A New Approach to Design Domain Specific Ontology Based Web Crawler

Debajyoti Mukhopadhyay, Arup Biswas, Sukanta Sinha

Web Intelligence & Distributed Computing Research Lab, Techno India Group
West Bengal University of Technology
EM 4/1, Salt Lake Sector V, Calcutta 700091, India
{debajyoti.mukhopadhyay, biswas.arup, sukantasinha2003}@gmail.com

Abstract

A domain specific Web search engine is a search engine which replies to domain specific user queries. The crawler in a domain specific search engine must crawl through the domain specific Web pages in the World Wide Web (WWW). For a crawler it is not an easy task to download the domain specific Web pages. Ontology can play a vital role in this context. Our focus will be to identify Web pages for a particular domain in WWW.

1. Introduction

A search engine is a document retrieval system which helps find information stored in a computer system, such as in the World Wide Web (WWW), inside a corporate or proprietary network, or in a personal computer. Surveys indicate that almost 25% of Web searchers are unable to find useful results in the first set of URLs that are returned. The term *ontology* [1] is an old term used in the field of Knowledge Representation, Information Modeling, etc. Typically ontology is a hierarchical data structure containing relevant entities, relationships and rules within a specific domain. Tom R. Gruber [2] defines ontology as a *specification of a conceptualization*.

2. Web Search Crawling

A standard crawler crawls through all the pages in breadth first strategy. So if we want to crawl through some domain then it will be very inefficient. In Figure 1 we show the general crawler crawling activity.

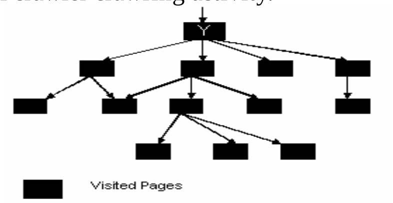

Fig.1. Standard Crawling

If some crawler crawls only through domain specific pages then it is a focused crawler. From Figure 2 we can see that a focused crawler crawls through domain specific pages. The pages which are not related to the particular domain are not considered.

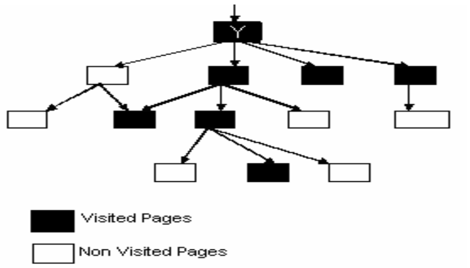

Fig.2. Focused (Domain Specific) Crawling

3. Our Approach

In our approach we crawl through the Web and add Web pages to the database, which are related to a specific domain (i.e. a specific ontology) and discard Web pages which are not related to the domain. In this section we will show how to determine domain specific page.

3.1 Relevance Calculation

In this section we describe our own algorithm depending on which we calculate relevancy of a Web page on a specific domain.

3.1.1 Weight Table. We want to add some weights to each term in the ontology. The strategy of assigning weights is that, the more specific term will have more weight on it. And the terms which are common to more than one domain have less weight. The sample Weight table for some terms of a given ontology of the table shown below:

Ontology terms	Weight
Assistant Professor	1.0
Assistant	0.6
Student	0.4
Worker	0.1
Publication	0.1

Fig.3. Weight table for the above ontology

0-7695-3068-0/07 $25.00 © 2007 IEEE
DOI 10.1109/ICIT.2007.20

3.1.2 Relevance calculation algorithm. In this section we design an algorithm how relevance score of a Web page is calculated.

INPUT: A Web page (P), a weight table.

OUTPUT: The relevance score of the Web page (P).

Step1 Initialize the relevance score of the Web page (P) to 0. RELEVANCE_P=0.

Step2 Select first term (T) and corresponding weight (W) from the weight table.

Step3 Calculate how many times the term (T) occurs in the Web page P. Let the number of occurrence is calculated in COUNT.

Step4 Multiply the number of occurrence calculated at step3 with the weight W. Let call this TERM_WEIGHT. And TERM_WEIGHT=COUNT * W.

Step5 Add this term weight to RELEVANCE_P. So new RELEVANCE_P will be, RELEVANCE_P = RELEVANCE_P + TERM_WEIGHT.

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

Step7 End.

In Figure 4 we have shown an example of the above algorithm.

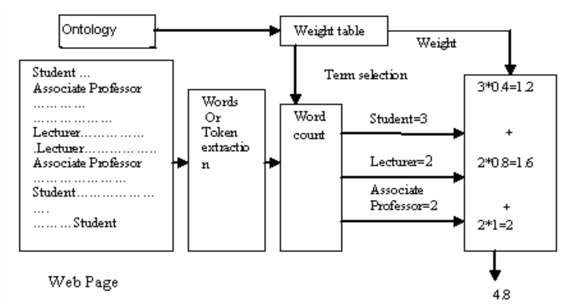

Fig.4. Relevance calculation of a Web page

From the Figure we can see that the total relevance of the Web page is 4.8(1.2+1.6+2.0). If our relevance limit is 4 then we can add this page to our database as a domain specific page.

This algorithm can be used as ranking algorithm also. If two pages P1 and P2 got relevance x, y respectively and x>y, then we can say that P1 is more important than P2.

3.2 Checking Domain of a Web page

Using ontological knowledge we can find relevant Web pages from the Web. See Figure 5.

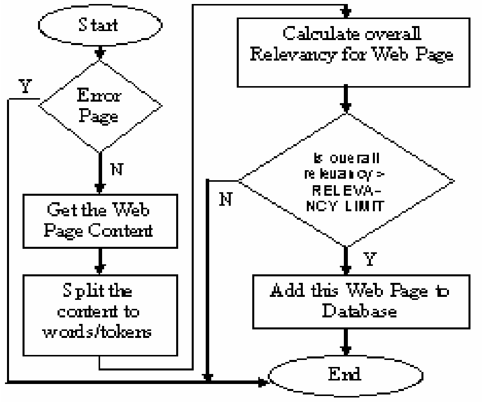
Start
Error Page
Y
N
Get the Web Page Content
Split the content to words/tokens
Calculate overall Relevancy for Web Page
Is overall relevancy > RELEVA-NCY LIMIT
N
Y
Add this Web Page to Database
End

Fig.5. Checking Domain of a Web page

3.3 Challenge Faced

In our approach we go along the link what are found in domain specific pages. From the Figure 6 we can see that at level 3 there are some irrelevant pages which are discarding domain specific pages at level 4 and 5 from the crawling path. As a solution of this problem we design an algorithm whose working principal are shown below:

Step1 If a Web page is not relevant then goto step2. Initialize the level value of the URL to 0.

Step2 Extract URLs from the Web page and add to the IRRE_TABLE with level value.

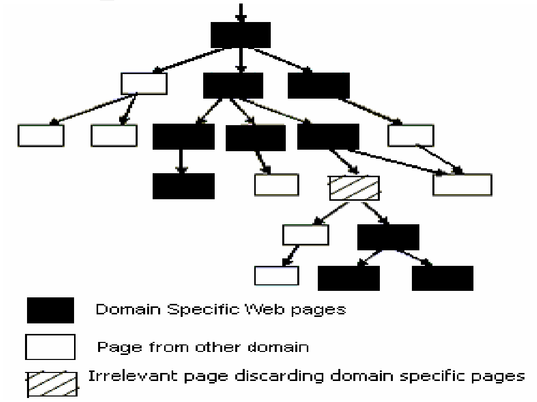
Domain Specific Web pages
Page from other domain
Irrelevant page discarding domain specific pages

Fig.6. Challenge in our approach.

Step3 Until all the IRRE_TABLE entry are visited, take one entry from IRRE_TABLE and goto step4.

Step4 If the level value of the URL is less than or equal to the tolerance limit then go to step5 otherwise discard the URL.

Step5 Calculate relevance for the Web page.

Step6 If the page is relevant to the domain then add it to the database, else goto step7.

Step7 Increase the level value of the URL and goto step2.

The value of the tolerance limit is very imp ortant, for this to get an optimal performance of the crawler we took an optimal value of tolerance limit.

4. Performance Analyses

In this section we describe a performance measure and describe the performance of our system according to the performance measure.

4.1 Harvest rate

Harvest rate [3] is a common measure on how well a focused crawler performs. It is expressed as HR= r/p, where HR is the harvest rate, *r* is the number of relevant pages found and *p* is the number of pages downloaded.

4.2 Test Settings

In this section we will describe different parameter settings to run the crawler.

4.2.1 Scope Filter. In order to ensure that our crawler only downloads files with textual content, not Irrelevant files like images and video, we have added a filter when performing the tests.

4.2.2 Seed URLs. For the crawler to start crawling we provide some seed URLs.
http://www.jnu.ac.in, http://www.annauniv.edu, http://www.wbut.net , http://www.vu.ernet.in, http://en.wikipedia.org/wiki/University ,

4.3 Test Results

In this section we have shown some test results through graph plot.

4.3.1 Harvest Rate for Unfocused Crawling. From the Figure 7 we can see that, general crawlers performance to crawl through a specific domain is not efficient, it crawl through a large number of Web pages but finds very few domain specific Web pages.

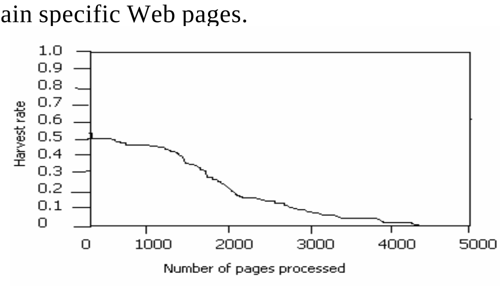

Fig.7. Harvest rate for unfocused crawling.

4.3.2 Harvest Rate For Relevance Limit 5 and Tolerance Limit 3. The harvest rate looks like Figure 8. From the Figure we can see that, after starting with some domain specific seeds, maximum crawled pages are domain specific.

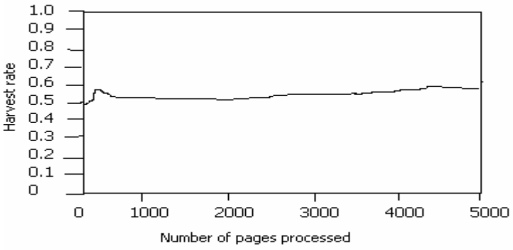

Fig.8. Harvest rate of focused crawler

In Figure 9 we have shown harvest rate for focused crawler with relevance limit 5 and tolerance limit 10. In the figure there are lots of ups and down. So it is important to set the tolerance limit to an optimal value. We have to set the tolerance limit to such a value so that the average harvest rate becomes satisfactory.

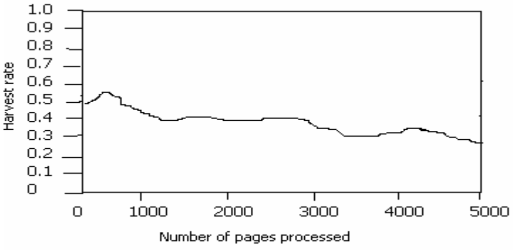

Fig.9. Harvest rate for focused crawling with tolerance limit 5 and relevance limit 5

5. Conclusions

Though new improved keyword-based technologies for searching the WWW are evolving constantly, the growth rate of these improvements is likely to be slight. Search engines based on a new concept as the domain specific search technology, are effectively able to handle the above mentioned problems.

References

[1]*Wikipedia:http://en.wikipedia.org/wiki/Ontology_(computer_science)*

[2]T.R.G ruber, "What is an Ontology?," http://wwwksl.stanford.edu/kst/what-is-an-ontology.html

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

[4] Swoogle 2005. http://swoogle.umbc.edu/.

[5] W.Roush, Search beyond Google. Technology Review; http://www.technologyreview.com/articles/print_version/roush0304.asp.