Multilingual Context Ontology Rule Enhanced Focused Web Crawler

Mukesh Kumar and Renu Vig
{mukesh_rai9@yahoo.com,renuvig@hotmail.com}
University Institute of Engineering and Technology, Panjab University, Chandigarh ,INDIA

***Abstract*— Rapidly growing size and increasing number of Non-English resources on World-Wide-Web poses unprecedented challenges for general purpose crawlers and Search Engines. It is impossible for any search engine to index the complete Web. Focused crawler cope with the growing size by selectively seeking out pages that are relevant to a predefined set of topics and avoiding irrelevant regions of the Web. Rather than collecting and indexing all accessible Web documents, focused crawler analyses its crawl boundary to find the links likely to be the most relevant for the crawl. This paper presents a focused crawler whose crawl strategy is based upon the scores calculated from context ontologies and adaptive classification rules, and which is capable to deal with intermediate multilinguity situations (the situations in which the query language is same as that of target language but the intermediate path may pass through some pages which are written in mixed, in query and some other language, way). It enhances the quality of pages retrieved, because it may be possible that the English meaning of the other language word sequence may itself or point to some pages which are most relevant to the query, and hence should be included in the results, which, yet, are left untouched by all the existing crawlers.**

***Index Terms*— Focused Crawler, Search Engines, Information Retrieval, Ontology, Adaptive Rules**

I. INTRODUCTION

The World Wide Web, having more than 350 million pages, continues to grow rapidly at a million pages per day [7]. About 600 MB of text changes every month. Such growth and flux poses basic limits of scale for today's generic crawlers [6] and search engines. It is not possible for any search engines to index the whole Web and to keep track upon the huge consistency management. The only way out of this problem is Focused Crawling [12]. A focused crawler tries to fetch only relevant region of the Web and avoiding irrelevant ones. Since initiation of World Wide Web most of the access is in English language which is the most dominating and preferred one. In recent times there is rapid growth in popularity of internet in semi -English speaking countries like India. Currently the 52% of the whole Web is English and rest is non-English and semi-English. A large fraction of this increasing semi-English population read and writes in mixed way, page written in English and other language (like Hindi).

In this paper focused crawler architecture is presented. The proposed crawler deals with problem of growing size of the Web by focusing its crawl on two parameters: relevancy score Figure: 1 show a Web page tree in which the black nodes are calculated from context ontologies, and adaptive classification rule score.

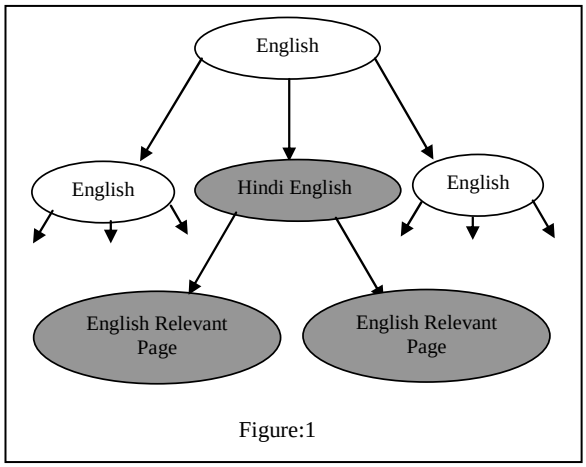

Figure:1

The nodes which are not to be traversed further by any of the existing crawler [2,3,4,5,6,8,9,10,11]. Though the page which is written in Hindi and English, do not contain any relevancy in terms of English text, yet there may be some text written in Hindi whose English transcription makes the page relevant to the user query and that further point to a relevant page written in English. This situation is termed as intermediate multilinguity.

The proposed crawler is able to deal with intermediate multilinguity situations, the situations in which the query language is same as that of target language but the intermediate path may pass through some pages which are written in mixed, in query and some other language, way by making the use of Bilingual dictionary approach.

II. RELATED WORK

The area of multilingual information retrieval has been well explored in the past few decades. The task was approached in two thoughts. One was a translation of the query followed by retrieval in monolingual domain, where as the second was translating the documents into query language and performing retrieval [4]. Broadly, it can be said that the task has been seen as a translation followed by retrieval approach. Bilingual dictionaries derived from Corpus are used for the translation.

Web crawling was simulated by a group of fish migrating on the Web [11]. In the so called fish search, each URL corresponds to a fish whose survivability is dependant on visited page relevance and remote server

Manuscript received May 8, 2009; revised August 11 2009.

doi:10.4304/jait.1.1.21-25

speed. Page relevance is estimated using a binary classification by using a simple keyword or regular expression match. Only when fish traverse a specified amount of irrelevant pages they die off. The fish consequently migrate in the general direction of relevant pages which are then presented as results

[5] Propose calculating the Page Rank [8] score on the graph induced by pages downloaded so far and then using this score as a priority of URLs extracted from a page. They show some improvement over the standard breadth-first algorithm. The improvement however is not large. This may be due to the fact that the Page Rank score is calculated on a very small, non-random subset of the web and also that the Page Rank algorithm is too general for use in topic-driven tasks.

[9] Considers an ontology-based algorithm for page relevance computation. After preprocessing, entities (words occurring in the ontology) are extracted from the page and counted. Relevance of the page with regard to user selected entities of interest is then computed by using several measures on ontology graph (e.g. direct match, taxonomic and more complex relationships).

A critical look at the available literature indicates that, the existing crawling approaches have following to be said:

1. None of them make use of efficient relevance score and tunneling (process of reaching to relevant pages from the irrelevant pages with in the current page) in combination to retrieve the Web documents.
.2. Lot of work has been done in general information retrieval, also in multilingual information retrieval, where user enters complete query in one language and results are in some other language, but the crawling is done through the single language only. No work has yet been done to tackle the intermediate multilinguity situations (Fig.1), which can considerably affect the harvest ratio of the crawler

III. PROPOSED CRAWLER

Fig.2 depicts architecture of the proposed focused Web crawler.It crawler works as per the following code segment:

1. Repeat until Maximum Crawler Limit is reached.
2. Take the user query and initialize the seeds along with their priority as according to the SeedInitializer algorithm discussed later. Set minimum ontology relevance Min_OntRel, and minimum look ahead relevance constant Min_LaRel to some constant values.
3. Download the seed pages as according to their priority and for each seed page go to Step 4.
4. For each link in page go to Step 5.
5. Generate the Context Link with the help of Context Link Extractor and then perform transcription for the intermediate multilinguity with the help of Multilingual Transcriptor and go to Step 6.
6. Calculate the OBRS and LARS (as discussed later) and put them in the priority queue as according to its OBRS.
7. Retrieve the link with highest OBRS. If its OBRS>Min_OntRel then download the Web page and go to Step 4 Else go to Step 8.
8. If it's LARS>Min_LaRel then download the Web page and go to Step 4.

Functioning of the main components is discussed below:

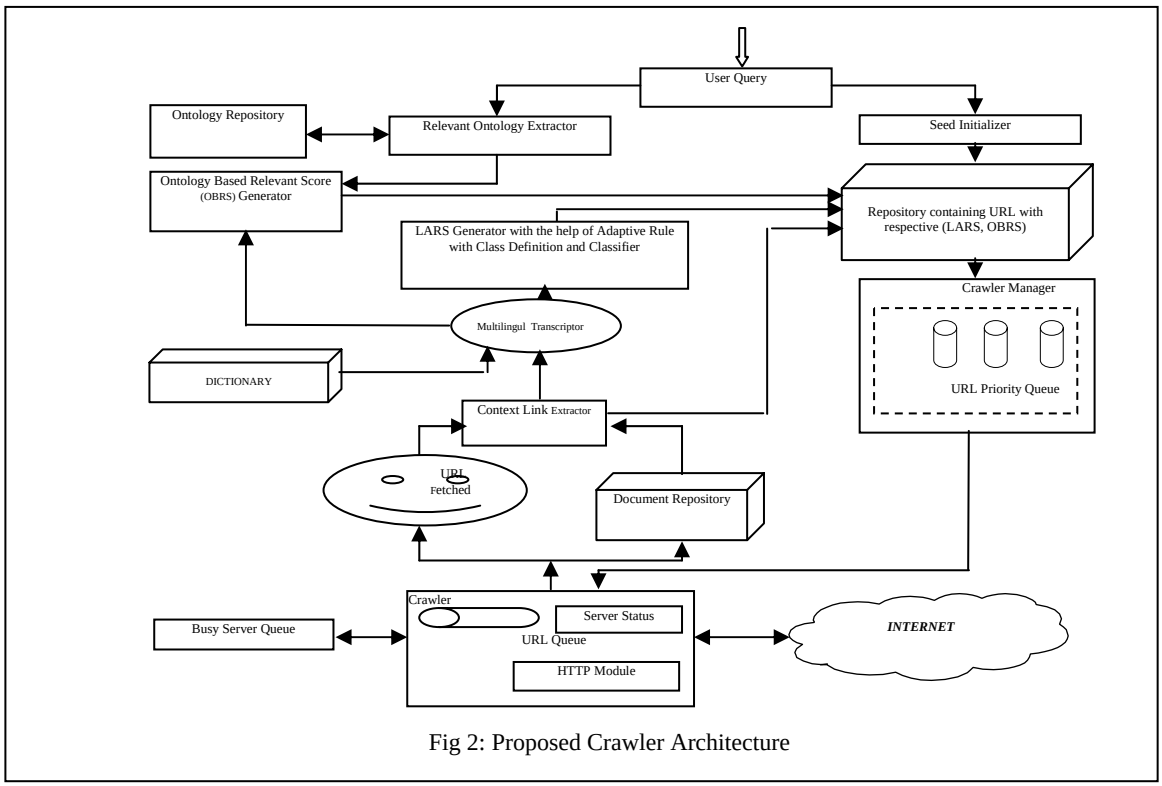

Fig 2: Proposed Crawler Architecture

A. SEED INITIALIZER

SeedInitializer algorithm work as per the seed detector discussed in [10]. It retrieves the seed URLs from the three most popular search engines Goggle, Yahoo and MSN for the specific keyword. It prioritizes the seeds in the following three classes:

High: URLs occurring across Search Engines more than once.

Medium: URLs repeatedly occurring within the Search Engine, not across the Search Engines.

Low: Other URLs occurring only once within the Search Engine.

B. CRAWLER MANAGER

Crawler manager fetches the URLs from the URL repository and add them to the priority queue. It generates the crawler instances that download the document.

C. CRAWLER

Crawler is a multi-threaded program [10] that is capable of downloading the Web pages from the Web and storing the documents in the document repository. Each crawler has its own queue, which holds the list of URLs to be crawled. The crawler fetches the URLs from the queue. The same or different crawlers would have sent a request to the same server. Busy Server Queue is maintained to have the list of URLs to which the crawlers have sent the request and awaiting for the response. Once the server is connected all the requests are fulfilled. Also instead of disconnecting, it keeps connected for a fixed time interval for the future requests.

D. CONTEXT LINK EXTRACTOR

Context Link Extractor [10] fetches the document from the Document Repository and extracts the URLs. It then checks for the URLs extracted in the URL Fetched. If not found, the surrounding text which include a fixed number of letters preceding and succeeding the hyperlink, the heading or sub heading under which the hyperlink appears is extracted from the document. The extracted link with the context information is passed to Multingual Transcriptor and URL Repository.

E. MULTILINGUAL TRANSCRIPTOR

This is the component that deals with the intermediate multilinguity situations. It makes use of a bilingual dictionary (e.g. Hindi to English) for transcription.It works as according to the following code segment:

1. If the Context Link is in English language then go to Step 3 , Else go to Step 2
2. Locate the Hindi context, remove the noise words and transcript for that in English by making use of the Bilingual Dictionary available, generate the transcripted Context Link, and go to Step 3.
3. Pass the Context Link to the OBRS Generator and LARS Generator.

F. OBRS GENERATOR

Ontology Based Crawling [2, 9] is one of the backbone features of our crawler. OBRS generator makes use of the relevant ontology extracted by the Relevant Ontology Extractor (ROE) and context link passed by the Context Link Extractor (CLE). It uses an Importance Table that importance of each term occurring in the relevant ontology passed from the ROE. A more relevant term to the query will have the more importance and the terms which are common to more than one domain have less importance. Importance Table for a given ontology is given in Table 1.

The following code segment is used for calculation of OBRS for a Context Hyper Link passed from the CLE with the help of Importance Table

Table 1: Importance Table

Ontology terms	*Importance*
Comp. Sc. And Engg.	1.0
Comp. Engg.	0.9
Comp. Sc.	0.8
Information Tech.	0.5
Computer	0.4
Engineering	0.3

OBRS Generator Algorithm

INPUT: A Context Link (CL) corresponding to a Web page, an Importance Table.

OUTPUT: The relevance score (OBRS) for each Context Link (CL).

Step1: Initialize the relevance score of the Context Link (CL) to 0 i.e. OBRS=0.

Step2: Select first term (T) and corresponding Importance (IMP) from the Importance Table.

Step3: Calculate how many times the term (T) occurs in the Context Link (CL). Let the number of occurrence is calculated in COUNT.

Step4: Multiply the number of occurrence calculated in step 3 with the Importance IMP. Let call this TERM_IMP. And TERM_IMP=COUNT * IMP

Step5: Add this term importance to OBRS. So new OBRS will be, OBRS = OBRS + TERM_IMP.

Step6: Select the next term and weight from Importance table and go to step3, until all the terms in the weight table are visited.

Step7: End.

G. LARS GENERATOR

Pirkola [1] pointed that for crawling based upon topic it should make use of historical accesses to that particular domain The proposed crawler makes use of adaptive Rules [3] derived from the classes and link access to improve the crawl and to handle the situation where an irrelevant link in a page may further point to relevant page. For doing this crawler's classifier component is trained with a class and taxonomy, and a set of example documents for each class and call this as train-0 set. Next for each class in the train-0 set we gather all Web pages that the example Web pages in the corresponding class point to. Now we have a collection of class names train-1

set and a set of fetched pages for each class. We know the class distribution of pages to which the documents in each train-0 set class point. For each class in the set train-0, we count the number of referred classes in corresponding train-1 set and generate rules of the form $C_i \to C_j(P)$, means a page of class C_i can point to a page of class C_j with probability P. P is the ratio of train-1 pages in C_i to all train-1 pages that C_j pages in train-0 refer to. The proposed crawler while seeking Web pages of class C_i attaches priority score P to the pages that it encounters. To demonstrate the approach example is presented. The taxonomy includes four classes:

- Computer Sc. and Engineering (CSE)
- Information Technology (IT)
- Engineering (ENGG)
- Computer (COMP)

Train-0 set can be constructed from any existing directory. Next, for each class , we retrieve the pages that this class's example pages refer to. Assume that we fetch 10 such pages example pages for each class in the train-0 set and that the class distribution among these newly fetched pages i.e train-1 set is as listed in Table 2. Then we can obtain the rules of Table 3.

Adaptive rules may support the tunneling for longer paths using simple application of transitivity among the rules. Also this mechanism is independent of a page's similarity, but rather relies on the probability that a given page's class refer to a target class. This P acts as the Look Ahead Score for the CORE. It can be further manipulated to obtain finer results.

Table 2: Class distribution into train-1 set for each class in train-0 set

CSE	***IT***	***ENGG***	***COMP***
8 URLs for IT	2 URLs for CSE	3 URLs for CSE	10 URLs for COMP
1 URL for ENGG	4 URLs for IT	4 URLs for IT	
1 URL for COMP	4 URLs for ENGG	3 URLs for ENGG	

Table 3: Adaptive Rules for the distribution in Table 1, (the number following each rule is the probability P)

$CSE:$ $CSE \to IT(0.8)$ $CSE \to COMP(0.1)$ $CSE \to ENGG(0.1)$
$IT:$ $IT \to CSE(0.2)$ $IT \to IT(0.4)$ $IT \to ENGG(0.4)$
$ENGG:$ $ENGG \to CSE(0.3)$ $ENGG \to IT(0.4)$ $ENGG \to ENGG(00.3)$
$COMP:$ $COMP \to COMP(1.0)$

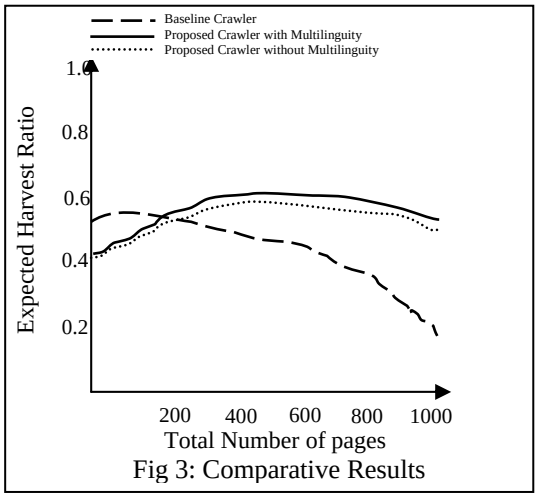

Fig 3: Comparative Results

IV. RESULTS

The proposed crawler is being simulated with 1000 pages and an ontology repository containing ontologies related to all engineering branches under a particular university. And for each test case 0.2 X number of pages out of total X pages are written in mixed way (i.e in English and Hindi), such that the important terms in these 0.2 X pages appears in Hindi. By taking the values of Min_OntRel=5, and Min_LaRel=3 the simulated results are shown in Fig 3. It is a plot of number of relevant pages found against the total number of pages downloaded i.e plot for harvest rate for the baseline focused crawler, non-multilingual context ontology rule enhanced focused web crawler, and multilingual context ontology rule enhanced focused web crawler.

V. CONCLUSION

A multilingual focused web crawler is presented that makes use of context ontology score and adaptive classification rules for relevancy calculation, and multilingual transcriptor to deal with the intermediate multilinguity. The proposed crawler can deliver improved search results by going through the intermediate multilingual documents. Ontologies served as efficient concepts representation technique. The proposed crawler shows improved harvest ratio when tested for retrieving information for courses run by particular university. It can be used in digital libraries for retrieving documents distributed in the form of documents written in a mixed way. Further work could be done to replace the multilingual transcriptor with multilingual translator for

semantic matching of the intermediate text, which can further enhance the search results.

REFERENCES

[1]. Ari Pirkola, 2007, "Focused Crawling: AMeans ToAcquire Biological Data from the Web**",** VLDB '07, September 23-28, Vienna, Austria, ACM.

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

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

[4]. Jaime G.Carbonell,Yimming Yang,Robert E. Frederking,Ralf D.Brown, Yibing Geng, and Danny lee. Translingual information Retrieval: a comparative evaluation. In IJCAI(1), pages 708-715,1997.

[5]. J. Cho, H. Garcia-Molina, L. Page. , April 1998 "Efficient Crawling Through URL Ordering" In Proceedings of the 7th International WWW Conference, Brisbane, Australia.

[6] Junghoo Cho, Hetor Gasrcia-Molina, WWW 2002 "Parallel Crawlers".

[7].K.Bharat and A. Broder. A technique for measuring the relative size and overlap of public web search engines. In Proc. of the 7th WWW Conference 1998.

[8]. L. Page, S. Brin, R. Motwani, T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web", Stanford Digital Library Technologies Project.

[9]. M.Ehrig, A. Maedche., 2003 "Ontology-focused Crawling of Web Documents", In Proceedings of the ACM symposium on Applied computing.

[10] M.Yuvrani, N.Ch.S.N.Iyengar, A.Kanan, 2006 *"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.

[11] P.M.E. De Bra and R.D.J.Post, "Information retrieval in the World-Wide Web: Makingclient-based searching feasible", Computer Networks and ISDN Systems. vol. 27, no. 2, pp. 183-192.

[12] S. Chakrabarti, M. van den Berg, B. Domc,1999, "Focused crawling: a new approach to topic-specific Web resource discovery", *Proceedings of the 8th international World WildWeb Conference,* Toronto, Canada.