction. In: *Proceedings of the 2011 IEEE/WIC/ACM*

International Conference on Web Intelligence—WI 2011, 2011, 1, 497–498. https://doi.org/10.1109/WI-IAT.2011.253.

46. Dong H, Hussain FK. Focused crawling for automatic service discovery, annotation, and classification in industrial digital ecosystems. *IEEE Trans Ind Electron* 2011, 58:2106–2116. https://doi.org/10.1109/tie.2010.2050754.
47. Dong H, Hussain FK. SOF: a semi-supervised ontology-learning-based focused crawler. *Concurr Comput Pract Exp* 2013, 25:1755–1770. https://doi.org/10.1002/cpe.2980.
48. Dong H, Hussain FK. Self-adaptive semantic focused crawler for mining services information discovery. *IEEE Trans Ind Informatics* 2014, 10:1616–1626. https://doi.org/10.1109/TII.2012.2234472.
49. Dong H, Hussain FK, Chang E. A transport service ontology-based focused crawler. In: *2008 Fourth International Conference on Semantics, Knowledge and Grid*, IEEE, 2008, 49–56. https://doi.org/10.1109/SKG.2008.56.
50. Dong H, Hussain FK, Chang E. A framework for discovering and classifying ubiquitous services in digital health ecosystems. *J Comput Syst Sci* 2011, 77:687–704. https://doi.org/10.1016/j.jcss.2010.02.009.
51. Dong Q. Search-engine-oriented theme crawler design. In: *2010 International Conference on Systems Science, Engineering Design and Manufacturing Informatization*, IEEE, 2010, 303–306. https://doi.org/10.1109/ICSEM.2010.169.
52. Du Y, Pen Q, Gao Z. A topic-specific crawling strategy based on semantics similarity. *Data Knowl Eng* 2013, 88:75–93. https://doi.org/10.1016/j.datak.2013.09.003.
53. Du Y, Hai Y, Xie C, Wang X. An approach for selecting seed URLs of focused crawler based on user-interest ontology. *Appl Soft Comput J* 2014, 14:663–676. https://doi.org/10.1016/j.asoc.2013.09.007.
54. Fan H, Zeng G, Li X. Crawling strategy of focused crawler based on niche genetic algorithm. In: *Proceedings of the 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, IEEE, 2009, 591–594. https://doi.org/10.1109/DASC.2009.49.
55. Filipowski K. Comparison of scheduling algorithms for domain specific Web crawler. In: *2014 Eur. Netw. Intell. Conf.*, IEEE, 2014, 69–74. https://doi.org/10.1109/ENIC.2014.14.
56. Fu T, Abbasi A, Zeng D, Chen H. Sentimental Spidering Leveraging Opinion Information in Focused Crawlers. *ACM Trans Inf Syst* 2012, 30:1–30. https://doi.org/10.1145/2382438.2382443.
57. Gao K, Yonggen Gu. Analyzing an agent-based selective information retrieval. In: *IEEE International Conference on Services Computing 2004*. (*SCC 2004*). *Proceedings 2004*, IEEE, 2004, 427–430. https://doi.org/10.1109/SCC.2004.1358035.
58. Gao W, Lee HC, Miao Y. Geographically focused collaborative crawling. In: *Proceedings of the 15th international conference on World Wide Web—WWW '06*, New York, NY, ACM Press, 2006, 287. https://doi.org/10.1145/1135777.1135822.
59. Gao Z, Du Y, Yi L, Peng Q, Yang Y. Incrementally updating concept context graph (CCG) for focused Web crawling based on FCA. In: *2009 Asia-Pacific Conference on Information Processing*, IEEE, 2009, 40–43. https://doi.org/10.1109/APCIP.2009.146.
60. Ghozia A, Sorour H, Aboshosha A. Improved focused crawling using bayesian object based approach. In: *2008 National Radio Science Conference* (*NRSC*), IEEE, 2008, 1–8. https://doi.org/10.1109/NRSC.2008.4542363.
61. Gonzlez I, Marcus A, Meredith DN, Nguyen LA. Effective Web-scale crawling through website analysis. In: *Proceedings of the 15th international conference on World Wide Web—WWW '06*, New York, NY, ACM Press, 2006, 1041. https://doi.org/10.1145/1135777.1136005.
62. Gouriten G, Maniu S, Senellart P. Scalable, generic, and adaptive systems for focused crawling. In: *Proceedings of the 25th ACM conference on Hypertext and Social Media—HT '14*, New York, NY, ACM Press, 2014, 35–45. https://doi.org/10.1145/2631775.2631795.
63. Guerriero A, Ragni F, Martines C. A dynamic URL assignment method for parallel Web crawler. In: *2010 I.E. International Conference on Computer Intelligent Measurement Systems and Application*, IEEE, 2010, 119–123. https://doi.org/10.1109/CIMSA.2010.5611764.
64. Hao H, Mu C, Yin X, Li S, Wang Z. An improved topic relevance algorithm for focused crawling. In: *2011 I.E. International Conference on Systems, Man and Cybernetics—SMC*, IEEE, 2011, 850–855. https://doi.org/10.1109/ICSMC.2011.6083759.
65. Hijazi HW, Itmazi JA. Crawler based context aware model for distributed e-courses through ubiquitous computing at higher education institutes. In: *2013 Fourth International Conference on e-Learning "Best Practices in Management, Design and Development of e-Courses: Standards of Excellence and Creativity*," IEEE, 2013, 9–14. https://doi.org/10.1109/ECONF.2013.28.
66. Liu H, Milios E. Probabilistic models for focused Web crawling. *Comput Intell* 2012, 28:289–328. https://doi.org/10.1111/j.1467-8640.2012.00411.x.
67. Hu K, Wong WS. A probabilistic model for intelligent Web crawlers. In: *Proceedings of the 27th Annu. Int. Comput. Softw. Appl. Conf. COMPAC 2003*, *IEEE*

Comput. Soc, 2003, 278–282. https://doi.org/10.1109/CMPSAC.2003.1245354.

68. Huang R, Lin F. Focused crawling with heterogeneous semantic information. In: *International Conference of the Web Intell. Intell. Agent Technol. WI-IAT*, 2008, 525–531. https://doi.org/10.1109/WIIAT.2008.87.
69. Huang W, Zhang L, Zhang J, Zhu M. Focused crawling for retrieving e-commerce information based on learnable ontology and link prediction. In: *2009 Int. Symp. Inf. Eng. Electron. Commer.*, IEEE, 2009, 574–579. https://doi.org/10.1109/IEEC.2009.127.
70. Huang X, Zhou L, Wang C. Design and implementation of digital products vertical search engine based on android client. *2013 Information Science and Technology International Conference* 2013, 828–831. https://doi.org/10.1109/ICIST.2013.6747669.
71. Huang Y, Ye Y. wHunter: a focused Web crawler – a tool for digital library. In: Chen Z, Chen H, Miao Q, Fu Y, Fox E, Lim E, eds. *Lecture Notes in Computer Science (LNCS) 3334*. Berlin Heidelberg: Springer; 2004, 519–522. https://doi.org/10.1007/978-3-540-30544-6_59.
72. Jamali M, Sayyadi H, Hariri B, Abolhassani H. A method for focused crawling using combination of link structure and content similarity. In: *2006 IEEE/WIC/ACM International Conf. Web Intell. (WI 2006 Main Conf. Proceedings)(WI'06)*, IEEE, 2006, 753–756. https://doi.org/10.1109/WI.2006.19.
73. Jannach D, Shchekotykhin K, Friedrich G. Automated ontology instantiation from tabular Web sources—the allright system⋆. *Web semantics: science, services and agents on World Wide Web* 2009, 7:136–153. https://doi.org/10.1016/j.websem.2009.04.002.
74. Jung JJ. Towards open decision support systems based on semantic focused crawling. *Expert Syst Appl* 2009, 36:3914–3922. https://doi.org/10.1016/j.eswa.2008.02.057.
75. Ji L, Yan J, Liu N, Zhang W, Fan W, Chen Z. ExSearch. In: *Proceeding 18th ACM Conference on Information Knowledge Management.—CIKM '09*, New York, NY, ACM Press, 2009, 1357. https://doi.org/10.1145/1645953.1646125.
76. Jiang J, Song X, Yu N, Lin C. FoCUS: learning to crawl Web forums. *IEEE Trans Knowl Data Eng* 2013, 25:1293–1306. https://doi.org/10.1109/TKDE.2012.56.
77. Khalilian M, Boroujeni FZ, Mustapha N. Improving performance in constructing specific Web directory using focused crawler: an experiment on botany domain. In: Elleithy K, ed. *Advanced Technology in Computer Science and Software Engineering*. Dordrecht: Springer Netherlands; 2010, 461–466. https://doi.org/10.1007/978-90-481-3660-5_79.
78. Kozanidis L. An ontology-based focused crawler. In: Kapetanios E, Sugumaran V, Spiliopoulou M, eds. *Natural Language Processing and Information Systems*. Berlin, Heidelberg: Springer; 2008, 376–379. https://doi.org/10.1007/978-3-540-69858-6_48.
79. Kumar M, Vig R. Learnable focused meta crawling through Web. *Procedia Technol* 2012, 6:606–611. https://doi.org/10.1016/j.protcy.2012.10.073.
80. Kumar M, Vig R. Term-frequency inverse-document frequency definition semantic (TIDS) based focused Web crawler. In: Krishna VP, Babu Rajasekhara M, Ariwa E, eds. *Global Trends in Information Systems and Software Applications, Communications in Computer and Information Science*. Berlin and Heidlberg: Springer; 2012, 31–36. https://doi.org/10.1007/978-3-642-29216-3_5.
81. Lawless S, Hederman L, Wade V. OCCS: enabling the dynamic discovery, harvesting and delivery of educational content from open corpus sources. In: *2008 Eighth IEEE International Conference of the Adv. Learn. Technol.*, IEEE, 2008, 676–678. https://doi.org/10.1109/ICALT.2008.28.
82. Li J, Furuse K, Yamaguchi K. Focused crawling by exploiting anchor text using decision tree. In: *Spec. Interes. Tracks Posters 14th International Conference on World Wide Web—WWW '05*, New York, NY, ACM Press, 2005, 1190. https://doi.org/10.1145/1062745.1062933.
83. Li X, Xing M, Zhang J. A comprehensive prediction method of visit priority for focused crawler. In: *2011 2nd Int. Symp. Intell. Inf. Process. Trust. Comput.*, IEEE, 2011, 27–30. https://doi.org/10.1109/IPTC.2011.14.
84. Li Y, Wang Y, Du J. E-FFC: an enhanced form-focused crawler for domain-specific deep Web databases. *J Intell Inf Syst* 2013, 40:159–184. https://doi.org/10.1007/s10844-012-0221-8.
85. Li Y, Wang Y, Tian E. A new architecture of an intelligent agent-based crawler for domain-specific deep Web databases. In: *2012 IEEE/WIC/ACM International Conf. Web Intell. Intell. Agent Technol.*, IEEE, 2012, 656–663. https://doi.org/10.1109/WI-IAT.2012.103.
86. Liu H, Milios E, Janssen J. Focused crawling by learning hmm from user's topic-specific browsing. In: *IEEE/WIC/ACM Int. Conf. Web Intell.*, IEEE, 2004, 732–732. https://doi.org/10.1109/WI.2004.10057.
87. Liu H, Janssen J, Milios E. Using HMM to learn user browsing patterns for focused Web crawling. *Data Knowl Eng* 2006, 59:270–291. https://doi.org/10.1016/j.datak.2006.01.012.
88. Luo N, Zuo W, Yuan F, Zhang C. A new method for focused crawler cross tunnel. In: *First Int. Conf. Rough Sets Knowl. Technol.*, Berlin and Heidelberg, Springer, 2006, 632–637. https://doi.org/10.1007/11795131_92.

89. Luong HP, Gauch S, Wang Q. Ontology-based focused crawling. In: *2009 International Conference of the Information, Process. Knowl. Manag.*, IEEE, 2009, 123–128. https://doi.org/10.1109/eKNOW.2009.26.

90. Van de Maele F, Spyns P, Meersman R. An ontology-based crawler for the semantic Web. In: Meersman R, Tari Z, Herrero P, eds. *Expert Systems with Applications*. Berlin and Heidelberg: Springer; 2008, 1056–1065. https://doi.org/10.1007/978-3-540-88875-8_133.

91. Makris C, Panagis Y, Sakkopoulos E, Tsakalidis A. Category ranking for personalized search. *Data Knowl Eng* 2007, 60:109–125. https://doi.org/10.1016/j.datak.2005.11.006.

92. Mali S, Meshram BB. Focused Web crawler with revisit policy. In: *Proceedings of the International Conference of the Work. Emerg. Trends Technol.—ICWET '11*, New York, New York, ACM Press, 2011, 474. https://doi.org/10.1145/1980022.1980125.

93. Mangaravite V, Assis GT, Ferreira AA. Improving the efficiency of a genre-aware approach to focused crawling based on link context. In: *2012 Eighth Lat. Am. Web Congr.*, IEEE, North Latin American, 2012, 17–23. https://doi.org/10.1109/LA-WEB.2012.24.

94. Menczer F, Pant G, Srinivasan P, Ruiz ME. Evaluating topic-driven Web crawlers. In: *Proceedings of the 24th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.—Special Interest Group on Information Retrieval '01*, New York, NY, ACM Press, 2001, 241–249. https://doi.org/10.1145/383952.383995.

95. Meusel R, Mika P, Blanco R. Focused crawling for structured data. In: *Proceedings of the 23rd ACM International Conf. Conference on Information Knowledge Management.—CIKM '14*, New York, NY, ACM Press, 2014, 1039–1048. https://doi.org/10.1145/2661829.2661902.

96. Kc M, Hagenbuchner M, Tsoi AC. Quality information retrieval for the world wide Web. In: *2008 IEEE/WIC/ACM International Conf. Web Intell. Intell. Agent Technol.*, IEEE, 2008, 655–661. https://doi.org/10.1109/WIIAT.2008.378.

97. Navrat P, Jastrzembska L, Jelinek T. Bee hive at work: story tracking case study. In: *2009 IEEE/WIC/ACM International Jt. Conf. Web Intell. Intell. Agent Technol.*, IEEE, 2009, 117–120. https://doi.org/10.1109/WI-IAT.2009.244.

98. Neunerdt M, Niermann M, Mathar R, Trevisan B. Focused crawling for building Web comment corpora. In: *2013 I.E. 10th Consum. Commun. Netw. Conf.*, IEEE, 2013, 685–688. https://doi.org/10.1109/CCNC.2013.6488526.

99. Nhan NQ, Son VT, Binh HTT, Khanh TD. Crawl topical vietnamese Web pages using genetic algorithm. In: *2010 Second International Conference of the Knowl. Syst. Eng.*, IEEE, 2010, 217–223. https://doi.org/10.1109/KSE.2010.25.

100. Ning H, Wu H, He Z, Tan Y. Focused crawler URL analysis model based on improved genetic algorithm, *2011 I.E. Int. Conf. Mechatronics Autom.*, 2011, 2159–2164. https://doi.org/10.1109/ICMA.2011.5986315.

101. Özel SA. A web page classification system based on a genetic algorithm using tagged-terms as features. *Expert Syst Appl* 2011, 38:3407–3415. https://doi.org/10.1016/j.eswa.2010.08.126.

102. Özmutlu HC, Özmutlu S. An architecture for SCS: a specialized Web crawler on the topic of security. *Proceedings of the Am Soc Inf Sci Technol* 2005, 41:317–326. https://doi.org/10.1002/meet.1450410138.

103. Pant G, Srinivasan P. Learning to crawl: comparing classification scheme. *ACM Trans Inf Syst* 2005, 23:430–462. https://doi.org/10.1145/1095872.1095875.

104. Peng L, Wen-Da T. A focused Web crawler face stock information of financial field. In: *2010 I.E. Int. Conf. Intell. Comput. Intell. Syst.*, IEEE, 2010, 512–516. https://doi.org/10.1109/ICICISYS.2010.5658277.

105. Peng T, Liu L. Focused crawling enhanced by CBP–SLC. *Knowledge-Based Syst* 2013, 51:15–26. https://doi.org/10.1016/j.knosys.2013.06.008.

106. Peng T, Zhang C, Zuo W. Tunneling enhanced by web page content block partition for focused crawling. *Concurr Comput Pract Exp* 2008, 20:61–74. https://doi.org/10.1002/cpe.1211.

107. Pesaranghader A, Pesaranghader A, Mustapha N, Sharef NM. Improving multi-term topics focused crawling by introducing term Frequency-Information Content (TF-IC) measure. In: *2013 Int. Conf. Res. Innov. Inf. Syst.*, IEEE, 2013, 102–106. https://doi.org/10.1109/ICRIIS.2013.6716693.

108. Priyatam PN, Vaddepally SR, Varma V. Domain specific search in indian languages. In: *Proceedings of the First Work. Inf. Knowl. Manag. Dev. Reg.—IKM4DR '12*, New York, NY, ACM Press, 2012, 23. https://doi.org/10.1145/2389776.2389782.

109. Putra WE, Akbar S. Focused crawling using dictionary algorithm with breadth first and by page length methods for Javanese and Sundanese corpus construction. *Procedia Technol* 2013, 11:870–876. https://doi.org/10.1016/j.protcy.2013.12.270.

110. Qin J, Chen H. Using genetic algorithm in building domain-specific collections: an experiment in the nanotechnology domain. In: *Proceedings of the 38th Annu. Hawaii International Conference of the Syst. Sci.*, IEEE, 2005, 102b–102b. https://doi.org/10.1109/HICSS.2005.659.

111. Qin JQJ, Zhou Y, Chau M. Building domain-specific Web collections for scientific digital libraries: a meta-search enhanced focused crawling method. *Proceedings of the 2004 Jt. ACM/IEEE Conf. Digit. Libr. 2004*, 2004, 135–141. https://doi.org/10.1109/JCDL.2004.1336110.

112. Radu I, Rebedea T. A focused crawler for Romanian words discovery. In: *2014 RoEduNet Conf. 13th Ed. Netw. Educ. Res. Jt. Event RENAM 8th Conf.*, IEEE, 2014, 1–6. https://doi.org/10.1109/RoEduNet-RENAM.2014.6955323.

113. Ravakhah M, Kamyar M. Semantic similarity based focused crawling. In: *2009 First International Conference of the Comput. Intell. Commun. Syst. Networks*, IEEE, 2009, 448–453. https://doi.org/10.1109/CICSYN.2009.92.

114. Rocco D, Caverlee J, Liu L, Critchlow T. Domain-specific Web service discovery with service class descriptions. In: *IEEE Int. Conf. Web Serv.*, IEEE, 2005, 1–8. https://doi.org/10.1109/ICWS.2005.49.

115. Safran MS, Althagafi A, Che D. Improving relevance prediction for focused Web crawlers. In: *2012 IEEE/ACIS 11th International Conference of the Comput. Inf. Sci.*, IEEE, 2012, 161–166. https://doi.org/10.1109/ICIS.2012.61.

116. Samarawickrama S, Jayaratne L. Automatic text classification and focused crawling. In: *2011 Sixth Int. Conf. Digit. Inf. Manag.*, IEEE, 2011, 143–148. https://doi.org/10.1109/ICDIM.2011.6093329.

117. Schuh G, Brakling Apfel AK. Identification of requirements for focused crawlers in technology intelligence. In: *Portland International Conference on Management of Engineering & Technology (PICMET)*, 2014, 2918–2923.

118. Selamat A, Ahmadi-Abkenari F. Application of clickstream analysis as Web page importance metric in parallel crawlers. In: *2010 Int. Symp. Inf. Technol.*, IEEE, 2010, 1–6. https://doi.org/10.1109/ITSIM.2010.5561354.

119. Selamat A, Ahmadi-Abkenari F. Architecture for a parallel focused crawler for clickstream analysis. In: *Intelligent Information and Database Systems*. Nguyen NT, Kim C-G, Janiak A, eds. Berlin and Heidelberg: Springer; 2011, 27–35. https://doi.org/10.1007/978-3-642-20039-7_3.

120. Shchekotykhin K, Jannach D, Friedrich G. xCrawl: a high-recall crawling method for Web mining. In: *2008 Eighth IEEE International Conference of the Data Min.*, IEEE, 2008, 550–559. https://doi.org/10.1109/ICDM.2008.121.

121. Yang S-Y. A focused crawler with ontology-supported website models for information agents. *Expert Syst Appl* 2010, 37:5381–5389. https://doi.org/10.1016/j.eswa.2010.01.018.

122. Shokouhi M, Chubak P, Raeesy Z. Enhancing focused crawling with genetic algorithms. In: *International Conference of the Inf. Technol. Coding Comput,—Vol. II*, IEEE, 2005, Vol. 2, 503–508. https://doi.org/10.1109/ITCC.2005.145.

123. Sirisha Gadiraju NVG, Krishna Chaitanya R, Padma Raju G. Effect of feature selection method on the performance of focused crawlers—a case study on traditional and accelerated focused crawlers. In: *2010 Int. Conf. Netw. Inf. Technol.*, IEEE, 2010, 482–487. https://doi.org/10.1109/ICNIT.2010.5508468.

124. Sizov S, Siersdorfer S, Theobald M, Weikum G. The BINGO! focused crawler: from bookmarks to archetypes. In: *Proceedings of the 18th International Conference of the Data Eng.*, IEEE Comput. Soc, 2002, 337–338. https://doi.org/10.1109/ICDE.2002.994746.

125. Su C, Gao Y, Yang J, Luo B. An efficient adaptive focused crawler based on ontology learning. In: *Fifth International Conference of the Hybrid Intell. Syst.*, IEEE, 2005, 6 pp. https://doi.org/10.1109/ICHIS.2005.19.

126. Sun Y, Jin P, Yue L. A framework of a hybrid focused Web crawler. In: *2008 Second International Conference of the Futur. Gener. Commun. Netw. Symp.*, IEEE, 2008, 50–53. https://doi.org/10.1109/FGCNS.2008.73.

127. Tang TT, Hawking D, Craswell N, Griffiths K. Focused crawling for both topical relevance and quality of medical information. In: *Proceedings of the* 14th ACM International *Conference on Information Knowledge Management.*—CIKM '05, New York, NY, ACM Press, 2005, 147. https://doi.org/10.1145/1099554.1099583.

128. Thukral A, Mendiratta V, Behl A, Banati H, Bedi P. FCHC: a social semantic focused crawler. In: *Advances in Computing and Communications*. Abraham A, Mauri JL, Buford JF, Suzuki J, Thampi SM, eds. Berlin and Heidelberg: Springer; 2011, 273–283. https://doi.org/10.1007/978-3-642-22714-1_29.

129. Tsay J-J, Shih C-Y, Wu B-L. AuToCrawler: an integrated system for automatic topical crawler. In: *Fourth Annu. ACIS International Conference of the Comput. Inf. Sci.*, IEEE, 2005, 462–467. https://doi.org/10.1109/ICIS.2005.33.

130. Uzun E, Serdar Güner E, Kılıçaslan Y, Yerlikaya T, Agun HV. An effective and efficient Web content extractor for optimizing the crawling process. *Softw Pract Exp* 2014, 44:1181–1199. https://doi.org/10.1002/spe.2195.

131. Wang H, Wang X, Wang Y, Bhattacharjee A. , S.-K. Basireddy, A. Cherian, Preliminary study on design and development of a journal focused crawler system using EBD methodology: Part I; Design task and environment analysis. In: *Proceedings of the* 2014 Int. Conf. Innov. Des. Manuf., IEEE, 2014, 59–64. https://doi.org/10.1109/IDAM.2014.6912671.

132. Wang H, Wang X, Wang Y, Bhattacharjee A, Basireddy SK, Cherian A. A preliminary study on design and development of a journal focused crawler system using EBD methodology. Part II—conflict identification and solution generation. In: *Proceedings of the 2014 Int. Conf. Innov. Des. Manuf.*, 2014, 123–128.

133. Wang N. Design and implementation of a crawling system in shopping search engine. In: *2009 Second Int. Work. Comput. Sci. Eng.*, IEEE, 2009, 212–216. https://doi.org/10.1109/WCSE.2009.798.

134. Wang W, Chen X, Zou Y, Wang H, Dai Z. A focused crawler based on naive bayes classifier. In: *2010 Third Int. Symp. Intell. Inf. Technol. Secur. Informatics*, IEEE, 2010, 517–521. https://doi.org/10.1109/IITSI.2010.30.

135. Wei B, Liu J, Ma J, Zheng Q, Zhang W, Feng B. DFT-extractor: a system to extract domain-specific faceted taxonomies from Wikipedia. In: *Proceedings of the 22Nd Int. Conf. World Wide Web Companion*, *International World Wide Web Conferences Steering Committee*, Switzerland, Republic and Canton of Geneva, 2013, 277–280. http://dl.acm.org/citation.cfm?id=2487788.2487922 (accessed March 18, 2015).

136. Wu Y, Shou L, Hu T, Chen G. Query triggered crawling strategy: build a time sensitive vertical search engine. In: *2008 Int. Conf. Cyberworlds*, IEEE, 2008, 422–427. https://doi.org/10.1109/CW.2008.35.

137. Wu Z, Wu J, Khabsa M, Williams K, Chen H, Huang W, et al., Towards building a scholarly big data platform: challenges, lessons and opportunities. In: *IEEE/ACM Jt. Conf. Digit. Libr.*, IEEE, 2014, 117–126. https://doi.org/10.1109/JCDL.2014.6970157.

138. Xi S, Sun F, Wang J. A cognitive crawler using structure pattern for incremental crawling and content extraction. In: *Intergovernmental Panel on Climate Change (Ed.), 9th IEEE International Conference on Cognitive Informatics & Cognitive Computing*. Cambridge: IEEE; 2010, 238–244. https://doi.org/10.1109/COGINF.2010.5599733.

139. Xiang L, Meng X. A data mining approach to topic-specific Web resource discovery. In: *2009 Second International Conference of the Intell. Comput. Technol. Autom.*, IEEE, 2009, 595–599. https://doi.org/10.1109/ICICTA.2009.378.

140. Xin C, Yong Z, Fuyan Z, Changbao N. Architecture design of subject-oriented Web crawler. In: *2013 Fourth International Conference of the Intell. Syst. Des. Eng. Appl.*, IEEE, 2013, 174–177. https://doi.org/10.1109/ISDEA.2013.444.

141. Xu L, Eli S, Xu H. A method of focused crawling for software components. In: *Proceedings of the 2011 Int. Conf. Transp. Mech. Electr. Eng.*, IEEE, 2011, 1560–1563. https://doi.org/10.1109/TMEE.2011.6199506.

142. Xu Q, Zuo W. First-order focused crawling. In: *Proceedings of the 16th International Conference of the World Wide Web—WWW '07*, New York, NY, ACM Press, 2007, 1159. https://doi.org/10.1145/1242572.1242744.

143. Ya-jun D, Yang X, Zhan-shen L, Dong-mei Qi. Discussion on interest spider's algorithm of search engine. In: *Proceedings of the 2004 IEEE International Conference of Information Reuse and Integration. 2004. IRI 2004.*, IEEE, n.d., 588–593. https://doi.org/10.1109/IRI.2004.1431525.

144. Yang J, Kang J, Choi J. A focused crawler with document segmentation. In: *International Conference on Intelligent Data Engineering and Automated Learning*, 2005, 94–101. https://doi.org/10.1007/11508069_13.

145. Yang S-Y. An ontological website models-supported search agent for Web services. *Expert Syst Appl* 2008, 35:2056–2073. https://doi.org/10.1016/j.eswa.2007.09.024.

146. Yifeng C, Hengkai Z, Xiaoqing Y, Wanggen W. Research of theme crawling strategy based on genetic algorithm. In: *IET Int. Commun. Conf. Wirel. Mob. Comput. (CCWMC 2009)*, IET, 2009, 472–475. https://doi.org/10.1049/cp.2009.1993.

147. Ying L, Zhou X, Yuan J, Huang Y. A novel focused crawler based on breadcrumb navigation. *Adv Swarm Intell* 2012, 7332:264–271. https://doi.org/10.1007/978-3-642-31020-1_31.

148. Yuan F, Yin C, Liu J. Improvement of pagerank for focused crawler. *Proceedings of the Eighth ACIS International Conference of the Softw. Eng. Artif. Intell. Netw. Parallel Distributed Comput.* 2007, Vol. 02. 3, 797–802. https://doi.org/10.1109/SNPD.2007.314.

149. Yuan F, Yin C, Liu J, Zhang Y. An integrated crawling strategy for domain-specific resource discovery. *2007 Third Int. IEEE Conf. Signal-Image Technol. Internet-Based Syst.*, 2007, 329–336. https://doi.org/10.1109/SITIS.2007.70.

150. Zhang XZX, Li ZLZ, Hu CHC. Adaptive focused crawler based on tunneling and link analysis. In: *2009 11th Int. Conf. Adv. Commun. Technol.*2009, 03, 2225–2230.

151. Zhang Z, Nasraoui O, Van Zwol R. Exploiting tags and social profiles to improve focused crawling. In: *2009 IEEE/WIC/ACM International Jt. Conf. Web Intell. Intell. Agent Technol.*, IEEE, 2009, 136–139. https://doi.org/10.1109/WI-IAT.2009.27.

152. ZHENG H, KANG B, KIM H. An ontology-based approach to learnable focused crawling. *Inf Sci* 2008, 178:4512–4522. https://doi.org/10.1016/j.ins.2008.07.030.

153. Zheng S. Genetic and ant algorithms based focused crawler design. In: *2011 Second International Conference of the Innov. Bio-Inspired Comput. Appl., IEEE*, 2011, 374–378. https://doi.org/10.1109/IBICA.2011.98.

154. Zheng X, Zhou T, Yu Z, Chen D. URL rule based focused crawler. In: *2008 IEEE International Conference of the e-Business Engineering* IEEE, 2008, 147–154. https://doi.org/10.1109/ICEBE.2008.61.

155. Zhou B, Xiao B, Lin Z, Zhang C. A distributed vertical crawler using crawling-period based strategy. *Proceedings of the 2010 2nd International Conference of the Futur. Comput. Commun. ICFCC 2010*, 2010, 1, 306–311. https://doi.org/10.1109/ICFCC.2010.5497780.

156. Zhu Q. An algorithm OFC for the focused Web crawler. *Proceedings of the Sixth International Conference of the Mach. Learn. Cybern. ICMLC 2007*, 2007, 7, 4059–4063. https://doi.org/10.1109/ICMLC.2007.4370856.

157. Zhuang Z, Wagle R, Giles CL. What's there and what's not? In: *Proceedings of the 5th ACM/IEEE-CS Joint conference on digital library JCDL '05*, New York, NY, ACM Press, 2005, 301. https://doi.org/10.1145/1065385.1065455.

158. Zunino R, Bisio F, Peretti C, Surlinelli R, Scillia E, Ottaviano A, et al., An analyst-adaptive approach to focused crawlers. In: *Proceedings of the 2013 IEEE/ACM International Advances in Social Networks Analysis and Mining—ASONAM '13*, New York, NY, ACM Press, 2013, 1073–1077. https://doi.org/10.1145/2492517.2500328.

159. Diligenti M, Coetzee FM, Lawrence S, Giles CL, Gori M. Focused crawling using context graphs. *26th Int. Conf. Very Large Databases*, 2000, 527–534.

160. Batsakis S, Petrakis EGM, Milios E. Improving the performance of focused Web crawlers. *Data Knowl Eng* 2009, 68:1001–1013. https://doi.org/10.1016/j.datak.2009.04.002.

161. Maimunah S, Widyantoro DH, Kuspriyanto, Sastramihardja HS. Co-citation & co-reference concepts to control focused crawler exploration. In: *Proceedings of the 2011 Int. Conf. Electr. Eng. Informatics*, IEEE, 2011, 1–7. https://doi.org/10.1109/ICEEI.2011.6021677.

162. Pappas N, Katsimpras G, Stamatatos E. An agent-based focused crawling framework for topic- and genre-related Web document discovery. In: *2012 I.E. 24th International Conference of the Tools with Artif. Intell.*, IEEE, 2012, 508–515. https://doi.org/10.1109/ICTAI.2012.75.

163. Cho J, Garcia-Molina H, Parallel crawlers. In: *Proceedings of the Elev. International Conference of the World Wide Web—WWW '02*, New York, NY, ACM Press, 2002, 124. https://doi.org/10.1145/511446.511464.

164. Akilandeswari J, Gopalan NP. A novel design of hidden Web crawler using reinforcement learning based agents. In: *Adv. Parallel Process. Technol.*, Berlin and Heidelberg, Springer, 2007, 433–440. https://doi.org/10.1007/978-3-540-76837-1_47.

165. Álvarez M, Raposo J, Pan A, Cacheda F, Bellas F, Carneiro V. DeepBot: a focused crawler for accessing hidden Web content. In: ACM *International Conference of the* Proceeding Ser. Vol. 236, New York, NY, ACM, 2007, 18. https://doi.org/10.1145/1278380.1278385.

166. An YJ, Geller J, Wu Y-T, Chun SA. Automatic generation of ontology from the deep Web. In: *18th International Conference of the Database Expert Syst. Appl. (DEXA 2007)*, IEEE, 2007, 470–474. https://doi.org/10.1109/DEXA.2007.43.

167. An YJ, Geller J, Wu Y-T, Chun SA. Semantic deep Web: automatic attribute extraction from the deep Web data sources. In: *Proceedings of the 2007 ACM Symp. Appl. Comput.—SAC '07*, New York, NY, ACM, 2007, 1667. https://doi.org/10.1145/1244002.1244355.

168. An YJ, Chun SA, Huang K, Geller J. Assessment for ontology-supported deep Web search. In: *2008 10th IEEE Conf. E-Commerce Technol. Fifth IEEE Conf. Enterp. Comput. E-Commerce E-Services*, IEEE, 2008, 382–388. https://doi.org/10.1109/CECandEEE.2008.117.

169. Arya KVV, Vadlamudi BR, An ontology-based topical crawling algorithm for accessing deep Web content. In: *2012 Third Int. Conf. Comput. Commun. Technol.*, 2012, 1–6. https://doi.org/10.1109/ICCCT.2012.10.

170. Barbosa L, Freire J. An adaptive crawler for locating hiddenwebentry points. In: *Proceedings of the 16th International Conference of the World Wide Web—WWW '07*, New York, NY, ACM Press, 2007, 441. https://doi.org/10.1145/1242572.1242632.

171. Bergholz A, Childlovskii B. Crawling for domain-specific hidden Web resources. In: *Proceedings of the 7th International Conference of the Prop. Appl. Dielectr. Mater.* (Cat. No.03CH37417), IEEE Comput. Soc, 2003, 125–133. https://doi.org/10.1109/WISE.2003.1254476.

172. Chandramouli A, Gauch S. A co-operative Web services paradigm for supporting crawlers. In: *Large Scale Semant. Access to Content* (*Text*, *Image*, *Video*, *Sound*), *LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE*, Paris, 2007, 475–489. https://doi.org/10.1.1.106.2411.

173. Cho J, Garcia-Molina H, Haveliwala T, Lam W, Paepcke A, Raghavan S, et al. Stanford WebBase components and applications. *ACM Trans Internet*

Technol 2006, 6:153–186. https://doi.org/10.1145/1149121.1149124.

174. El-desoky AI, Abd El-Gwad AO, Okasha ME. Exploiting ontology for retrieving data behind searchable Web forms. In: *2009 Int. Conf. Netw. Media Converg.*, IEEE, 2009, 97–102. https://doi.org/10.1109/ICNM.2009.4907197.
175. El-Desouky AI, Ali HA, El-Ghamrawy SM. A new framework for domain-specific hidden Web crawling based on data extraction techniques. In: *2006 ITI 4th International Conference of the Inf. Commun. Technol.*, IEEE, 2006, 1–1. https://doi.org/10.1109/ITICT.2006.358295.
176. El-desouky A, Ali H, El-ghamrawy S. An automatic label extraction technique for domain-specific hidden Web crawling (LEHW). In: *2006 Int. Conf. Comput. Eng. Syst.*, IEEE, 2006, 454–459. https://doi.org/10.1109/ICCES.2006.320490.
177. Fontes ADC, Silva FS. SmartCrawl: a new strategy for the exploration of the hidden Web. In: *Proceedings of the 6th Annu. ACM International Work. Web Inf. Data Manag.—WIDM '04*, New York, NY, ACM Press, 2004, 9. https://doi.org/10.1145/1031453.1031457.
178. Ipeirotis PG, Agichtein E, Jain P, Gravano L. Towards a query optimizer for text-centric tasks. *ACM Trans Database Syst* 2007, 32:21–es. https://doi.org/10.1145/1292609.1292611.
179. Ipeirotis PG, Gravano L, Sahami M. Probe, count, and classify. In: *Proceedings of the 2001 ACM SIGMOD International Conference of the Manag. Data—SIGMOD '01*, New York, NY, ACM Press, 2001, 67–78. https://doi.org/10.1145/375663.375671.
180. Li H, Guo M, Cai L, Yang Y. An incremental update strategy in deep Web. In: *2010 Sixth Int. Conf. Nat. Comput.*, IEEE, 2010, 131–134. https://doi.org/10.1109/ICNC.2010.5583330.
181. Liang H, Ren F, Zuo W. The preliminary process of modeling in deep Web information fusion system. In: *2009 Int. Forum Inf. Technol. Appl.*, IEEE, 2009, 723–726. https://doi.org/10.1109/IFITA.2009.27.
182. Liang H, Zuo W, Ren F, Sun C. Accessing deep Web using automatic query translation technique. In: *2008 Fifth International Conference of the Fuzzy Syst. Knowl. Discov.*, IEEE, 2008, 267–271. https://doi.org/10.1109/FSKD.2008.18.
183. Liang H, Zuo W, Ren F, Wang J. Translating query for deep Web using ontology. In: *2008 International Conference of the Comput. Sci. Softw. Eng.*, IEEE, 2008, 427–430. https://doi.org/10.1109/CSSE.2008.630.
184. Liu X, Maly K, Zubair M, Nelson ML. DP9: an OAI geteway service for Web crawlers. In: *Proceedings of the Second ACM/IEEE-CS Jt. Conf. Digit. Libr.—JCDL '02*, New York, NY, ACM Press, 2002, 283. https://doi.org/10.1145/544220.544284.
185. Ma W, Chen X, Shang W. Advanced deep Web crawler based on dom. In: *2012 Fifth Int. Jt. Conf. Comput. Sci. Optim.*, IEEE, 2012, 605–609. https://doi.org/10.1109/CSO.2012.138.
186. Madaan R, Dixit A, Sharma AK, Bhatia KK. A framework for domain specific incremental hidden Web crawler. *Int J Comput Sci Eng* 2010, 02:753–758.
187. Mesbah A, van Deursen A, Lenselink S. Crawling Ajax-based Web applications through dynamic analysis of user interface state changes. *ACM Trans Web* 2012, 6:1–30. https://doi.org/10.1145/2109205.2109208.
188. Moraes MC, Heuser CA, Moreira VP, Barbosa D. Prequery discovery of domain-specific query forms: a survey. *IEEE Trans Knowl Data Eng* 2013, 25:1830–1848. https://doi.org/10.1109/TKDE.2012.111.
189. Mundluru D, Xia X. Experiences in crawling deep Web in the context of local search. In: *Proceeding 2nd Int. Work. Geogr. Inf. Retr.—GIR '08*, New York, NY, ACM Press, 2008, 35. https://doi.org/10.1145/1460007.1460016.
190. Myllymaki J. Effective Web data extraction with standard XML technologies. *Comput Networks* 2002, 39:635–644. https://doi.org/10.1016/S1389-1286(02)00214-1.
191. Nguyen H, Kang EY, Freire J. Automatically extracting form labels. In: *2008 I.E. 24th International Conference of the Data Eng.*, IEEE, 2008, 1498–1500. https://doi.org/10.1109/ICDE.2008.4497602.
192. Nguyen H, Nguyen T, Freire J. Learning to extract form labels. *Proc VLDB Endow* 2008, 1:684–694. https://doi.org/10.14778/1453856.1453931.
193. Nguyen TH, Nguyen H, Freire J, PruSM: a prudent schema matching approach for Web forms. In: *Proceedings of the 19th ACM International Conference on Information Knowledge Management.—CIKM '10*, New York, NY, ACM Press, 2010, 1385. https://doi.org/10.1145/1871437.1871627.
194. Peisu X., Ke T, Qinzhen H. A framework of deep Web crawler. In: *2008 27th Chinese Control Conf.*, IEEE, 2008, 582–586. https://doi.org/10.1109/CHICC.2008.4604881.
195. Rajaraman A. Kosmix: high-performace topic exploration using the deep Web. *Proc VLDB Endow* 2009, 2:1524–1529. https://doi.org/10.14778/1687553.1687581.
196. Singh L, Sharma DK. An approach for accessing data from hidden Web using intelligent agent technology. In: *2013 3rd IEEE Int. Adv. Comput. Conf.*, IEEE, 2013, 800–805. https://doi.org/10.1109/IAdCC.2013.6514329.
197. Singh L, Sharma DK. An architecture for extracting information from hidden Web databases using intelligent agent technology through reinforcement learning. In: *2013 IEEE conference on Information &*

Communication Technologies IEEE, 2013, 292–297. https://doi.org/10.1109/CICT.2013.6558108.

198. Taylan D, Poyraz M, Akyokus S, Ganiz MC. Intelligent focused crawler: learning which links to crawl. In: *2011 Int. Symp. Innov. Intell. Syst. Appl.*, IEEE, 2011, 504–508. https://doi.org/10.1109/INISTA.2011.5946150.
199. Furche T, Gottlob G, Grasso G, Guo X, Orsi G, Schallhart C. The ontological key: automatically understanding and integrating forms to access the deep Web. *Very Large Databases Journal (VLDB)* 2013, 22:615–640. https://doi.org/10.1007/s00778-013-0323-0.
200. Wang X, Wang L, Wei G, Zhang D, Yang Y. Hidden Web crawling for SQL injection detection. In: *2010 3rd IEEE Int. Conf. Broadband Netw. Multimed. Technol.*, IEEE, 2010, 14–18. https://doi.org/10.1109/ICBNMT.2010.5704860.
201. Wang Z, Hu R, Hu J. Research of a traffic advisory system based on deep web. In: *2009 International Conference of the Commun. Softw. Networks*, IEEE, 2009, 537–540. https://doi.org/10.1109/ICCSN.2009.64.
202. Yan Z, Li Q, Dong Y, Ding Y. An ontology-based integration of Web query interfaces for house search. In: *2008 Int. Conf. Inf. Autom.*, IEEE, 2008, 190–194. https://doi.org/10.1109/ICINFA.2008.4607994.
203. Yu H, Guo J, Yu Z, Xian Y, Yan X. A novel method for extracting entity data from deep Web precisely. In: 26th *Chinese Control Decis. Conf.* (*2014 CCDC*), 2014, 5049–5053. https://doi.org/10.1109/CCDC.2014.6853078.
204. Ntoulas A, Pzerfos P, Cho JCJ. Downloading textual hidden Web content through keyword queries, *Proceedings of the 5th ACM/IEEE-CS Jt. Conf. Digit. Libr.* (*JCDL '05*), 2005, 100–109. https://doi.org/10.1145/1065385.1065407.
205. Zhang Z, Dong G, Peng Z, Yan Z. A framework for incremental deep Web crawler based on URL classification. In: Gong Z, Luo X, Chen J, Lei J, Wang FL, eds. *International Journal of Computational Science and Engineering*. Berlin and Heidelberg: Springer; 2011, 302–310. https://doi.org/10.1007/978-3-642-23982-3_37.
206. Huang Q, Li Q, Li H, Yan Z. An approach to incremental deep Web crawling based on incremental harvest model. *Procedia Eng* 2012, 29:1081–1087. https://doi.org/10.1016/j.proeng.2012.01.093.
207. Raghavan S, Garcia-Molina H. *Crawling the Hidden Web*. San Francisco, CA: Morgan Kaufmann Publishers Inc.; 2001. http://ilpubs.stanford.edu:8090/456/.
208. Aggarwal CC. Collaborative crawling: mining user experiences for topical resource discovery. In: *Proceedings of the Eighth ACM SIGKDD International Conference of the Knowl. Discov. Data Min.—KDD '02*, New York, NY, ACM Press, 2002, 423. https://doi.org/10.1145/775047.775108.
209. Aggarwal CC, Al-Garawi F, Yu PS. On the design of a learning crawler for topical resource discovery. *ACM Trans Inf Syst* 2001, 19:286–309. https://doi.org/10.1145/502115.502119.
210. Baykan E, Henzinger M, Marian L, Weber I. Purely URL-based topic classification. In: *Proceedings of the 18th International Conference of the World Wide Web—WWW '09*, New York, NY, ACM Press, 2009, 1109–1110. https://doi.org/10.1145/1526709.1526880.
211. Can AB, Baykal N. MedicoPort: a medical search engine for all. *Comput Methods Programs Biomed* 2007, 86:73–86. https://doi.org/10.1016/j.cmpb.2007.01.007.
212. Chung C, Clarke CLA. Topic-oriented collaborative crawling. In: *Proceedings of the Elev. Int. Conference on Information Knowledge Management.—CIKM '02*, New York, NY, ACM Press, 2002, 34. https://doi.org/10.1145/584792.584802.
213. Davison BD. Topical locality in the Web. In: *Proceedings of the 23rd Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.—SIGIR '00*, New York, NY, ACM Press, 2000, 272–279. https://doi.org/10.1145/345508.345597.
214. Greenwood M, Nenadic G. Lexical profiling of existing web directories to support fine-grained topic-focused Web crawling. In: *Proceedings of the 2008 BCS IRSG Conf. Corpus Profiling*, Swinton, British Computer Society, 2008. http://dl.acm.org/citation.cfm?id=2227976.2227982.
215. Hsu C-C, Wu F. Topic-specific crawling on the Web with the measurements of the relevancy context graph. *Inf Syst* 2006, 31:232–246. https://doi.org/10.1016/j.is.2005.02.007.
216. Huifu Z, Yaping Z, Ping L, Xiaolan Z. Research and implementation on topic crawler of rotating machinery fault knowledge. In: *Proceedings of the 2011 Int. Conf. Comput. Sci. Netw. Technol., IEEE*, 2011, 1464–1467. https://doi.org/10.1109/ICCSNT.2011.6182242.
217. Luo L, Wang R, Huang X, Chen Z. A novel shark-search algorithm for theme crawler. In: *WISM'12 Proceedings of the 2012 international conference on Web Information Systems and Mining*, 2013, 603–609. https://doi.org/10.1007/978-3-642-33469-6_75.
218. Menczer F, Pant G, Srinivasan P. Topical Web crawlers: evaluating adaptive algorithms. *ACM Trans Internet Technol* 2004, 4:378–419. https://doi.org/10.1145/1031114.1031117.
219. Mouton A, Marteau P. Exploiting routing information encoded into backlinks to improve topical crawling. In: *2009 International Conference of the Soft Comput. Pattern Recognit.*, IEEE, 2009, 659–664. https://doi.org/10.1109/SoCPaR.2009.129.
220. Mukherjea S. WTMS: a system for collecting and analyzing topic-specific Web information. *Comput*

Networks 2000, 33:457–471. https://doi.org/10.1016/S1389-1286(00)00035-9.

221. Noh S, Choi Y, Seo H, Choi K, Jung G. An Intelligent Topic-Specific Crawler Using Degree of Relevance. In: Yang ZR, Yin H, Everson RM, eds. *3177 LNCS*. Berlin and Heidelberg: Springer; 2004, 491–498. https://doi.org/10.1007/978-3-540-28651-6_72.
222. Pant G, Srinivasan P. Link contexts in classifier-guided topical crawlers. *IEEE Trans Knowl Data Eng* 2006, 18:107–122. https://doi.org/10.1109/TKDE.2006.12.
223. Pant GPG, Tsioutsiouliklis K, Johnson J, Giles CL. Panorama: extending digital libraries with topical crawlers, *Proceedings of the 2004 Jt. ACM/IEEE Conf. Digit. Libr. 2004*, 2004, 142–150. https://doi.org/10.1109/JCDL.2004.1336111.
224. Peng Q, Du Y, Hai Y, Chen S, Gao Z. Topic-Specific Crawling on the Web with Concept Context Graph Based on FCA, in: *2009 International Conference of the Manag. Serv. Sci.*, IEEE, 2009, 1–4. https://doi.org/10.1109/ICMSS.2009.5302301.
225. Pesaranghader A, Mustapha N, Pesaranghader A. Applying semantic similarity measures to enhance topic-specific Web crawling. In: *2013 13th Int. Conf. Intellient Syst. Des. Appl.*, *IEEE*, New York, NY, 2013, 205–212. https://doi.org/10.1109/ISDA.2013.6920736.
226. Qian R, Zhang K, Zhao G. A topic-specific Web crawler based on content and structure mining. In: *Proceedings of the 2013 3rd Int. Conf. Comput. Sci. Netw. Technol.*, IEEE, 2013, 458–461. https://doi.org/10.1109/ICCSNT.2013.6967153.
227. Rungsawang A, Angkawattanawit N. Learnable topic-specific Web crawler. *J Netw Comput Appl* 2005, 28:97–114. https://doi.org/10.1016/j.jnca.2004.01.001.
228. Saha S, Murthy CA, Pal SK. Rough set based ensemble prediction for topic specific Web crawling. In: *2009 Seventh International Conference of the Adv. Pattern Recognit.*, IEEE, 2009, 153–156. https://doi.org/10.1109/ICAPR.2009.17.
229. Vikas O, Chiluka NJ, Ray PK, Meena G, Meshram AK, Gupta A, et al., WebMiner—anatomy of super peer based incremental topic-specific Web crawler. In: *Sixth Int. Conf. Netw.*, IEEE, 2007, 32–32. https://doi.org/10.1109/ICN.2007.104.
230. Wei-jiang L, Hua-suo R, Kun H, Jia L. A new algorithm of blog-oriented crawler. In: *2009 Int. Forum Comput. Sci. Appl.*, IEEE, 2009, 428–431. https://doi.org/10.1109/IFCSTA.2009.110.
231. Wei-jiang L, Hua-suo R, Tie-jun Z, Wen-mao Z. A new algorithm of topical crawler. In: *2009 Second Int. Work. Comput. Sci. Eng.*, IEEE, 2009, 443–446. https://doi.org/10.1109/WCSE.2009.706.
232. Yang Y, Du Y, Sun J, Hai Y. A Topic-specific web crawler with concept similarity context graph based on FCA. In: Huang D-S, Wunsch DC, Levine DS, Jo K-H, eds. *Advanced Intelligent Computing Theories and Applications with Aspects of Contemporary Intelligent Computing Techniques*. Berlin and Heidelberg: Springer; 2008, 840–847. https://doi.org/10.1007/978-3-540-85984-0_101.
233. Yang Y, Du Y, Hai Y, Gao Z. A topic-specific Web crawler with web page hierarchy based on HTML dom-tree. In: *2009 Asia-Pacific Conf. Inf. Process.*, IEEE, 2009, 420–423. https://doi.org/10.1109/APCIP.2009.110.
234. Zhang H, Lu J. SCTWC: an online semi-supervised clustering approach to topical Web crawlers. *Appl Soft Comput* 2010, 10:490–495. https://doi.org/10.1016/j.asoc.2009.08.017.
235. Zhang W, Xu B, Lu H. Web page's blocks based topical crawler. *Proceedings of the 4th IEEE International Symposium on Service-Oriented System Engineering. SOSE 2008*, 2008, 44–49. https://doi.org/10.1109/SOSE.2008.10.
236. Zhang Y-H, Zhang F. Research on new algorithm of topic-oriented crawler and duplicated web pages detection. In: *8th International Conference of the Intell. Comput. Theor. Appl.*, 2012, 35–42. https://doi.org/10.1007/978-3-642-31576-3_5.
237. Zhao M, Zhu P, He T. An intelligent topic Web crawler based on DTB. In: *2010 International Conference of the Web Inf. Syst. Min.*, IEEE, 2010, 84–86. https://doi.org/10.1109/WISM.2010.155.
238. Zong X, Shen Y, Liao X. Improvement of HITS for topic-specific Web crawler. In: *Advances in Intelligent Computing*. Huang D-S, Zhang X-P, Huang G-B, eds. Berlin Heidelberg: Springer; 2005, 524–532. https://doi.org/10.1007/11538059_55.
239. Bergmark D. Collection synthesis. In: *Proceedings of the Second ACM/IEEE-CS Jt. Conf. Digit. Libr.—JCDL '02*, New York, NY, ACM Press, 2002, 253. https://doi.org/10.1145/544220.544275.
240. Chen L, Li Z, Yu Z, Han G. Classifier-guided topical crawler: a novel method of automatically labeling the positive URLs. In: *2009 Fifth International Conference of the Semant. Knowl. Grid*, IEEE, 2009, 270–273. https://doi.org/10.1109/SKG.2009.60.
241. Xu Y, Ai-na S, Zhan-kun T. Topical crawler based on multi-level vector space model and optimized hyperlink chosen strategy. In: *9th IEEE Int. Conf. Cogn. Informatics*, IEEE, 2010, 430–435. https://doi.org/10.1109/COGINF.2010.5599702.
242. Dixit A, Sharma AK. Security system for migrating crawlers. In: *2011 International Conference of the Comput. Intell. Commun. Networks*, IEEE, 2011, 667–671. https://doi.org/10.1109/CICN.2011.145.
243. Gupta A, Dixit A, Sharma AK. Prospective terms based architecture for migrating crawler. In: *2012 Fourth International Conference of the Comput.*

Intell. Commun. Networks, IEEE, 2012, 915–919. https://doi.org/10.1109/CICN.2012.168.

244. Kausar MA, Nasar M, Singh SK. Maintaining the repository of search engine freshness using mobile crawler. In: *2013 Annu. International Conference of the Emerg. Res. Areas 2013 Int. Conf. Microelectron. Commun. Renew. Energy*, IEEE, 2013, 1–6. https://doi.org/10.1109/AICERA-ICMiCR.2013.6575995.
245. Miller RC, Bharat K. SPHINX: a framework for creating personal, site-specific Web crawlers. *Comput Networks ISDN Syst* 1998, 30:119–130. https://doi.org/10.1016/S0169-7552(98)00064-6.
246. Pandey S, Mishra RB. Intelligent Web mining model to enhance knowledge discovery on the Web. In: *2006 Seventh International Conference of the Parallel Distrib. Comput. Appl. Technol.*, IEEE, 2006, 339–343. https://doi.org/10.1109/PDCAT.2006.74.
247. Upadhyay V, Balwan J, Shankar G, Amritpal A. Security approach for mobile agent based crawler. In: *Advances in Computer Science, Engineering & Applications*. Wyld DC, Zizka J, Nagamalai D, eds. Berlin and Heidelberg: Springer; 2012, 119–123. https://doi.org/10.1007/978-3-642-30111-7_12.
248. Wang Y, Du Y, Chen S. The understanding between two agent crawlers based on domain ontology. In: *2009 Int. Conf. Comput. Intell. Nat. Comput.*, IEEE, 2009, 47–50. https://doi.org/10.1109/CINC.2009.204.
249. Singhal N, Agarwal RP, Dixit A, Sharma AK. Information retrieval from the Web and application of migrating crawler. In: *2011 International Conference of the Comput. Intell. Commun. Networks*, IEEE, 2011, 476–480. https://doi.org/10.1109/CICN.2011.99.
250. Bal S, Nath R. A novel approach to filter non-modified pages at remote site without downloading during crawling. In: 2009 *Int. Conf. Adv. Recent Technol. Commun. Comput.*, IEEE, 2009, 165–168. https://doi.org/10.1109/ARTCom.2009.11.
251. Nath R, Bal S. A novel mobile crawler system based on filtering off non-modified pages for reducing load on the network. *Int Arab J Inf Technol* 2011, 8:272–279.
252. Pahal N. Security on mobile agent based crawler (SMABC). *Int J Comput Appl* 2010, 1:5–11.
253. Gao Q, Xiao B, Lin Z, Chen X, Zhou B. A high-precision forum crawler based on vertical crawling. In: 2009 *IEEE Int. Conf. Netw. Infrastruct. Digit. Content*, IEEE, 2009, 362–367. https://doi.org/10.1109/ICNIDC.2009.5360990.
254. Sachan A, Lim W-Y, Thing VLL. A generalized links and text properties based forum crawler. In: *Proceedings of 2012 IEEE/WIC/ACM International Conference on Web Intelligent Agent Technology* Washington, DC, IEEE Computer Society, 2012, 01, 113–120. http://dl.acm.org/citation.cfm?id=2457524.2457671
255. Heydon A, Najork M. Mercator: a scalable, extensible Web crawler. *World Wide Web* 1999, 2:219–229. https://doi.org/10.1023/A:1019213109274.
256. Chen R, Desai BC, Zhou C. CINDI robot: an intelligent Web crawler based on multi-level inspection. *Proc. Int. Database Eng. Appl. Symp. IDEAS*, 2007, 93–101. https://doi.org/10.1109/IDEAS.2007.4318093.
257. Cleverdon CW, Keen M. Aslib Cranfield research project-factors determining the performance of indexing systems. Volume 2, Test results., Technical Report, 1966.
258. Rijsbergen V, Joost C. Foundation of evaluation. *Journal of Documentation* 1974, 30:365–373.
259. Kausar A. Web crawler: a review. *Int J Comput Appl* 2013, 63:31–36.

APPENDIX A: OTHER PERFORMANCE METRICS FOR WEB CRAWLERS

Metric	Citation	Metric	Citation
Relevant webpage/all webpages	55, 82, 93, 96, 128, 157, 160, 232, 237	Geo-focus, geo-coverage, geo-connectivity	58
Fallout rate	46, 47, 49, 50	General and strict miss rate	138
Crawling time	47, 48, 69	Matching ratio	74
Efficiency	77, 221, 236	Mean average precision	46, 49
Maximum average similarity	65, 86, 87	Mean similarity	94
URL overlap	24, 58, 163	Performance/cost	125
Average precision	49, 50	Quality	163
Bandwidth saving	172, 251	Retrieval rate	36
Communication overhead	163, 250	Robustness	6

(*continued overleaf*)

APPENDIX A | Continued

Metric	Citation	Metric	Citation
Effectiveness	76, 206	S-Throughput	30
Freshness	186, 205	Satisfaction	77
Number of on topic webpages	129, 172	Scalability	187
Average target recall	123	Specificity	179
Consistency	221	Submission efficiency	207
Content similarity	72	Sum of information	39
Crawling path	150	Target length	106
Discard ratio	80	Total coverage rate and incremental coverage rate	206
Error rate	105	Total similarity	128
Exclusive harvest ratio	24	URLs to be seen/processed URLs	63
Focus	33	Webpages/time	19

APPENDIX B: ACRONYMS

ACHE	Adaptive Crawler for Hidden Web Entries
ACM	Association for Computing Machinery
ANN	Artificial Neural Network
AutoCrawler	Automatic Topical crawler
BINGO	Bookmark INduced Gathering of infOrmation
BioCrawler	BioTope's intelligent based crawler
BLFC	BaseLine Focused Crawler
BOB	Bayesian Object Based
BOB	Bayesian Object Based crawler
CBP-SLC	Content Block Partition- Selective Link Context
CCG	Concept Context Graph
CEI	Crawl-Extraction-Ingestion (CEI)
ccTLD	Classification Country Code-Top Level Domain
CINDI	Concordia Indexing and Discovering system
CRD	Centre for Reviews and Dissemination
DFT	Domain-specific Faceted Extractor
DMOZ	Directory Mozilla
DOM	Document Object Model
DWCIS	Distributed Web Crawling and Indexing System
EFFC	Enhanced Form Focused Crawler
FCA	Formal Concept Analysis
FCHC	Focused Crawling using Human Cognition
FCHC	Focused Crawling using Human Cognition
FFC	Form Focused Crawler
FoCUS	Forum Crawler Under Supervision
GA	Genetic Algorithms
GAV	Global As View

(*continued overleaf*)

APPENDIX B | Continued

GBS	Graph-Based Sentiment
GBS	Graph Based Sentiment crawler
Gcrawler	Genetic crawler
HiWe	Hidden Web expose Crawler
HMM	Hidden Markov Model
HTML	Hyper Text Mark-up Language
IEEE	Institute of Electrical and Electronics Engineers
IQIBO	Integrating Query Interfaces Based on Ontology
IR	Information Retrieval
Ispider	Intelligent Spider
JTCL	Java Tool Command Language
KCS	Kosmix Categorization Service
LAV	Local As View
LEHW	Label Extraction for Hidden Web crawler
LSI	Latent Semantic Index
LVS	Labeled Value set
MA	Master Agent
NSE	Nano Science & Engineering
OAI	Open Archives Initiative
OCCS	Open Corpus Content Service
ODP	Open Directory Project
OER	Open Educational Resources
ontoCrawler	Ontology crawler
OPAL	Ontology-based Web pattern Analysis with Logic
PruSM	Prudent Schema Matching
PVA	Programmer Viewpoint Attributes

(*continued overleaf*)

APPENDIX B | Continued

QBLP	Q-learning-Based Link Prediction
QProber	Query Prober
QTC	Query Triggered Crawling
RDF	Resource Description Framework
RSS	Rich Site Summary
RPHCI	Relevance Prediction based on Hierarchal Context Information
SA	Slave Agent
SASF	Self-Adaptive Semantic Focused crawler
SASF	Self-Adaptive Semantic Focused crawler
SCD	Service Class Description
SCS	Specialized Crawler for Security
SCS	Specialized Web Crawler for Security
SCTWC	Semi supervised Clustered Topical Web Crawler
SFC	Semantic Focused Crawler
SOF	Semi supervised Ontology-based Focused crawler

(*continued overleaf*)

APPENDIX B | Continued

SPHINX	Site-oriented Processor for Html INformation eXtraction
SVM	Support Vector Machines
TF-IC	Term Frequency-Information Content
Tf-idf	Term frequency-inverse document frequency
TSVS	Time Sensitive Vertical Search
UBFC	URL Rule-Based Focused Crawler
UIUC	University of Illinois Urbana Champaign
UQI	Universal Query Interface
URL	Uniform Resource Locator
UVA	User Viewpoint Attributes
VLDB	Very Large DataBases
VSM	Vector Space Model
WTMS	Web Topic Management System
XLST	Extensible Stylesheet language