An Approach for Identifying URLs Based on Division Score and Link Score in Focused Crawler

Debashis Hati
Assistant Professor, School of Computer Engineering
KIIT University
Bhubaneswar, India

Amritesh Kumar
Research Associate, School of Computer Engineering
KIIT University
Bhubaneswar, India

ABSTRACT
The rapid growth of the World Wide Web (WWW) poses unprecedented scaling challenges for general-purpose crawlers. Crawlers are software which can traverse the internet and retrieve web pages by hyperlinks. The focused crawler of a special-purpose search engine aims to selectively seek out pages that are relevant to a pre-defined set of topics, rather than to exploit all regions of the Web. Focused crawler is developed to collect relevant web pages of interested topics from the Internet. Maintaining currency of search engine indices by exhaustive crawling is rapidly becoming impossible due to the increasing size of the web. Focused crawlers aim to search only the subset of the web related to a specific topic, and offer a potential solution to the problem. In our proposed approach, we calculate the link score based on average relevancy score of parent pages (because we know that the parent page is always related to child page which means that for detailed information any author prefers the child page) and division score (means how many topic keywords belong to division in which particular link belongs). After finding out link score, we compare the link score with some threshold value. If link score is greater than or equal to threshold value, then it is relevant link. Otherwise, it is discarded. Focused crawler first fetches that link which has greater value compared to all link scores and threshold.

General Terms
Crawler, Focused Crawler

Keywords
crawler; focused crawler; division score; link score

1. INTRODUCTION
In an effort to keep up with the tremendous growth of the World Wide Web (WWW), many research projects are targeted on how to retrieve and organize information in a way, that will make it easier for the end users to find the information they want efficiently and accurately. A Web Crawler searches through all the Web Servers to find information about a particular topic. However, searching all the Web Servers and the pages, are not realistic given the growth of the Web and their refresh rates. Crawling the Web quickly and entirely is an expensive, unrealistic goal because of the required hardware and network resources. Focused Crawling is designed to traverse a subset of the Web to gather documents on a specific topic [4]. It also aims to identify the promising links that lead to target documents, and avoid off-topic searches. Focused Crawlers support decentralizing the crawling process, which is a more scalable approach. The existing focused crawlers predict the probability of a document's relevance to the search topic employing probabilistic models, rules. Due to the fast expansion of the Web and the inherently limited resources in a search engine, no single search engine is able to index more than one-third of the entire Web. In order to gain better coverage various approaches have been introduced, such as the development of meta-search engines, special-purpose search engines and so on. The special-purpose search engine is constructed and optimized in accordance with domain knowledge. The focused crawler of a special-purpose search engine aims to selectively seek out pages that are relevant to a predefined set of topics, rather than to exploit all regions of the Web [2, 3].

This focused crawling technique enables a search engine to operate efficiently within a topically limited document space. The basic procedure of running a focused crawler is as follows. The crawler starts with several seed pages, which are topic-relevant. Whenever it fetches a Web page, the unvisited URLs are extracted from that page and scored by their relevance to the topics. The crawler then picks up the URL with the highest score to crawl. One of the major problems of the focused crawler is how to assign a proper order to the unvisited pages that the crawler will visit later. Many measurements, such as by means of exploiting the structure of linkage information on the Web, have been proposed to predict the importance of the documents. Domain-specific knowledge is used to rank the importance of a web page and to guide the crawler's search through the hyperlinks [1].

The topic-specific search engine is constructed and optimized in accordance with domain knowledge. A topic specific search engine can provide the information with higher precision than a general or directory search engine does. Naturally specialized search engines and domain-specific web portals have seen an increasing popularity in recent years. These are also called vertical engines and vertical web portals respectively [5]. A crawler is an agent which can automatically search and download webpages. Focused (topical) crawlers are a group of distributed crawlers that specialize in certain specific topics. Each crawler will analyze its topical boundary when fetching webpages.

2. BACKGROUND WORK
The WWW, having over 350 million pages, continues to grow rapidly at a million pages per day. About 600 GB of text changes every month. Such growth and flux poses basic limits of scale for today's generic crawlers and search engines. The general-purpose search engines offer a high coverage of the possible information on the Web, but they often provide results with low

precision. A directory search engine, such as Yahoo, is another type of search engines. It can limit the scope of its search upon the relevant categories that are compiled manually. The directory search engines return the results with higher precision [7].

Many research efforts have been put in the area of the choosing strategy for focused crawlers. In earlier days, researchers considered the link analysis method for general search engines to score the importance factors of web pages (URLs), and to first retrieve the page with a higher importance score. For example, Jungoo Cho proposed a Page Rank method to rank web pages [8]. The web pages are retrieved according to their Page Rank [9] values. This method considers the importance of web pages and ignores the relationship between the web pages and the specific topics. As a result, the crawlers designed by using this kind of methods easily lose their directions to a specific topic and retrieve fewer topic-specific pages. These crawlers should not be called focused crawlers because the ratio of the number of topic-specific web pages to the total number of web pages retrieved may decrease to zero [8]. So, it is a key problem for focused crawlers to discover and predict the relationship between a retrieved webpage and a specific topic. Davison [10] uses the TF-IDF vector space model to calculate the comparability among the web pages in an Internet sub graph containing 100,000 pages collected by the DiscoWeb system. He proposed the concept 'topical locality' of web pages to mean that two pages linked through hyperlinks have higher comparability than any two random web pages. This discovery pointed out a direction for focused crawling. Many research activities are conducted by this discovery to predict the relatedness between web pages and some specific topics. That is, if a page is relevant to a specific topic, the pages hyperlinked by it would be more likely to be related to this topic. So, it is more probable to find other topic-specific pages under the help of links from webs relevant to the specific topic. These crawlers just utilize the relativity between father page and some topic to predict the relationship between son pages and that topic to supervise their crawling. Altingovde [11] and his partners call these crawlers BaseLine Focused Crawl (BLFC).

In order to find pages of a particular type or on a particular topic, focused crawlers, which were first introduced by Chakrabarti et al. [6], aim to identify links that are likely to lead to target documents, and avoid links to offtopic branches. However the concept of prioritizing unvisited URLs on the crawl frontier for specific searching goals is not new and Fish-Search by De Bra et al. and Shark-Search by Hersovici et al. were some of the earliest algorithms for crawling for pages with keywords specified in the query. In Fish-Search, the system is query driven. Starting from a set of seed pages, it considers only those pages that have content matching a given query (expressed as a keyword query or a regular expression) and their neighborhoods (pages pointed to by these matched pages). Shark-Search is a modification of Fish-search which differs in two ways: a child inherits a discounted value of the score of its parent, and this score is combined with a value based on the anchor text that occurs around the link in the Web page.

3. PROPOSED ARCHITECTURE

The proposed architecture is shown below in Fig. 1.

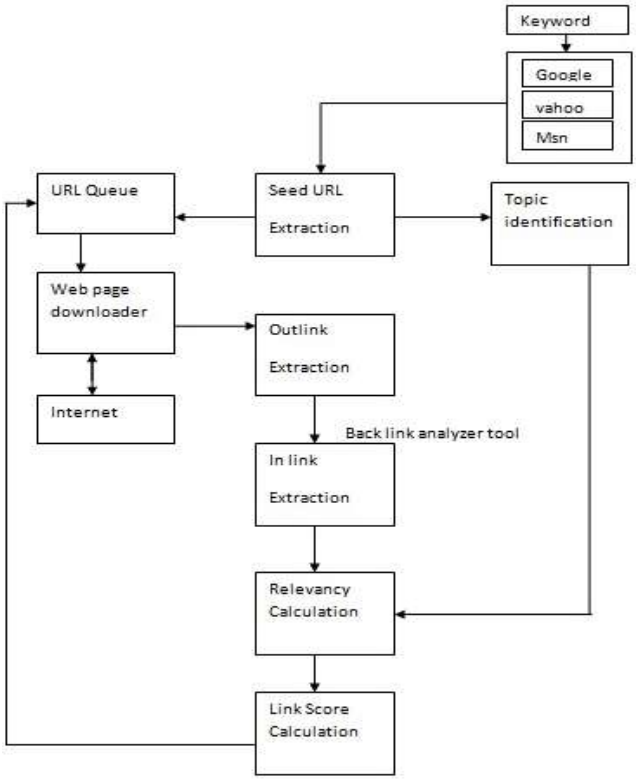

Figure 1. The Proposed Architecture.

4. PROPOSED APPROACH

4.1 Seed URL Extraction

In our proposed approach, seed URLs are extracted by one search engine known as www.threesearches.com. We put a query in this search engine and it shows the result of three most popular search engines like Google, Yahoo, and MSN search. We take resulting URLs which are common in all the three search engines or common in at least two search engines. A URL which is common in all three search engines like Google, Yahoo, and MSN search, we assume that this common search result URL is most relevant for this query and thus these URLs are the seed URLs, and a URL which is common in three search engines result by experiment, belongs to most relevant category seed URLs. These common URLs are referred by all three search engines for query. We also assume that a resulting URL which is common in two search engine results is not most relevant for topics but it is relevant for topics and we are putting it also in seed URLs category.

4.2 Key Generation

In our proposed approach, key generation process is based on seed URL. First we find out all seed URLs because we know that the seed page is most relevant page for topic keywords. Then, focused crawler fetches seed pages from internet. We extract top ten weighed common words from all seed pages. Weight of words is defined by the term frequency. Term frequency is calculated by number of occurrences of particular term divided by total number of terms in particular web page. After fetching top ten weighed common words from all seed pages, we

normalize the term weight. For example, we put the query "computer books." We find out different seed URLs. Here we find out two most relevant seed URLs.

www.freecomputerbooks.com

www.computer-books.us

From these two seed URLs, we extract top 10 weighted common terms after removing "stop" word. Then, we order the word by their weights and extract a certain number of words with high weights as the topic keywords. After that, the weights are normalized as:

$$W = Wi / Wmax \quad \text{(i)}$$

where Wi is the weight of keyword i and Wmax is weight of keyword with highest weight. Table 1 shows the sample weight table for topic "computer books."

Table 1. Sample weight table

Term	Weight
Book	1
Program	0.623376623
Free	0.533910533
Code	0.493506493
Source	0.44011544
Software	0.422799422
Process	0.409812409
Java	0.366522366
Linux	0.317460317
Basic	0.298781298

4.3 Link Score Calculation

After fetching seed pages from internet, our approach finds out all the out links of fetched web pages by java program and then we calculate the link score of each URL. In focused crawler the link score is calculated before fetching the web page because the main work of focused crawler is to determine which link is more topical relevant before the link will be fetched. In our proposed approach, focused crawler fetches link first which has more link score. Link score is calculated based on division score and average relevancy score of total parent page of particular link. In our proposed approach, division score of link has been taken for finding out relevancy of link because we know that the detailed description of links is available in division in which the links belong. One division in HTML web page is for which the URL will be fetched first. An unvisited URL which has more link score is fetched first by focused crawler from internet. For example, here we take one seed URL www.freecomputerbooks.com and total outlinks of this seed URL are 124.

In our proposed approach, average relevancy score of parent page is calculated because hyperlink is a reference of a child web page that is contained in a parent web page. When the hyperlink is clicked on in a parent web page, then the browser displays child web page. This functionality alone is not helpful for web information retrieval. However, the way hyperlinks are typically used by authors of web pages can give them valuable information content. Typically, authors create links because they think they will be useful for the readers of the pages. Thus, links are usually either navigational aids that, for example, bring the reader back to the homepage of the site, or links that point to pages whose content augments the content of the current page. The child links tend to point to high-quality pages that might be on the same topic as the page containing the link. Based on this motivation, link analysis makes the following simplifying assumptions:

- A link from parent page to child page is a recommendation of child page by the author of parent page.
- If parent page and child page are connected by a link the probability that they are on the same topic is higher than if they are not connected.
- URL score is calculated for identifying that which URL is most relevant for topics, so it will be fetched first.

Based on the above mentioned sentences, we have assumed that the contents of parent page of a link may be very useful for calculating the link score with respect to topical relevancy. We cannot directly say that a URL which belongs to a web page with more topical relevancy value has more relevant score compared to a URL which belong to a web page which is relevant but it has less relevant score compared to that web page. For finding out which URL is more relevant for topics, we calculate the score of link. For this, we first calculate the relevancy score of parent page's in which this link exist. Then we find out average relevancy score of parent page. In Fig. 2, we represent only some URLs out of total.

Figure 2. URLs.

We calculate the link score by this formula.

Link score(j) = div_score(j) * avg(total_relevancy _score_of_parent page_of(j))

where div_score is calculated based on topic keywords available in division i.e. we find out how many topic keywords are present in a division in which this particular URL exists. If all topic keywords are available in division in which this URL belongs, then division score of this URL is 1. Otherwise, it depends on percentage value of topic keyword's appearance in division. There is one division, under this number of sub divisions, and under sub division number of sub sub divisions. We find out division score based on sub division and "j' is the link which is to be fetched.

If five topic keywords are present in division in which particular URL belong, then division score of this URL is 0.5. Now based on our approach, we calculate the link score of some link.

For example, we take two outlinks out of total links from Fig. 1 and find out which link will be fetched by focused crawler first.

a. http://astore.amazon.com/compubookstut-20
b. http://freecomputerbooks.com/javaJ2eeBooks.html

First, we calculate the division score of both the links.

Division score of link(a) = 1

Division score of link(b) = 1

Both links belong to same division in HTML page of www.freecomputerbooks.com.

Maya Online Super Store

Post under Java EE (J2EE) and EJB (Enterprise Java Beans)

Second, we calculate the "total_relevancy_score_of_parent page_of(j)" which is based on parent page relevancy of this particular link. We extract all parent pages of particular link and calculate the relevancy score of this parent page with respect to topic keywords and sum to all relevancy scores of parent pages. Here, we take only five parent pages of each link.

a. http://astore.amazon.com/compubookstut-20

The five parent pages of this link are:

1. http://astore.amazon.com/numericalrecipes/detail/0139642625
2. http://www.dlrptoday.com/news/disney-village/earl-of-sandwich
3. http://www.wildwolfwomen.com/ss.htm
4. http://www.mybloglog.com/buzz/members/wingedpower
5. http://www.problogger.net/archives/2006/11/20/how-to-optimize-a-shoplink-store

Now, we calculate the relevancy score of each parent page with respect to topic keywords.

4.3 Relevancy Calculation

After fetching web pages, based on our approach, we remove the "stop" word, and words are stemmed using porter stemming algorithm. Then, we extract top 10 weighted words from fetched page. Weight is calculated by term frequency of each terms divided by total number of terms, and this term frequency is the weight of particular terms.

Term frequency = number of times term appear in web page/total number of terms in web pages.

Now, term weight is normalized by equation (i).

Relevancy score is calculated based on vector space model.

$$Relevance\ (t,\ p) = \frac{\sum k \varepsilon (t \cap p)\ wkt\ wkp}{\sqrt{\sum k \varepsilon t\ (wkt)^2 \quad \sum k \varepsilon t\ (wkp)^2}}$$

where R(t, p) is the relevancy between topic keywords and web page, wkt is the weight of topic keywords in topic table, and wkp is the weight of topic keywords in web page table.

For example, here Table 2 is representing page table of the link http://astore.amazon.com/numericalrecipes/detail/0139642625.

Table 2. Page table

Term	Weight
Book	0.524647887
Program	0.239436619
Free	0
Code	0
Source	0
Software	0
Process	0.239436619
Java	0
Linux	0
Basic	0

Based on equation Relevance (t, p), the relevancy score of this link with topic keywords are:

0.772031175/1.041511347 = 0.741260455.

We will calculate rest of the four links by using the above approach.

0.759192422

0.650288546

0.480327328

0.705386868

The average relevancy score of all these links is:

0.667291123

Now, the link score of the following link is:

http://astore.amazon.com/compubookstut-20

Link score(j) = div_score(j) * avg(total_relevancy _score_of_parent page_of(j))

= 1 * 0.667291123

= 0.667291123

Now, we calculate the total_relevancy_score_of_parent page_of link

http://freecomputerbooks.com/javaJ2eeBooks.html

The five parent pages of this link are:

1. http://freecomputerbooks.com/webAjaxBooks.html
2. http://freecomputerbooks.com/Programming-Scala.html
3. http://freecomputerbooks.com/the-microsoft-net-developer-ebook.html
4. http://freecomputerbooks.com/Web-Style-Guide.html
5. http://freecomputerbooks.com/langBasicBooks.html

The relevancy score of all these five links are:

0.83595444

0.886155978

0.865003484

0.879136756

0.898675612

The average relevancy score of all these links is:

0.872985254

The link score of the link http://freecomputerbooks.com/javaJ2eeBooks.html

is:

Link score(j) = div_score(j) * avg(total_relevancy _score_of_parent page_of(j))

= 1 * 0. 872985254

= 0.872985254

The link score in these two links are (a) 0.667291123, and (b) 0.872985254 respectively. So, obviously focused crawler selects first the link http://freecomputerbooks.com/javaJ2eeBooks.html and after that http://astore.amazon.com/compubookstut-20.

5. PROPOSED ALGORITHM

Step 1: Select seed pages from www.threesearches.com.

/*For seed pages, we put the query in threesearches.com. Then, we find out all common results in all three search engines result like Google, Yahoo and MSN. We put these all common URLs in most relevant category. A URL which is common in two search engine results, we put that URL in relevant category. In our example, the query is "computer books."*/

Step 2: Prepare topic keywords table.

/*From all most relevant seed pages, we extract top ten weighed common words. Weight is calculated by term frequency. Now, the weight is normalized by weight of single term divided by maximum weighted terms.*/

Step 3: Focused crawler fetches seed pages from internet.

Step 4: With the help of java program, we extract all the out links of fetched page.

/*Let there are n out links in web pages and we have to calculate the relevancy score of all out links.*/

Step 5: Calculate the link score of each out link.

/*To decide which out link is most relevant to our topic, we calculate the link score of each link. Link score is calculated based on division score of link and average relevancy score of parent pages of link. For average relevancy score of parent pages of link first our approach find out relevancy score of parent pages of link by vector space model. Division score is calculated based on number of keywords existing in a division in which a particular URL exists. Average relevancy score of pages is calculated in Step 6 to Step 9.*/

Step 6: For i = 1 to n

/* Here, n is the total out links of fetched web page.*/

Step 7: Fetch all in links of each link(i).

/*With the help of back link analyzer tool, we extract all in links of link(i). We assume that there are m in links of each link (i)*/

Step 8: For j = 1 to m

Step 9: calculate the relevancy of each link(j) by vector space model .

$$Relevance\ (t,\ p) = \frac{\sum k \varepsilon (t \cap p)\ wkt\ wkp}{\sqrt{\sum k \varepsilon t\ (wkt)^2 \quad \sum k \varepsilon t\ (wkp)^2}}$$

Step 10: Calculate average relevancy score from all parent page relevancy score.

/* In our proposed approach, it has been define that average relevancy score of parent page because average value defines that how much parent pages are relevant of this particular link with respect to topics.*/

Step 11: Link score(i) = division_score(i) + average_relevancy_score(j).

/* Here, j is the parent page of link i. Average relevancy score of Link j means first it find out relevancy score from 1 to j link and find out average relevancy score.*/

Step 12: Based on link score(i), focused crawler decides which link will be crawled first and which link will be crawled next. In our proposed approach, it uses some threshold value v in all link

score. If the link score value is less than threshold value, then it is rejected.

6. CONCLUSION AND FUTURE WORK

Focused crawler is an important member of search engine family. But one of the key problems of vertical search engines is to develop an effective algorithm for the topic-specific search and similarity measurement. Our approach is based on calculating the relevancy score of URL based on its division score with respect to topic keywords available in division i.e. we find out how many topic keywords are present in a division in which this particular URL exists and calculating the total relevancy score of parent page of URL which is based on parent page relevancy of this particular URL. Future work also includes code optimization and frontier optimization, because crawler efficiency not only depends on to retrieve maximum number of relevant pages but also to finish the operation as soon as possible. Several researchers are working on this area which is still in its infancy. In the near future, this proposed algorithm will be validated.

7. REFERENCES

[1] X. Zhang, T. Zhou, Z.Yu and D.Chen, "URL Rule Based Focused Crawlers", IEEE International Conference on e-Business Engineering, 2008.

[2] A. Pal, D. S. Tomar and S.C. Shrivastava. "Effective Focused Crawling Based on Content and Link Structure Analysis", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 2, No. 1, June 2009.

[3] Y. Zhang, C. Yin and F. Yuan. "An Application of Improved PageRank in Focused Crawler", Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007IEEE).

[4] Q. Cheng, W. Beizhan and W. Pianpian. "Efficient focused crawling strategy using combination of link structure and content similarity", IEEE 2008.

[5] X. Chain and X. Zhang. "HAWK: A Focused Crawler with Content and Link Analysis", IEEE International Conference on e-Business Engineering, 2008.

[6] S. Chakrabarti, M. van den Berg and B. Dom. "Focused crawling: a new approach to topic-specific Web resource discovery", 8th International WWW Conference, May 1999.

[7] M. Yuvarani, N. Ch. S. N. Iyengar and A. Kannan, "LSCrawler: A Framework for an Enhanced Focused Web Crawler based on Link Semantics" in Proceedings of the 2006 IEEE/WIC/ACM International Conference on WebIntelligence.

[8] Novak, B., "A survey of focused web crawling algorithms", in Proceedings of SIKDD 2004 at Multiconference IS. 2004, ACM Press: Slovenia. p. 55-58.

[9] Sergey, B., Lawrence, Page. "The anatomy of a largescale hypertextual Web search engine", Computer Networks and ISDN Systems 1998. 30(1-7): p. 107-117.

[10] Davison, B.D. "Topical locality in the Web", in Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval 2000: Athens, Greece. p. 272-279.

[11] Altingovde, I.S., Ulusoy, O. "Exploiting interclass rules for focused crawling", IEEE Intelligent Systems, 2004. 19(6): p. 66-73.