A New Approach to Design Graph Based Search Engine for Multiple Domains Using Different Ontologies

Debajyoti Mukhopadhyay[1,3], Sukanta Sinha[2,3]
[1]Calcutta Business School, D.H. Road, Bishnupur 743503, India
[2]Tata Consultancy Services,Whitefield Rd, Bangalore 560066, India
[3] WIDiCoReL, Green Tower C- 9/1, Golf Green, Kolkata 700095, India
{ debajyoti.mukhopadhyay, Sukantasinha2003}@gmail.com

Abstract

Search Engine has become a major tool for searching any information from the World Wide Web (WWW). While searching the huge digital library available in the WWW, every effort is made to retrieve the most relevant results. But in WWW majority of the Web pages are in HTML format and there are no such tags, which tells the crawler to find any specific domain. To find more relevant result we use Ontology for that particular domain. If we are working with multiple domains then we use multiple ontologies. Now in order to design a domain specific search engine for multiple domains, crawler must crawl through the domain specific Web pages in the WWW according to the predefined ontologies.

1. Introduction

In this paper, we discuss the basic idea of a graph based searching and describe a design and development methodology for multiple domain specific search engine based on multiple ontology matching and relevance limits, which not only overcomes the problem of knowledge overhead but also supports conventional queries. Further, it is able to produce exact answer from the graph that satisfies user queries.

2. Domain Specific Web Search Crawling

In this section we describe working principle of a single domain specific crawler and multiple domains specific crawler.

2.1 Single Domain Specific Crawler

In domain specific Web search crawler, the crawler crawls down the pages, which are relevant to our domain. To find the domain we need to visit all the Web pages and calculate the relevance value. Now consider the situation where the page is not related to the given domain but it belongs to another domain. For this scenario want to offer a new proposal to working with multiple domains. In Figure 1 we show the single domain specific crawler crawling activity.

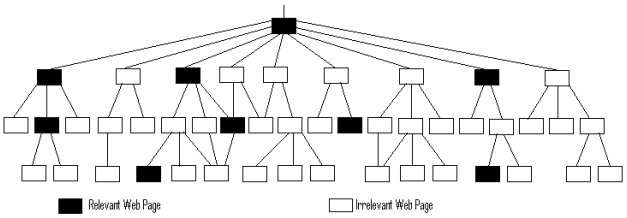

Figure 1. Single Domain Specific Crawling

2.2 Multiple Domains Specific Crawler

In multiple domains specific Web search crawler crawls down the Web pages and checking multiple domains simultaneously by using multiple Ontology terms

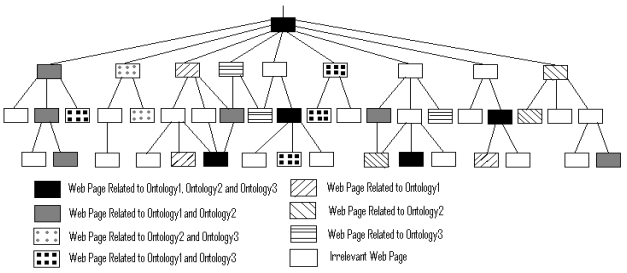

Figure 2. Multiple Domains Specific Crawling

and finding which page is related to which domain. The Web page not only related to the single domain but also it may be related with multiple domains. In our approach we taking a track to finding a Web page related to how many domains and what are their relevance scores. In Figure 2 shown how we are working with multiple domains.

3. Ontology Based Domain Specific Crawling

Ontology based domain specific crawler means, a crawler that find domain specific Web pages using that domain ontology. The term *Ontology [4]* is a data model that represents a set of concepts within a domain and the relationships between those concepts. It is used to reason about the objects within that domain. Ontologies are used in artificial intelligence, the Semantic Web, software engineering, biomedical informatics, Library Science, and information architecture as a form of knowledge representation about the world or some part of it.

978-0-7695-3513-5/08 $25.00 © 2008 IEEE
DOI 10.1109/ICIT.2008.46

4. WordNet

WordNet [6] is a semantic lexicon for the English language. A *semantic lexicon* is a dictionary of words labeled with semantic classes so associations can be drawn between words that have not previously been encountered. WordNet groups' English words into sets of synonyms called *synsets*, provides short, general definitions, and records the various semantic relations between these synonym sets. The purpose is twofold: to produce a combination of dictionary and *thesaurus* that is more intuitively usable, and to support automatic text analysis and artificial intelligence applications. A thesaurus is an indexed compilation of words with similar, related and opposite meanings.

Syntable is one type of table which contains synonyms of all ontology terms in a table. We generate a syntable for each ontology to generate more accurate relevance score of a Web page.

5. Proposed Approach

In our approach we crawl through the Web and add Web pages to the database which are related to our specified domains (i.e. related to our specified ontologies) and discard Web pages which are not related to our domains. To finding the domains we calculate relevance scores of that Web page for all domains. In this section we will show the process of relevance calculation using multiple ontologies as described in section 3 and how this process can be used to determine whether a page is related to our specified domains or not.

5.1. Relevance Calculation for All Domains

In this section we describe our own algorithm depending on which we calculate relevancy of a Web page for multiple domains. Our algorithm is very simple and very effective as well. Here we assign some weight to all the ontology terms and use these weights for relevance calculation for each domain.

5.1.1. Weight Table. Weight table is a table, which is constructed using the given ontologies. This table contains two columns; one column for the ontology terms and another for term corresponding weights. For a term which belongs to more than one domain. 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 in Figure 3.

Wicket	1.0
Bat	1.0
Crease	0.8
Test Match	0.6
One day Match	0.4
Ball	0.2
Ground	0.1
Player	0.1

Figure 3. Weight table for some terms in cricket ontology

5.1.2. Relevance calculation algorithm. In this section we design an algorithm which calculates relevance scores of a Web page for multiple domains. Each domain represents by an ontology. Here we are taking a Web page, weight tables for each domain ontology terms and syntables for each domain ontology terms as input. And the following algorithm calculates relevance scores for each domain.

INPUT: A Web page (σ), Weight Tables for each domain and SynTables for each domain.
OUTPUT: The Relevance score of the Web page (σ) for each domain.
Step1 Initialize the relevance scores of the Web page (σ) for each domain to 0. $REL\_ONT_1\_\sigma = 0$, $REL\_ONT_2\_\sigma = 0$ and $REL\_ONT_3\_\sigma = 0$.
Step2 Select first common ontology term (α) for all domains ($Ontology_1$, $Ontology_2$ and $Ontology_3$) and corresponding SynTerms $s\alpha_1$, $s\alpha_2$ and $s\alpha_3$ from SynTables of $Ontology_1$, $Ontology_2$ and $Ontology_3$ respectively and also select their term weights from their weight tables.
Step3 Calculate how many times α, $s\alpha_1$, $s\alpha_2$ and $s\alpha_3$ occurs in the Web page (σ).
Step4 Multiply the number of occurrence with their corresponding weights and call them TERM_WEIGHT. Add these term weights to their respective domains relevance value.
Step5 Select the next common term and their SynTerms for all domains and their weights from weight tables and go to Step3 until all the common terms for all domains are visited.
Step6 Select ontology term (β_1) from the remaining ontology terms and β_1 exists in both $Ontology_1$ and $Ontology_2$ and corresponding SynTerms $s\beta_{11}$ and $s\beta_{12}$ from SynTables of $Ontology_1$ and $Ontology_2$ respectively and also select their term weights from their weight tables.
Step7 Calculate how many times β_1, $s\beta_{11}$ and $s\beta_{12}$ occurs in the Web page (σ).
Step8 Multiply the number of occurrence with their corresponding weights and call them TERM_WEIGHT. Add these term weights to their respective domains relevance value.
Step9 Select the next common term and their SynTerms for $Ontology_1$ and $Ontology_2$ and their weights from weight tables and go to Step7 until all the common terms for $Ontology_1$ and $Ontology_2$ are visited.

Step10 Select ontology term (β_2) from the remaining ontology terms and β_2 exists in both $Ontology_2$ and $Ontology_3$ and corresponding SynTerms $s\beta_{22}$ and $s\beta_{23}$ from SynTables of $Ontology_2$, and $Ontology_3$ respectively and also select their term weights from their weight tables.
Step11 Calculate how many times β_2, $s\beta_{22}$ and $s\beta_{23}$ occurs in the Web page (σ).
Step12 Multiply the number of occurrence with their corresponding weights and call them TERM_WEIGHT. Add these term weights to their respective domains relevance value.
Step13 Select the next common term and their SynTerms for $Ontology_2$ and $Ontology_3$ and their weights from their weight tables and go to Step11 until all the common terms for $Ontology_2$ and $Ontology_3$ are visited.
Step14 Select ontology term (β_3) from the remaining ontology terms and β_3 exists in both $Ontology_1$ and $Ontology_3$ and corresponding SynTerms $s\beta_{31}$ and $s\beta_{33}$ from SynTables of $Ontology_1$ and $Ontology_3$ respectively and also select their term weights from their weight tables.
Step15 Calculate how many times β_3, $s\beta_{31}$ and $s\beta_{33}$ occurs in the Web page (σ).
Step16 Multiply the number of occurrence with their corresponding weights and call them TERM_WEIGHT. Add these term weights to their respective domains relevance value.
Step17 Select the next common term and their SynTerms for $Ontology_1$ and $Ontology_3$ and their weights from their weight tables and go to Step15 until all the common terms for $Ontology_1$ and $Ontology_3$ are visited.
Step18 Select remaining ontology terms γ_1, γ_2 and γ_3 for $Ontology_1$, $Ontology_2$ and $Ontology_3$ respectively and corresponding SynTerms $s\gamma_{11}$, $s\gamma_{22}$ and $s\gamma_{33}$ from SynTables of $Ontology_1$, $Ontology_2$ and $Ontology_3$ respectively their weights from their weight tables.
Step19 Calculate how many times γ_1, γ_2, γ_3, $s\gamma_{11}$, $s\gamma_{22}$ and $s\gamma_{33}$ occurs in the Web page (σ).
Step20 Multiply the number of occurrence with their corresponding weights and call them TERM_WEIGHT. Add these term weights to their respective domains relevance value.
Step21 Select the next terms and their SynTerms for all domains and their weights from their weight tables and go to Step19 until all the terms for all domains are visited.
Step22 End.

In Figure 4 we describe how the above algorithm works to calculate relevance scores. First we take ontology terms for different domains. Then we were finding common terms for minimizing comparison. We extracts the terms (α) which belongs to all domains (here we working with three domains). Then find the terms (β_1, β_2 and β_3) from the remaining ontology terms which belongs to any two ontologies i.e. two domains and the remaining terms (γ_1, γ_2 and γ_3) belong to a single domain. All terms have a weight and it varies domain to domain for a single term. Each ontology term has an entry to syntable which contains the synonyms of the ontology terms. Here syntables are $WordNet_1$, $WordNet_2$ and $WordNet_3$ and weight tables are $WeightTable_1$ $WeightTable_2$ and $WeightTable_3$.

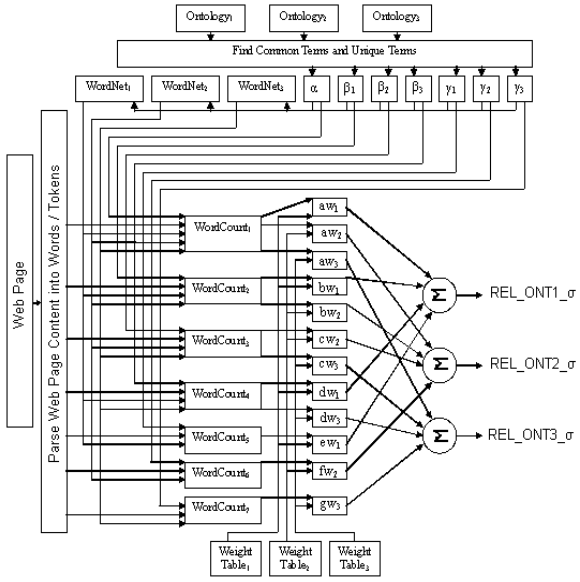

Figure 4. Relevance calculation of a Web page

From Figure 4 we can see that the relevance scores of the Web page (σ) are $REL\_ONT_1\_\sigma$, $REL\_ONT_2\_\sigma$ and $REL\_ONT_3\_\sigma$ for three domains.

5.2. How to Collect Relevant Pages from Irrelevant Links

In our approach we go along the link what are found in domain specific pages. We are not checking link found in the irrelevant pages. If some domain specific pages are partitioned by some irrelevant pages which are not of the specific domain then, the performance of the crawler will degrade. From the Figure 6 we can see that at level 1 there are some irrelevant pages which are discarding domain specific pages at level 2 and 3 from the crawling path. If we can't process those pages then the performance of the crawler will degrade. As a solution [7] of this problem we take a tolerance limit. When some page is irrelevant then the URLs found in the Web page are stored in a different table we call this table as *IRRE_TABLE*. We crawl down through those URLs in *IRRE_TABLE* up to the tolerance limit level.

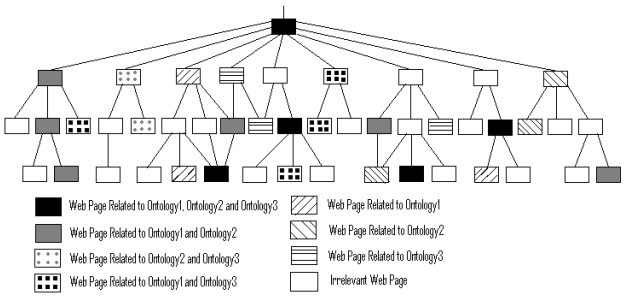

Figure 6. Challenge in our approach

5.3. Checking Domain of a Web page

Using ontological knowledge we can find relevant Web pages from the Web. When the crawler finds a new page then it calculate the relevance of the Web page (i.e. it compares the content of the Web page with ontological knowledge). If the calculated relevance is more than a predefined relevance then we called the Web page is of the specific domain. If a Web page overcomes all the relevance limits for all domains then we called the Web page belongs to all domains.

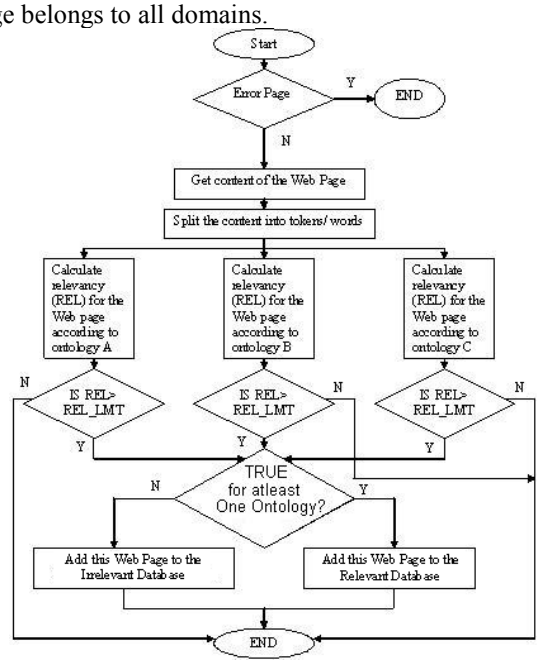

Figure 5. Checking Domain of a Web-page

5.4. Generation of Graph

In this section we design an algorithm which generates a weighted graph according to their domains. Each domain or ontology represented by a node and these nodes are plot on the 2D plane. Now we considering another node into space that represents common pages i.e. the page belong to all domains. And the following algorithm generates a weighted graph for all domains.

INPUT: A set of Web pages and their relevance scores for all domains.
OUTPUT: A Weighted Graph.

Step1 Assign node for each Ontology. Here we assign A, B and C for $Ontology_1$, $Ontology_2$ and $Ontology_3$ respectively and assign another node D in space for storing all domains related pages. Each node and each edge in the plane contains a Database and the edges in space are contains a weight. Initially all the Databases are blank and all weights in the space edges are 0.
Step2 Find out the Web pages which are relevant to only one domain i.e. relevance score cross the relevance limit for only one domain and Store the Web pages in the respective node Database.
Step3 Find out the Web pages which are relevant to all domains i.e. relevance score cross the relevance limit for all domains and Store the Web pages into the space node (i.e. node D).
Step4 Count number of pages in the space node and assign the space edge weights by the count value.
Step5 Find out the Web pages which are relevant to any two domains i.e. relevance score cross the relevance limit for any two domains and Store the Web pages in the respective edge Database.
Step6 End.

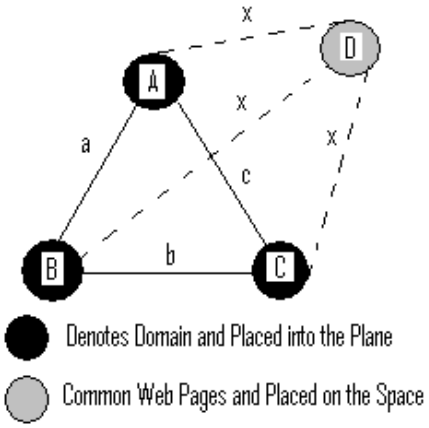

Figure 7. Graph representation of Web pages.

In the above graph *A, B* and *C* represent three domains and contain a database of single domain pages. *a, b* and *c* are weights which contains set of web pages. '*a'* contain *A* and *B* domain related pages. '*b'* contain *B* and *C* domain related pages. '*c'* contain *A* and *C* domain related pages. And '*x'* contains number of pages in *D.* node *D* contains pages which belongs to all domains.

5.5. User Interface

In Figure 8 shows a part of User Interface in our search engine. Initially *Go* button can't appear in the User Interface. First we put a search query into the *Input String Box* then select domains. After domain selection *Go* button appears on the screen. Here we are working with

three domains *Cricket, Football* and *Hockey*. These domains are very closer to each other and our challenges are to find pages from such close domains.

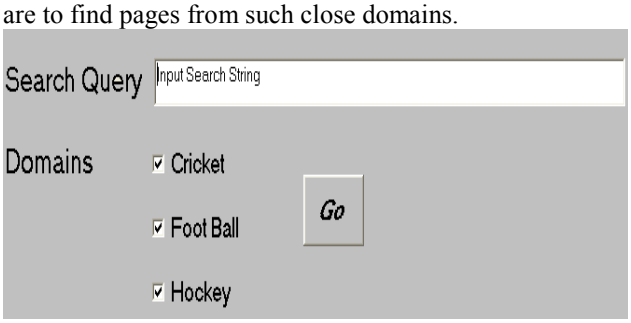

Figure 8. A Part of User Interface.

6. Performance Analyses

In this section we describe our test settings and describe the performance of our system.

6.1. Test Settings

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

6.1.1. Seed URLs. For the crawler to start crawling we provide some seed URLs depending on the Ontologies. http://www.hindustantimes.com, http://www.cricket-time.com, http://www.sportsofworld.com, http://icc-cricket.yahoo.com, http://www.hockeygiant.com, http://www.whockey.com, http://www.fifa.com, http://www.webindia123.com/sports/hockey/index.htm, http://www.footballtransfers.info, http://www.napit.co.uk, http://www.footballguys.com.

6.1.2. Syntable. Synonyms for each Ontology terms are shown in Figure 9, 10 and 11. The Syntables are constructed using different Ontologies. This table contains two columns; one column for Ontology terms and another for synonyms of that term. Here NA defines no such synonyms are present there.

National Match	Intra state game
Not Out	Batting
Off Stump	Right side Wicket
One day	50 over match
Out	Dismissed

Figure 9. Syntable for Cricket Ontology

Center	middle
Centre Circle	NA
Club	Association
Corner	area
Crowd	mass

Figure 10. Syntable for Foot-Ball Ontology

Defender	protector
Draw	NA
Elbow Pads	NA
EQUIPMENTS	Apparatus
Field Hockey	NA

Figure 11. Syntable for Hockey Ontology

6.1.3. Weight Table. Weight for each Ontology terms is shown in Figure 12, 13 and 14. The weight tables are constructed using different Ontologies. This table contains two columns; one column for Ontology terms and another for weight of that term.

Not Out	0.8
Off Stump	0.8
One day	0.4
National Match	0.1

Figure 12. Weight table for Cricket Ontology

Free kick	0.8
Centre Circle	0.4
Center	0.2
Crowd	0.1

Figure 13. Weight table for Foot-Ball Ontology

Field Hockey	0.9
Hockey Stick	0.9
Elbow Pads	0.6
Draw	0.1

Figure 14. Weight table for Hockey Ontology

6.2. Test Results

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

6.2.1. Page Distribution in Different Domains. Figure16 shows page distribution of each domain.

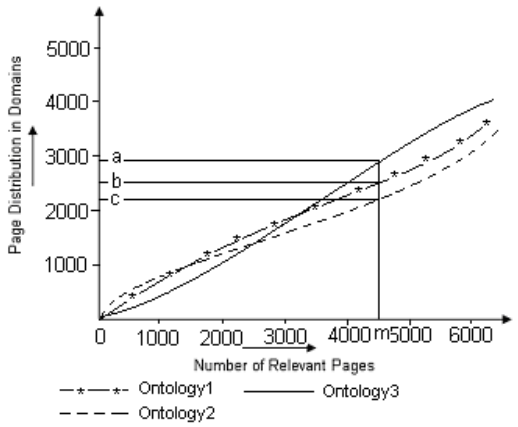

Figure 16. Page Distribution in Domain wise.

From the figure we conclude that one page must belong to more than one domain. Here *m* is number of relevant pages and *b, c* and *a* number of relevant pages belong to domain *1, 2* and *3* respectively and m always less than equal to (*a*+*b*+*c*).

6.2.2. Performance of multiple domains crawling over single domain crawling. From the Figure 15 we can see that, single domain specific crawler crawling time is more than the multiple domains specific crawler crawling time.

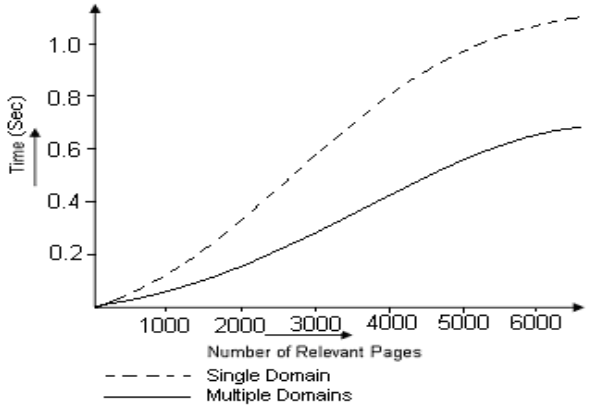

Figure 15. Time taken in Single Domain Crawling and Multiple Domains crawling.

When we work through large number of Web pages in single domain specific crawler, most of the Web pages are irrelevant and we discard those pages but in multiple domains specific crawler, most of the pages does not irrelevant page, it belongs to any one domain and if these domains are match with our domains then our crawler performance increase.

7. Conclusions

Web searchers faced major problems by imprecise and irrelevant results, especially with the continued expansion of the Web. For this we incorporate domain specific concept for crawling Web pages from WWW.

In our experiment, we have developed a prototype that uses multiple ontologies to perform multiple domains specific crawling. The prototype uses information of a specified domains are kept in structure way into ontology to guide the crawler in its search for Web pages that are relevant to the topics specified in ontologies. Firstly, our approach has been able to successfully eliminate the problem of irrelevant results which is one of the main problems encountered by the users of a regular search engine. By searching domain specific Web pages the search engine effectively fetches the exact information. Secondly, by producing exact information as the result, the search engine eliminates the need to go through numerous results as in case of a regular search engine. Finally, our design although based on three domains, is highly scalable and can be easily adopted by other enterprises as their site search tool. This would only require the enterprise to feed in the relevance limit, weight tables based on the ontology of the different domains.

8. References

[1] Berners-Lee.T,1999. *Weaving the Web:The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*,New York:Harper SanFrancisco.

[2] A Guide to Creating Your First Ontology:Natalya F. Noy and Deborah L. McGuinness; Stanford U. Report.

[3] T. R. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*,5(2):199-220, 1993.

[4] *N. F. Noy, D. L. McGuinness, "Ontology Development 101: A Guide* to Creating Your First Ontology" Available on:http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noymcguinness.html [Accessed May 2005]

[5] *Swoogle 2005*. http://swoogle.umbc.edu/.

[6] WordNet. http://en.wikipedia.org/wiki/WordNet

[7] Debajyoti Mukhopadhyay, Arup Biswas, Sukanta Sinha; *A New Approach to Design Domain Specific Ontology Based Web Crawler;* 10th International Conference on Information Technology, ICIT 2007 Proceedings; Bhubaneswar, India; IEEE Computer Society Press, California, USA; December 17-20, 2007; pp.289-291.

[8] Tim Bray, *What is RDF?* http://www.xml.com/lpt/a/2001/01/24/rdf.html

[9] W3C, *RDF Primer*, W3C Working Draft 23 January 2003, http://www.w3.org/TR/2003/WD-rdf-primer-20030123/

[10] WordNet. http://en.wikipedia.org/wiki/George_A._Miller