Focused Crawling Based Upon Tf-Idf Semantics and Hub Score Learning

Mukesh Kumar, Renu Vig
Computer Science and Engineering Department, University Institute of Engineering and Technology
Panjab University
Chandigarh (INDIA)
Email: mukesh_rai9@yahoo.com, renuvig@hotmail.com

***Abstract*—A focused crawler traverses the Web to collect documents related to a particular topic, and can be used to build topic specific collection of documents for use in digital libraries and domain specific search. General crawlers make use of breath first search method to traverse the Web for as much amount of information as possible. Focused crawler help the search indexer to index all documents present on the World Wide Web related to a specific domain which in turn provides search engine's users complete and fresher most information. In this paper we present a focused crawler capable of learning from the previous crawl results to collect the documents related to the sports domain. Crawling results for four consecutive crawls are shown. Results shows significant improvement in the precision value for the crawler with respect to the number of crawling attempts made.**

***Index terms*—Web, Internet, Retrieval, Focused Web Crawler, Search Engine.**

I. Introduction

With the ongoing growth of web, finding the right information becomes an increasingly difficult task which often leads to undesired results. This made it important to develop document discovery mechanism. A crawler is a program used by search engine that retrieves Web pages by wandering around the Internet following one link to another. Web search engines such as Google, AtlaVista provides s. "Due to the Web's immense size and dynamic nature no crawler is able to cover the entire Web and to keep up with all the changes. This fact has motivated the development of focused (topical) crawlers. The focused crawlers are designed to download Web documents that are relevant to a predefined domain (e.g. genomics or immunology), and to avoid irrelevant areas of the Web", [32]. Focused crawler results in huge savings in network and computation resources by ignoring non relevant portion of the World Wide Web.

The exponential growth of the World Wide Web enforced the universal search engines to address the scalability limitations with huge amounts of hardware and network resources and by distributing the crawling process across users, queries, or even client computers. It makes difficult to discover topic relevant information that can be used in specialized portals and on-line search. To tackle this issue the focused web crawlers are emerging. Focused crawlers dynamically browse the Web by choosing the most promising links in order to try to maximize the relevancy of the retrieved pages and thus saving significantly time and computational resources.

II. Related Work

A general-purpose web crawler's basic task is to fetch a page, parse the links and repeat. It normally tries to gather as many pages as it can to build up a web-graph as complete as possible. Search engines use these web-graphs to identify the most authoritative pages related to a user query or topic. That process is called topic distillation. A brief overview of different distillation techniques is given below:

The first technique uses the textual similarity between pages. Similarity has been well studied in the Information Retrieval (IR) community and has been applied to the WWW environment [1]. Based on these statistics, the relevance of a page to a certain query is computed.

If one wanders on the Web for an indefinite time, following a random link out of each page, then different pages will be visited at different rates; popular pages with many in-links will tend to be visited more often. In other words, the importance of a page P is the number of links to P that appear on the entire Web. Intuitively, a page P that is linked to by many pages is more important than one that is seldom referenced. This" citation" count metric has been used extensively to evaluate the impact of published papers.

While the back link metric treats every link the same, page rank recursively defines the importance of a page P to be the weighted sum of the back links to P. This is the core of the PageRank algorithm, invented by Brin and Page [2]. The Pagerank algorithm crawl the web and simulates such a random walk on the Web graph in order to estimate the visitation rate, which is used as a score of popularity. Given a key word query, matching documents are ordered by this score. Note that the Pagerank popularity score is pre computed independently from the query; hence Google can potentially be as fast as any search engine that purely ranks the query results based on the input query.

doi:10.4304/jetwi.5.1.70-77

Hyperlink Induced Topic Search [3] is slightly different: it does not crawl or preprocess the Web, but depends on a search engine. A query to HITS is forwarded to a search engine such as AltaVista, which retrieves a sub graph of the Web of which the nodes (pages) match the query. Pages citing or cited by these pages are also included. This expanded graph is analyzed for popular nodes using a procedure similar to the Google, the difference being that not one, but two scores emerge: the measure of a page being an authority, and the measure of a page being a hub (a compilation of links to authorities). A variant of this technique, the Companion algorithm, has been used by Dean and Henzinger [4] to find similar pages on the Web using link-based analysis alone. Bharat, Henzinger, Kumar and Subramanian [5] improved the speed by fetching the Web graph from a connectivity server which has substantial pre-crawled portions of the Web.

Maintaining currency of search engine indices by exhaustive crawling is rapidly becoming impossible due to the enormous growth and dynamic content of the web. One of the ideas proposed in recent years is focused crawling. A focused crawler is designed to only gather pages relevant to a certain, pre-defined set of topics, without having to explore all Web pages. By definition, focused crawlers are preferred crawlers, that is, they must use some sort of heuristic to rate pages according to their relevance to the given topic. Focused crawlers have the advantage of being driven by a rich context (topics, queries, user profiles) within which to interpret pages and select the links to visit.

It is obvious that the success of the crawler depends on the quality of the heuristic used. During the crawl, the crawler should stay focused around the given topic, that is, it should give correct scores to the pages so that links on irrelevant pages are not pursued by the crawler. On the contrary, the heuristic must not be to”strict” so that relevant pages are still not found by the crawler.

One of the first web crawlers was given by Kornatzky, De Bra and Houben [6]. They made use of client-based real-time retrieval system for hypertext documents, based on depth-first search. The”school-of-fish” metaphor is used: when food (relevant information) is found, fish (search agents) reproduce and continue looking for food, in the absence of food (no relevant information) or when the water is polluted (poor bandwidth), they die. In other words, the”fish” follow more links from relevant pages, based on keyword and regular expression matching. The authors acknowledge that this type of system can make heavy demands on the network, and propose various caching strategies to deal with this.

Hersovici, Jacovi, Maarek, Shtalhaim, and Sigalit [7] give Shark-Search algorithm, a version of the fish-search algorithm. Shark-Search algorithm overcomes some limitations of the fish search by analyzing the relevance of documents more precisely and, more importantly, making a finer estimate of the relevance of neighboring pages before they are actually fetched and analyzed.

Another early experience with a focused crawler based on a hypertext classifier, is described in [8]. It describes a prototype implementation that is comprised of three programs integrated via a relational database: a crawler, a hypertext classifier and a distiller. The basic idea is to classify crawled pages with categories in topic taxonomy. The relevance rating system uses a hypertext classifier to update the metadata with topic information from a large taxonomy: a user marks interesting pages as they browse, which are then placed in a category in the taxonomy. This was bootstrapped by using the Yahoo hierarchy. Relevance is not the only attribute used to evaluate a page while crawling: the popularity rating system updates metadata fields signifying the value of a page as an access point for a large number of relevant pages. The latter is based on connectivity analysis. The two mining or rating modules guide the crawler away from unnecessary exploration and focus its efforts on web regions of interest. But the methodology becomes very complex and difficult to evaluate.

A web page categorization technique by context is given in [9]; it extracts useful information for classifying a document from the context where a URL referring to it appears.

Mukherjea proposed WTMS [10], a system for Web Topic Management. WTMS crawler collects Web pages for a topic and introduces the user interface of the system that integrates several techniques for analyzing the collection and also presented the various views of the interface that allow navigation through the information space. Aggarwal, Al-Garawi, Philip S.Yu gives a technique that crawls the pages which satisfy arbitrary user-defined predicates [11], such as topical queries, keywords queries or any combination of both.

A focused crawling algorithm that builds a model for the context with in which topically relevant pages occur on the web, is proposed in [12]. The context model can capture typical link hierarchies within which valuable pages occur, as well as model content on documents that frequently co-occur with relevant pages.

Random walk as an efficient and accurate approach to approximate certain aggregate queries about the web pages is given in [13]. It uses a random walk to produce an almost uniform distribution sample of web pages.

Davison [14] found that the likelihood of linked pages having similar textual content to be high, the similarity of sibling pages increases when the links from the parent are close together, titles, description, and anchor text may represent at least part of the target page.

Brian, Terveen, Will Hill [15] evaluated a number of link and context-based algorithms using a dataset of web documents rated for quality by human topic experts. Link-based metrics did a good job of picking out high-quality items.

Michael Steinbach George Karypis Vipin [16] Kumar studied different document clustering techniques. They compared the two main approaches to document clustering, agglomerative hierarchical clustering and K-means. (For K-means we used a “standard” K-means algorithm and a variant of K-means, “bisecting” K-

means.) Hierarchical clustering portrayed as the better quality clustering approach, but is limited because of its quadratic time complexity. In contrast, K-means and its variants have a time complexity which is linear in the number of documents, but are thought to produce inferior clusters. technique is better than the standard K-means approach and as good or better than the hierarchical approaches that we tested for a variety of cluster evaluation metrics.

Junghoo Cho, Hetor Gasrcia-Molina [17] gives a parallel crawler architecture that can enhance the crawling process by instaciating more than one crawlers in a centrally distributed environment.

Pant, Srinivasan, and Menczer [18] build a framework and a number of quality measures to evaluate topic driven crawling algorithms, and proved that a mix of exploration and exploitation is essential for

1. Seeking new relevant pages starting from a known relevant subset.
2. Seeking relevant pages starting a few links away from the relevant subset.

Marina Buzzi [19] gives a scheme to permit a crawler to acquire information about the global state of a website before the crawling process takes place. It require Web server co-operation in order to collect and publish information on its content.

Johnson and Giles in [20] proved experimentally that a rank function that combines analysis of text and link structure yields effective strategies for focused crawling that performed better than Best First strategy.

Marc Ehrig, Alexander Maedche [21] proposed an approach for document discovery building on a comprehensive framework for ontology-focused crawling of Web documents. The framework includes means for using a complex ontology and associated instance elements. It defines several relevance computation strategies and provides an empirical evaluation which has shown promising results.

Ismail Sengor Altingovde and Ozgur Ulusoy [22] presents a rule-based focused crawler that uses linkage statistics among topics to improve a baseline focused crawler's coverage.

Bong and Narayanan [23] proposed a local feature selection measure namely, Categorical Descriptor Term for text categorization. The method explicitly chooses feature set for each category by only selecting set of terms from relevant category.

P. Srinivasan, F. Menczer, and G. Pant [24] present a general framework to evaluate topical crawlers. They identified a class of measures for fair comparative evaluations of crawlers along some dimensions including generalized notions of precision, recall, and efficiency.

G. Pant, P. Shrinivasan [25] compared different classification schemes. They modeled the crawling process as a parallel best-first search over a graph defined by the Web. The classifiers provided heuristics to the crawler thus biasing it towards certain portions of the Web graph. Results showed that Naive Bayes is a weak choice for guiding a topical crawler when compared with Support Vector Machine or Neural Network. Further, the weak performance of Naive Bayes could be partly explained by extreme skewness of posterior probabilities generated by it. T was observed that despite similar performances, different topical crawlers cover subspaces on the Web with low overlap.

Jamali, Sayyadi, Hariri and Abolhassani [26] introduced a simple framework for focused crawling using combination of two existing methods, the Link Structure analysis and Content Similarity. M.Yuvrani, N.Ch.S.N.Iyengar, A.Kanan, presents a focused crawler framework [27] that makes use of link semantics to retrieve relevant documents and suggested that rule inference mechanisms can be used as a future work to enhance the crawling process. G. Pant, P. Shrinivasan[28] investigated the effects of various definitions of link contexts on the crawling performance. They concluded that a crawler that exploits words both in the immediate vicinity of a hyperlink as well as the entire parent page performs significantly better than a crawler that depends on just one of those cues. Also a crawler that uses the tag tree hierarchy within Web pages provides effective coverage.

Chang, C., Kayed, M., Girgis, MR. and Shaalan, KF [29] presented a survey of the major Web data extraction approaches and compares them in three dimensions: the task domain, the automation degree, and the techniques used. The criteria of the first dimension explained why an IE system fails to handle some Web sites of particular structures. The criteria of the second dimension classified IE systems based on the techniques used. The criteria of the third dimension measured the degree of automation for IE systems.

Antonio Badia, Tulay Muezzinoglu and Olfa Nasraoui [30] built a focused crawler as part of a larger project. The The National Surface Treatment Center (NSTCenter) web site was created with the goal to become a premier forum for Navy officers, independent consultants, researchers and companies offering products and/or services involved in the process of servicing Navy ships. In order to help generate content, they developed a focused web crawler that searched the web for information relevant to the NSTCenter. They developed a crawling system that achieves significant precision. Debajyoti Mukhopadhyay, Arup Biswas, Ukanta Sinha in [31] proposed a domain specific ontology based crawler that makes use of number of query words in the page to find similarity of a page.

Ari Pirkola in [32] described negligence of historical results and inability to handle intermediate linguity as the

main problems for any crawler. Xu, Qingyang and Zuo, Wanli [33] presented general framework of focused web crawling based on "relational subgroup discovery". Predicates were used explicitly to represent the relevance clues of those unvisited pages in the crawl frontier, and then first-order classification rules were induced using subgroup discovery technique. The learned relational rules with sufficient support and confidence were used to guide the crawling process afterwards.

McCown, F. and Nelson, M**.** [34] examined how search results decay over time and built predictive models based on the observed decay rates. Based on their findings, it could take over a year for half of the top 10 results to a popular query to be replaced in Google and Yahoo; for MSN it may take only 2-3 months.

J.Akilandeshwari, N.P.Gopalan presented a parallel Web spider model, based on multi-agent system for co-operative information gathering [35]. The system collects Web pages related to the topic, and then employs two agents: master agent and retrieval agents. Bao, S., Li, R., Yu, Y. and Cao, Y. [36] proposed CoMiner algorithm to conduct a Web-scale mining in a domain independent manner. The CoMiner algorithm consists of three parts: 1) given an input entity, extracting a set of comparative candidates and then ranking them according to comparability, 2) extracting the domains in which the given entity and its competitors play against each other, and 3) identifying and summarizing the competitive evidence that details the competitors' strength.

Ontology based approach for multilingual environment and meta search approach based crawling along with a measure for calculating page refreshment are given in [39, 40, 41, and 42].

III. PROPOSED WORK

"Tf-Idf ***(****Term frequency–Inverse document frequency)* weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus", [37], or to the domain. If we are having a corpus of documents which are all highly related with a specific domain then the Tf-Idf weight of a word in a page gives the importance of that term for that document with respect to the whole corpus. Now if we add Tf-Idf score obtained by a term for all documents in the corpus, then the resulting score can be seen as a meaningful, semantic, score for that term with respect to the whole corpus. Based upon this thought a TIDS **(**Term frequency–Inverse document frequency Definition Semantic) Score Table is constructed, whose entries are supposed to help the crawler for deciding the future crawls. The TIDS Score Table generation algorithm is given in Algorithm 1. The initial collection of Web pages related to the Sports domain (Seed pages) is generated from the hierarchical categories of Open Directory Project from http://dmoz.org. ODP provides the categorized bifurcated repositories of URLs which are manually edited. From here we can find individual categories link. These URLs are put in the Relevant_Page_Set.

Algorithm 1: TIDS Score Table Generation

1. Initialize Relevant_Page_Set.
2. Remove Stop Words from each page in the Relevant_Page_Set
3. Apply Stemmer to each page in the Relevant_Page_Set
4. Generate Tf-Idf Score Inverted Index Table for all the documents in the Relevant_Page_Set.
5. For each term t in the Tf-Idf Score Inverted Index Table Do
   - 5.1. Calculate sum of the Tf-Idf score obtained by t in all documents from Tf-Idf Score Inverted Index Table, let it be TIDS_Score.
   - 5.2. Insert entry <t, TIDS_Score> into TIDS Score Table.
   - 5.3. Normalize the TIDS_Score values in TIDS Score Table.

According to the TIDS Score Table Generation Algorithm stemming, which is the "process for reducing inflected (or sometimes derived) words to their stem, base or root form, generally a written word form" [38], and stop words removal is performed upon the Relevant_Page_Set. Tf-Idf score of the collection is calculated. The term frequency $tf_{t,d}$ of term *t* in document *d* is defined to be number of times that *t* occurs in *d,* df_t is the document frequency of *t,* means the number of documents that contain *t*. The df_t is an inverse measure of the informativeness of *t* also $df_t \le N$ where N is the total number of pages in the Relevant Page Set. Then the Idf (inverse document frequency) of *t* is given by

$$\mathrm{idf}_t = \log\ (N/\mathrm{df}_t) \qquad (1)$$

The Tf-Idf weight of a term t in the document d ($w_{t,d}$) is the product of its tf weight and its idf weight and will be given by

$$w_{t,d} = \log(1 + \mathrm{tf}_{t,d}) \times \log\ (N / \mathrm{df}_t) \qquad (2)$$

The TIDS_Score of a term t is given by $\mathrm{TIDS\_Score}(t) = \sum_{d \in Relevant\_Page\_Set} \mathrm{tf}.\mathrm{idf}_{t,d}$

(3)

Algorithm 2: First Crawl

1. Create TIDS Score Table using Algorithm 1, for all the pages present in Relevant_Page_Set.
2. Initialize SeedUrls by selecting 200 random links from Relevant_Page_Set.
3. While SeedURls is not empty
   - 3.1 URL=SeedUrls.Next();
   - 3.2 URL_Score= Similarity score of URL.discription terms from TIDS Score Table.
   - 3.3 Enqueue(CrawlQueue,URL, URL_Score);
4. While CrawlQueue is not empty

4.1 URL=Dequeue(URL_with_maximum_score, CrawlQueue);
4.2 Doc= Download(URL)
4.3 If Doc is not present in the Crawler Repository then add Doc to the Crawler Repository else GOTO 4.
4.4 Doc_Score= Similarity score of URL.text terms from TIDS Score Table.
4.5 If Doc_Score is greater than or equal to the text Similarity score of Relevant Page Set pages and the Doc is not present in the Relevant Page Set
4.5.1 Add Doc to Relevant Page Set and regenerate TIDS Score Table.
4.6 For all Link in Doc.links
4.6.1 Linkscore= Similarity score of Link.anchor terms from TIDS Score Table.
4.6.2 Score= Doc_Score + Linkscore;
4.6.3 If Score > Relevancy_Threshold
4.6.3.1 Enqueue(CrawlQueue, Link, Score);

According to the Algorithm 2, SeedUrls is initialized by 200 random links chosen from the Relevant_Page_Set. SeedUrls were inserted one by one in the crawler queue, which is a priority queue, as according to their similarity score from TIDS Score table. The crawler picks the URL with maximum score from the queue and downloads the corresponding document. The content similarity score of the page is calculated, and a value for each link present in the document is obtained by merging the parent's content similarity score with the link's own anchor text similarity score, and the link is inserted into the crawler queue. The complete process is repeated until the crawl queue is empty or the maximum crawled page limit is not reached. We executed the First Crawl for collecting 6000 pages, which will act as the relevant page set, R, for the future crawls as they came by crawling seed pages which were related to the focused domain. Hub URL is the one which is pointing to many other URLs and authority URL is the one which is pointed to by many URLs. Best hub is the one which is pointing to many relevant pages and the best authority is the one which is pointed to by many relevant pages. We used the hub score as a learning parameter for the crawler to select best seed pages for the next crawling phase. Let R be the set of pages which are related to the domain and the page P in R bears the interlinked behavior shown in Fig.:1.

Then the hub score for the page P in R is given by

$$HUB_p = \sum_{\forall Q \text{ in } R \,\exists\, Link(P \to Q) \text{ in } R} AUTHORITY_Q \quad (4)$$

And authority score of P is given by

$$AUTHORITY_p = \sum_{\forall Q \text{ in } R \,\exists\, Link(Q \to P) \text{ in } R} HUB_Q \quad (5)$$

After finding the hub and authority scores we normalize those using mean square root method.

Algorithm 3: Consecutive Crawl

1. Calculate Hub score and Authority score for all the pages present in the set of relevant pages, R, came as a result from the previous crawl.
2. Choose top 200 pages with highest Hub score from R, and initialize them to the SeedUrls.
3. While SeedUrls is not empty
   3.1 URL=SeedUrls.Next().
   3.2 URL_Score=Hub score of the URL
   3.3 Enqueue(CrawlQueue,URL,URL_Score).
4. While CrawlQueue is not Empty
   4.1 URL=Dequeue(URL with maximum URL_Score, CrawlQueue).
   4.2 Doc=Download (URL).
   4.3 Doc_Score=Similarity score of the Doc text and URL anchor text from the TIDS Score Table.
   4.4 For all links in Doc.Links
      4.4.1 LinkScore=Similarity Score of Link.Anchor terms from the TIDS Score Table.
      4.4.2 Score=Merge(Doc_Score,LinkScore)
      4.4.3 Enqueue(CrawlQueue, Link, Score)

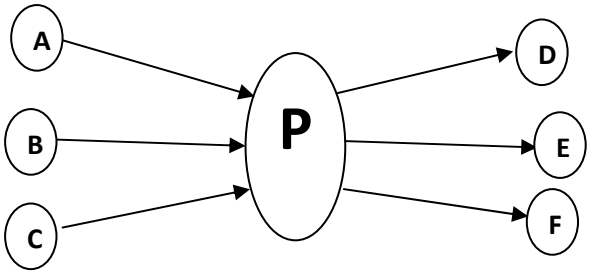
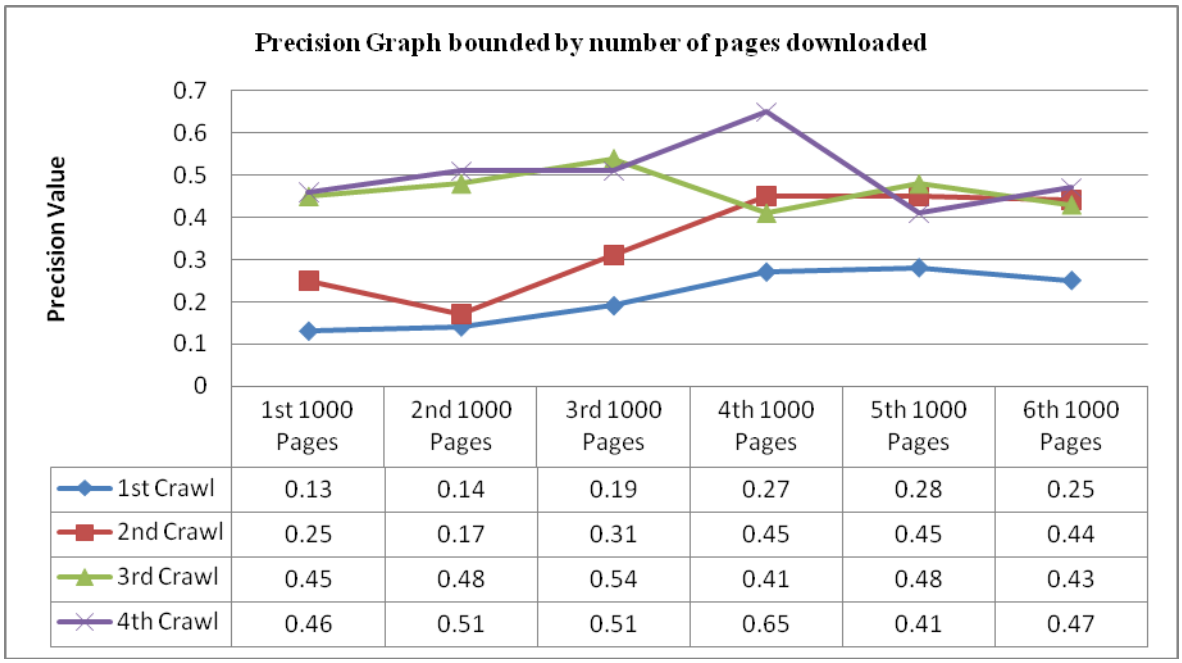
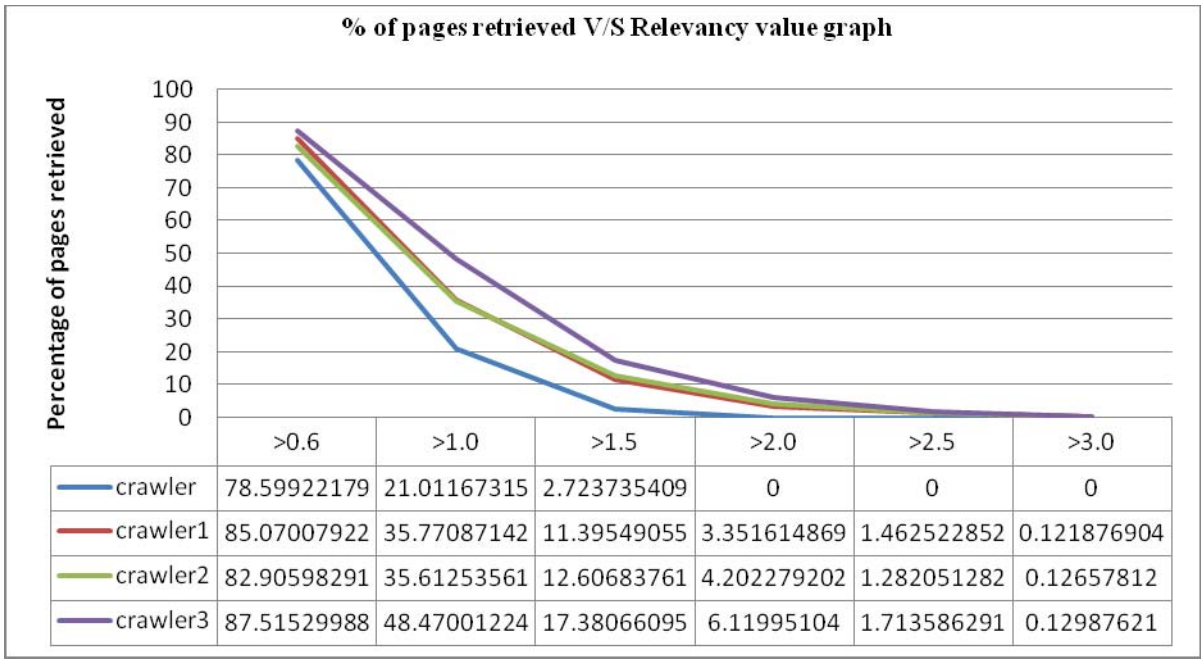


Fig 1: Interlinked behavior of Web page P in R where *{A,D,C,B,E,F,P} Є R*



	1st 1000 Pages	2nd 1000 Pages	3rd 1000 Pages	4th 1000 Pages	5th 1000 Pages	6th 1000 Pages
1st Crawl	0.13	0.14	0.19	0.27	0.28	0.25
2nd Crawl	0.25	0.17	0.31	0.45	0.45	0.44
3rd Crawl	0.45	0.48	0.54	0.41	0.48	0.43
4th Crawl	0.46	0.51	0.51	0.65	0.41	0.47

Fig. 2: Precision Graph showing the precision value (vertical axis) with respect to the number of pages downloaded by the various crawling phases (horizontal axis).



	>0.6	>1.0	>1.5	>2.0	>2.5	>3.0
crawler	78.59922179	21.01167315	2.723735409	0	0	0
crawler1	85.07007922	35.77087142	11.39549055	3.351614869	1.462522852	0.121876904
crawler2	82.90598291	35.61253561	12.60683761	4.202279202	1.282051282	0.12657812
crawler3	87.51529988	48.47001224	17.38066095	6.11995104	1.713586291	0.12987621

Fig. 3: Percentage of pages retrieved V/s relevancy value graph

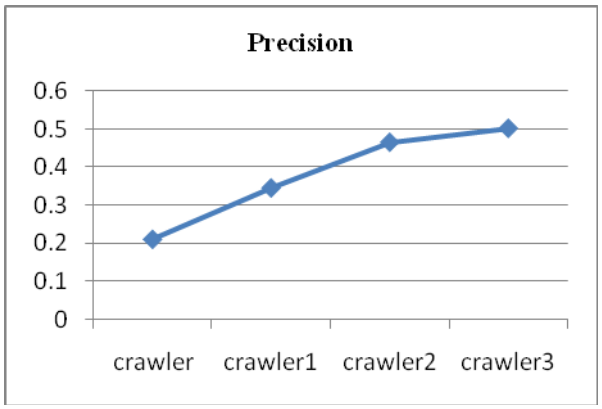


Figure 4: Consolidated Precision Graph

Consecutive Crawl algorithm works by finding the best hub and best authority pages among the pages which came as result of the previous crawl attempt, top 200 best hubs were chosen to act as the seed pages. All the seed pages are inserted one by one into the crawl queue, which is a priority queue, as according to their hub score. The URL with maximum score is chosen and the document corresponding to it is downloaded. The content similarity score of the page is calculated, and a value for each link present in the document is obtained by merging the parent's content similarity score with the link's own anchor text similarity score, and the link is inserted into the crawler queue. The complete process is repeated until the crawl queue is empty or the maximum crawled page limit is not reached.

IV. Experimental Results

The initial collection of Web pages (Seed pages) is generated from the hierarchical categories of ODP (Open Directory Project) from http://dmoz.org as suggested by Rungsawang, N.Angkawattanawit (2005)[2]. ODP provides the categorical collection of URLs that are manually edited and not biased by any commercial user. From here we can find individual categories link. The categories ending with "Sports", "cricket", "football", "base ball", "tennis", "badminton", "basket ball" etc. were retrieved from the ODP. The learning effect for four consecutive crawls is observed by finding the number of documents retrieved by the crawler having relevancy score greater than 1.0, all such pages are considered to be relevant to the sports domain. The results are plotted as graphs. Fig: 2 shows a graph between the precision value (vertical axis) and the number of pages downloaded by the various crawling phases (horizontal axis) bounded by the number of pages downloaded. Graph shows that the precision value tends to increase with the increasing number of the crawls for almost all set of the downloaded pages. Fig: 3 shows a graph between percentage of pages retrieved and relevancy value. It shows that the total number of pages retrieved by the crawler for different relevancy value increases with the increasing number of crawling phases. The consolidated precision graph is shown in Fig: 4, which shows that the precision value is showing an increasing trend with the increasing number of crawling phases.

V. Conclusion

Focused crawler based upon, Tf-Idf semantics and hub score based learning is proposed. Four consecutive runs of the proposed crawler were made to study the effect of learning. The results are plotted as graph between the precision value and the number of pages downloaded by the various crawling phases. Results show great improvement in average precision value with increasing number of crawls.

References

[1] Jerry H.Ying Dik L.Lee Budi Yuwono, SavioL. Lam, *A world wide web resource discovery system,* The Fourth International WWW Conference (Boston, USA), December 11-14, 1995.

[2] Sergey Brin and Lawrence Page,*The Anatomy of A Large-Scale Hypertextual Web Search Engine,* Computer Networks and ISDN Systems, 30:107-117, 1998.

[3] Jon M. Kleinberg, *Authoritative sources in a hyperlinked environment*, Journal of the ACM 46 no. 5, 604–632, 1999.

[4] Jeffrey Dean and Monika R. Henzinger, *Finding related pages in the World Wide Web*, Computer Networks 31 no. 11–16, 1467–1479, 1999.

[5] Krishna Bharat, Andrei Broder, Monika Henzinger, Puneet Kumar, and Suresh Venkata Subramanian, *The connectivity server: fast access to linkage information on the web,* Computer Networks. ISDN Systems 30,no. 1-7, 469–477, 1998.

[6] Y Kornatzky R Post P De Bra, G Houben, *Information retrieval in distributed hypertexts*, Proceedings of the 4th RIAO Conference(NewYork), pp.481–491, 1994.

[7] Michael Hersovici, Michal Jacovi, Yoelle S. Maarek, Dan Pelleg, Menanchem Shtalhaim, and Sigalit Ur, *The shark-search algorithm. an application: tailored web site mapping*, Computer Networks. ISDN Systems 30, no. 1-7, 317–326, 1998.

[8] S. Chakrabarti, M. van den Berg, B. Domc, *Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery*, Proceedings of the 8th international World Wild Web Conference, Toronto, Canada, 1999.

[9] Giuseppe Attardi, Antonio Gulli, Fabrizio Sebastiani, *Automatic Web Page Categorization by Link and Context Analysis,* in the proceedings of THAI-99, 1st Europpean Symposium on Telematatics, Hypermedia and Intelligence,1999.

[10] S. Mukherjea. *WTMS:System for Collecting and Analyzing Topic-Specific Web Information*, WWW 2000.

[11] Charu C. Aggarwal, Fatima Al-Garawi, Philip S.Yu, *Intelligent crawling through World Wide Web with Arbitrary Predicates*, ACM, WWW 10, Hong Kong., 2001.

[12] Michelangelo Diligenti, Frans Coetzee, Steve Lawrence, C. Lee Giles, Marco Gori. *Focused Crawling using Context Graphs*, In the Proceedings of 26th International Conference on Very Large Databases, VLDB 2000, Cairo, Egypt, pp. 527–534, 2000.

[13] Ziv Bar-Yossef, Alexander Berg Steve Chien, Jittat Fakcharoenpho, Dror Weitz *Approximating Aggregate Queries about Web Pages via Random Walks,* In the

Proceedings of the 26th VLDB Conference,Cairo, Egypt, 2000.

[14] Brian D. Davison, *Topical Locality in the Web*, in the Proceedings of the 23rd Annual International Conference on Research and Development in Information Retrieval , Athens, Greece, July 24-28,2000.

[15] Brian, Terveen, Will Hill*, Does "Authority" Mean Quality? Predicting Expert Quality Ratings of Web Documents,* In the Proceedings of 23rd Annual International ACM SIGIR Conference on Research and Development in Information retrieval, 2000.

[16] Michael Steinbach George Karypis Vipin Kumar, *A Comparison of Document Clustering Techniques*, 6th ACM SIGKDD, World Text Mining Conference, Boston, MA, 2000.

[17] Junghoo Cho, Hetor Gasrcia-Molina, *Parallel Crawlers*, WWW 2002.

[18] Gautam Pant, Padmini Srinivasan, Filippo Menczer *Exploration versus Exploitation in Topic Driven Crawlers.* In the Proceedings of 11th world wide Web Workshop on Dynamics, 2002.

[19] Marina Buzzi., *Cooperative Crawling*, Proceedings of the First Latin American Web Congress, 2003.

[20] Judy Johnson, Kotkas, C.Lee Giles, *Evolving Strategies for Focused Web Crawling,* In the Proceedings of 20th International Conference on Machine Learning, Washington, 2003.

[21] Marc Ehrig, Alexander Maedche, *Ontology-Focused Crawling of Web Documents,* Proceedings of Symposium on Applied Computing (SAC 2003), 2003.

[22] Ismail Sengor Altingovde and Ozgur Ulusoy, *Exploiting Interclass Rules for Focused Crawling*, published in Journal of IEEE Intelligent Systems, pp 66-73, November/December 2004.

[23] Bong Chih How, Narayanan K. *An Empirical Study of Feature Selection for Text Categorization based on Term Weightage, In the* Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, 2004.

[24] P. Srinivasan, F. Menczer, and G. Pant, *A general evaluation framework for topical crawlers*, Springer Science, Information Retrieval, 8, 417–447, 2005.

[25] G. Pant, P. Shrinivasan, *Learning to Crawl: Comparing Classification Schemes*, ACM Transactions on Information Systems (TOIS). Vol 23(4), 430-462. 2005.

[26] Mohsen Jamali, Hassan Sayyadi, Babak Bagheri Hariri and Hassan Abolhassani. A *Method for Focused Crawling Using Combination of Link Structure and Content Similarity*, Proceedings of the 2006 IEEWICIACM International Conference on Web,2006.

[27] M.Yuvrani, N.Ch.S.N.Iyengar, A.Kanan, *LSCrawler: a Framework for an Enhanced Focused Web Crawler based on Link Structures*, in the proceedings of the IEEE/ACM International Conference on Web Intelligence, 2006.

[28] G. Pant, P. Shrinivasan , *Link Contexts in Classifier-Guided Topical Crawlers*, IEEE Transactions on Knowledge and Data Engineering. Vol 18(01), 107-122. 2006.

[29] Chang, C., Kayed, M., Girgis, MR. and Shaalan, KF. , *A Survey of Web Information Extraction Systems*, IEEE Transactions on Knowledge and Data Engineering, TKDE-0475-1104.R3, 2006.

[30] Antonio Badia, Tulay Muezzinoglu and Olfa Nasraoui**,** *Focused Crawling: Experiences in a Real World Project,* Proceedings of 15th International Conference on WWW, Edinburgh, Scotland, 2006.

[31] Debajyoti Mukhopadhyay, Arup Biswas, Ukanta Sinha, *A New Approach to Design Domain Specific Ontology Based Web Crawler*, 10th International Conference on Information Technology, IEEE Computer Science, 289-291, 2007.

[32] Ari Pirkola, 2007, *Focused Crawling: A Means To Acquire Biological Data from the Web*, VLDB, September 23-28, Vienna, Austria, ACM, 2007.

[33] Xu, Qingyang and Zuo, Wanli**,** *First-order Focused Crawling*, Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada, Pages: 1159 – 1160, *2007.*

[34] McCown, F. and Nelson, M**.,** *Agreeing to Disagree: Search Engines and their Public Interface,* ACM IEEE Joint Conference on Digital Libraries (JCDL 2007). Vancouver, British Columbia, Canada. 309-318. June 17-23, 2007.

[35] J.Akilandeshwari, N.P.Gopalan, *Design of an enhanced Rule based Focused crawler*, IEEE International Conference on Emerging Trends in engineering and Technology, Page No. 798-801, 2008.

[36] Bao, S., Li, R., Yu, Y. and Cao, Y*., Competitor Mining with the Web Knowledge,* IEEE Transactions on Data Engineering, Volume: 20, Issue: 10, page(s): 1297-1310, Oct. 2008.

[37] http://en.wikipedia.org/wiki/TF-IDF (visited on 25-02-2012).

[38] http://en.wikipedia.org/wiki/Stemmig (visited on 28-02-2012).

[39] Mukesh Kumar, Renu Vig.:Design of CORE: context ontology rule enhanced focused web crawler. Published by ACM, New York, NY, Proceedings of the International Conference on Advances in Computing, Communication and Control, ISBN: 978-1-60558-351-8, Pages 494-497, doi>10.1145/1523103.1523201, (2009)

[40] Mukesh Kumar, Renu Vig.: Term-Frequency Inverse-Document Frequency Definition Semantic (TIDS) Based Focused Web Crawler. In Global Trends in Information Systems and Software Applications, Communications in Computer and Information Science, Published by Springer Berlin Heidelberg, ISBN: 978-3-642-29216-3, Vol. 270, pp. 31-36, DOI: 10.1007/978-3-642-29216-3_5, (2012).

[41] Neelam Goyal , Mukesh Kumar, Renu Vig.: Consistency Enforcement Using Ontology on Web. Journal of Computers, Academy Publishers, ISSN 1796-203X, Vol. 5, No. 10, Pages 1520-1526, doi:10.4304/jcp.5.10.1520-1526, (2010).

[42] Jaskirat Singh, Mukesh Kumar.: A Meta Search Approach to Find Similarity between Web Pages Using Different Similarity Measures. Advances in Computing, Communication and Control, Communications in Computer and Information Science, published by Springer Berlin Heidelberg, ISBN:978-3-642-18440-6, Vol. 125,pages 150-160, DOI: 10.1007/978-3-642-18440-6_19, (2011).