IOSR Journal of Engineering
Apr. 2012, Vol. 2(4) pp: 819-823

Semantic Web Crawler Based on Lexical Database

Anthoniraj Amalanathan , Senthilnathan Muthukumaravel
School of Computing Science and Engineering, VIT University, Vellore, India – 632014.

ABSTRACT
Crawlers are basic entity that makes search engine to work efficiently in World Wide Web. Semantic Concept is implied into the search engine to provide precise and constricted search results which is required by end users of Internet. Search engine could be enhanced in searching mechanism through semantic Lexical Database such as WordNet, ConceptNet, YAGO, etc; Search results would be retrieved from Lexical and Semantic Knowledge Base (KB) by applying word sense and metadata technique based on the user query. The Uniform Resource Locater (URL) could be added and updated by the user to Semantic knowledge base so that crawlers can easily extract meta data and text which is available in specified web page. The proposed methodology enables web crawler to extract all meta tags and metadata from the web page which are stored in Semantic KB, hence search results are expected to be more significant and effective.

Keywords **– Crawler, Knowledge Base, Lexical Database, Metadata, Semantic**

I. INTRODUCTION

The Internet is a huge collection of various categories of websites, which is growing tremendously day by day through adding number of websites to it, Web sites are indexed into search engines through special process known as crawling and it helps the search engines to provide the results based on user query request. Query may be refined through a special query processor that populates the search result in accordance with the lexical database which imparts the sense and other data. It is expected that the search results would be more precise and perfect if we add semantic techniques to search engine.

1.1. Crawler

Crawlers are used to extract information from a website; also various types of crawling methods are available to obtain data from a website.

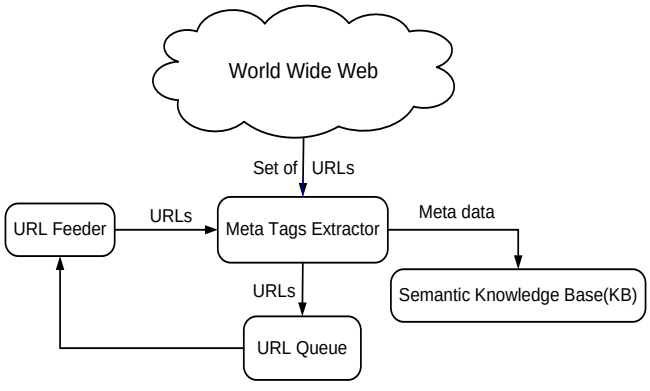

Fig.1. Semantic Web Crawler

Basically web crawlers fetch title, Meta data and content of the webpage and also website links or URL are retrieved recursively from one page to another page with some constraints. Crawling can be done to retrieve precise information from the webpage in a specific domain. In this paper semantic meta crawler technique is used to invoke the meta tags that are available in prescribed form of the website where those meta tags consists of Author name, Description, Keywords and Geographical position of the web page. Most Probably Keywords and Description would absolutely present in web pages which is used for adding semanticness to web crawlers.

1.2. Lexical Database

Lexical database is a large collection of Synonyms, Holonyms, Meronymy, Antonyms of English words. Lexical database also provides the synonym or sense of a given word which is called as Synset[1]. Holonyms is the relationship between a term denoting a part or a member of, the synonym, Meronymy is just opposite for Holonyms, Antonyms will provide alternate opposite meaning for synonym of a word. WordNet[2], ConceptNet[3] and YAGO[4] are best and widely used Lexical Database. These Lexical databases are mandatory to obtain the Semantic relationship and creating semantic knowledge base of words that are retrieved from Meta data.

II. MOTIVATION AND RELATED WORKS

Based on various methodologies, there are different types of search engines available online such as Page Ranking Algorithm, Focused Crawling Algorithm, Deep Crawling Algorithm, Path-ascending crawling and Breadth First Search Algorithm. In that, Google, Yahoo and Bing are most widely used search engines by using their own crawling algorithm as a highly confidential business secret. Few standard crawling algorithms are discussed in the rest of the section.

2.1. Page Rank

Page Rank algorithm has been designed such that the known relationships between web pages are taken into account. For example, if page P1 has a link to page P2, then, P2's subject is probably interesting for P1's creator. Therefore, the number of input links to a web page shows the interest degree of the page to others. Obviously, the interest degree of a page increases with the growing number of input links.

$$PR(\mathbf{P1}) = PR(\mathbf{A1})/L(\mathbf{A1}) + ... + PR(\mathbf{An})/L(\mathbf{An})$$

In order to find the Page Rank for a page, called **P1**, we need to find all the pages that linked to page **P1** and Out

Link from **P1**. We found a page **A1**, which has link from **P1** then page L(**A1)** will give no. of Outbound links to page **P1**. We do the same for **A2**, **A3** and all other pages linking to Main page **P** – and Sum of the values will provide Rank of the web page.

Moreover, when a web page receives links from an important page then certainly it should have a high rank. Therefore, Page Rank of a web page corresponds to the weighted sum of input links[5].

2.2. Path-ascending crawling

It is expected that the crawler to download as many resources as possible from a particular Web site. In that way a crawler would ascend to every path in each URL (Uniform Resource Locator) that it intends to crawl. For example, when given a seed URL of http://xyz.org/a/b/page.html, it will attempt to crawl /xyz.org/, /a/, /b/ and /page.html.

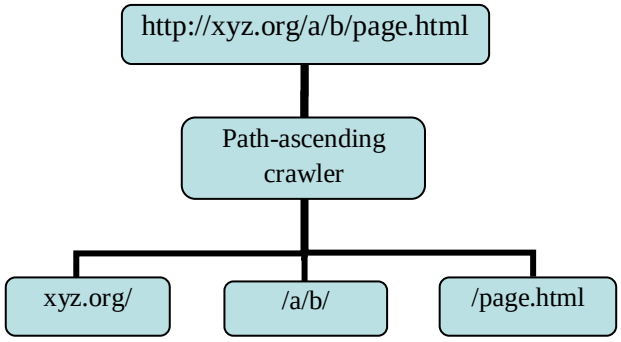

Fig. 2. Path - Ascending crawler

The advantage with Path-ascending crawler is that they are very effective in finding isolated resources, or resources for which no inbound link which would have been found in regular crawling[6].

2.3. Focused crawling

The significance of a page for a crawler can also be expressed as a function of the similarity of a page in a given query. In this approach we can intend web crawler to download pages that are similar to each other, thus it would be called focused crawler or topical crawler[7].

The main problem in focused crawling is that in the context of a web crawler, we would like to predict the similarity of the text of a given page to the query before actually downloading the page. This crawler would be used to complete content of the pages which is already visited and infer the similarity between the driving query and the pages that have not been visited yet. The performance of a focused crawling depends mostly on the richness of links in the specific topic being searched, and a focused crawling usually relies on a general Web search engine for providing starting points. Focused crawler can also extracts only the used specified type of files such as: .jpg, .png, .php, .aspx, etc. This type of crawler can be used to have specific type of search engines based on their file types[8].

2.4. Online Page Importance Calculation Algorithm

On-line Page Importance Computation (OPIC) in this method, each page has a cash value that is distributed equally to all output links (initially all pages have the same cash equal to 1/n). This is similar to Page Rank while it is done in one step.

If http://xyz.org has "m" no. of pages in it,

Then each page obtains 1/m cash.

In every state, the crawler will download web pages with higher cashes and cash will be distributed among the pages it points when a page is downloaded. Experiments were done on a synthetic web graph including at most 600,000 nodes with the power law distribution. There is no comparison between OPIC and other crawling strategies. Unfortunately, in this method, each page will be downloaded many times that will increase crawling time[9].

III. PROPOSED METHODOLOGY

The proposed methodology has two main modules such as URL Crawler and Searching modules. User can use search module to obtain results from Semantic KB also user can add URL to crawl and store the data in Semantic KB.

3.1. Architectural Diagram

URL Crawling module will be used to crawl the web pages to extract the meta data; User enters the URL which has to be indexed, then the meta data are stripped from web page which consists of keyword, description and author name (if exists) using Meta Data Extractor process.

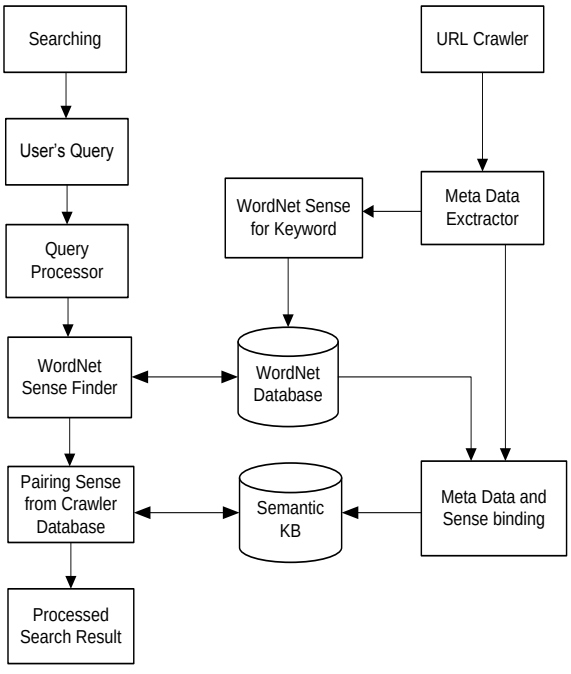

Fig.3. Proposed Architectural Diagram

Description and author name are directly sent to meta data and sense binding module whereas keywords are redirected to a special process where all keyword sense are matched from WordNet Lexical Database that information is again sent to meta data and sense binding there by both keyword sense, description and author name are brought together and stored in Semantic Knowledge Base.

In case of searching module, the query processor takes input query from the user in which Stop Words are removed and Stemming is done for the given query and then the refined query is forwarded to WordNet Sense Finder module from which word sense is obtained for the refined query from WordNet Database and then obtained Sense is directly referred to Semantic Knowledge Base through a relational database query which retrieves most relating results processed to display to the User.

3.2. Algorithmic Expression

URL Crawling and Searching module's working process are expressed in Algorithm which is given below.

3.2.1. URL Crawling Algorithm

In URL Crawling module *url* contains the Uniform Resource Location specified by the user to crawl, Metadata of URL are stored in *k* as keyword and description stored as *d*. Keyword *k* has *m* no. of keywords in it, using a foreach loop *m* no. of keyword's Sense are obtained from WordNet database

***procedure* URLCrawling( )**

url ← URL of webpage

d ← Meta description from ***url***

k ← Meta keyword from ***url***

m ← no. of Keywords in ***k***

foreach m in ***k*** do

sn ←WordNet Sense for ***k*** for ***m***

url, d, k, sn ←Stored in Crawler Database

end URLCrawling

and stored in *sn*. Now, *url*, *d*, *k*, *sn* are finally moved to Semantic Knowledge Base.

3.2.2. Searching Algorithm

In case of Searching Module *sq* Search Query is given by User, query is transferred to Query Processor *qp* in which Stemming and Stop words are polished off from the Query

***procedure searching*( )**

sq ← Search Query

qp ← Processed Query ***sq***

wns ←WordNet Sense for ***qp***

dbl ← Knowledge Base lookup for ***wns***

rs ← Result from ***dbl***

end *searching*

then for the *qp* WordNet Sense is obtained and stored in *wns*, Semantic Knowledge Base lookup is made for *wns* WordNet Sense obtained for the processed query and stored in *dbl* variable, finally *dbl* is processed to display the result to user through *rs*.

3.3. Stop Word

Stop Words are a negative dictionary used in automatic indexing to filter out words that would make poor index terms for a search result. Basically stop words are removed from the search query if it happens to appear. There are over 421 Stop Words [10] it should have maximum efficient and effective in filtering the most frequently occurring and semantically neutral words in general literature in English language. Removing stop word is the initial process of our proposed query processing technique [11].

Sample

S = {**Taj Mahal is very beautiful**}

On applying Stop Words in String S, we get,

S′ = {Taj Mahal beautiful}

3.4. Stemming

Stemming is the process of retrieving the present form of a word from its origin in search query to process and filter the query so that it could be more effective and expressive. In our proposed methodology, Stemming is carried out in query processing section of the search engine [12].

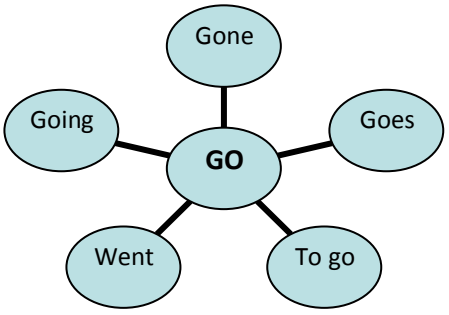

Fig.3. Stemming words for "GO"

Word "GO" can be expressed as going, gone, goes, went, gone away, to go, etc can be reduced to GO using Stemming Algorithm[13].

IV. PERFORMANCE ANALYSIS

The proposed system's performance is calculated through a 2 step process which is Database Query retrieval and through a precision and recall graph.

4.1. Query Analysis

Efficiency of data retrieval from Semantic KB is measured from two types of query equations. Equation (1) states that data from fields url, keyword, description, sense are taken from the Semantic KB where value for sense is like query given by the user is taken

$$\pi_{url,key,desc,sense}\,(\sigma_{sense\ LIKE\ query}(wordwn))$$

Equation (1)

from table name "wordwn", user query is obtained from query processor where the query is refined to search in Semantic KB.

Equation (2) provides the url, description, keyword and sense from the table wordlist where url, description,

$$\pi_{url,key,desc,sense}\,(\sigma_{(url,key,desc,sense)AGAINST(query)}(wordlist))$$

Equation (2)

keyword and sense are matched against Query given by the query processor from the table wordlist.

From equations (1) & (2), following graph is plotted by using five test queries to identify the performance of both equations, where in **X-axis** tends to **Test Cases** and in **Y-axis** time taken to retrieve result from Semantic KB in **MicroSeconds** are marked.

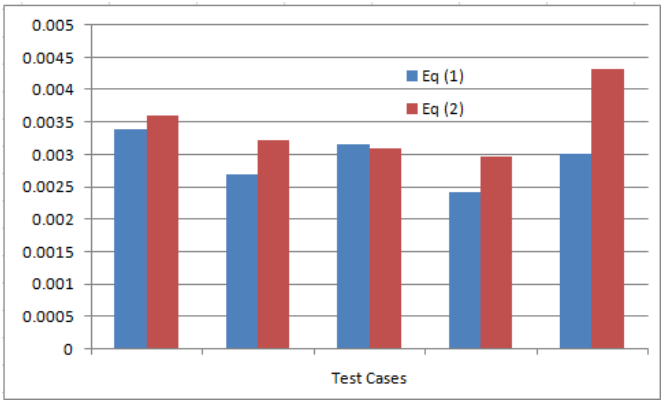

Fig.4. Time taken to retrieve data from semantic KB

From the above Graph we can conclude that **Equation (1)** is more efficient than Equation (2) in terms of fetching the search query from Semantic Knowledge Base (KB).

4.2. Precision and Recall

The search result of the system is analyzed by taking precision and recall graph, for our test case, five query and its search results have been taken to find precision and recall. Query to access the Semantic KB is obtained from the equation (1) from sub section 5.1. In Fig.5 X-axis in the graph represents the precision and recall level with limit of 0 to 0.9 and Y-axis tends to test query taken. Search results obtained by using Test query would have at least one of keyword in its URL, such that the sense for keyword is obtained from WordNet and sense for the query is compared and matching result will be displayed in search result; even results that are pertain to search query are omitted if there is no Keyword for the URL.

From the Fig.5. we can show that few queries has high precision with low recall and for some high precision with high recall but there is no high recall value than precision for any of these query

Let A be the result with the precision and recall

$$A = \{(p1,r1), (p2,r2).....(pn,rn)\}$$

Our proposed methodology shows that the precision and recall of result as

$$\text{Precision of } A = \{p1, p2, p2.....pn\} \geq \text{Recall of } A = \{r1,r2,.....rn)$$

(i.e) From above equation, we can say that always the precision is Greater than or Equal to recall for each test queries.

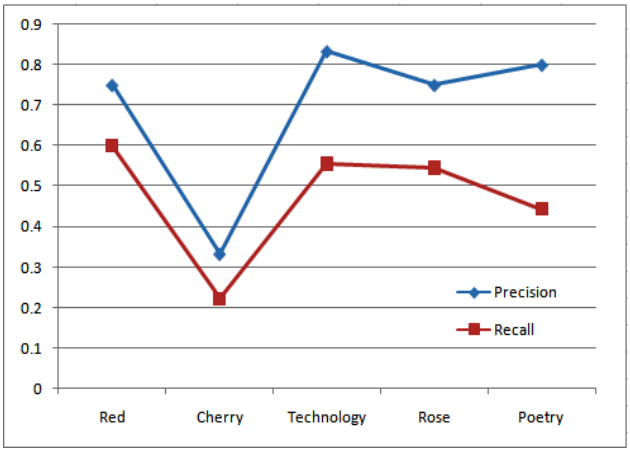

Fig.5. Precision and Recall graph

Thus the results are more precise and having less recall values when the Equation (1) query is used to access the data from Semantic KB. This shows the performance analysis of proposed methodology.

V. CONCLUSION

Searching Mechanism attained major advancement in recent years by using latest searching, indexing algorithms and

advanced query processing techniques. But still there is gap between keyword and semantic based search in all search engines. We proposed a novel semantic based lexical database crawler for more effective search results. It could be still enhanced if we combine other semantic web techniques like domain ontologies, Description Logic and Information Retrieval (IR) related algorithms.

REFERENCES

[1] Miller, G. A. "WordNet: A Lexical Database for English," *Communications of the ACM* (Vol. 38, No. 11), 1995, pp. 39-41.

[2] Fellbaum, C. *WordNet: An Electronic Lexical Database*, Cambridge, MA: MIT, 1998.

[3] Liu, H. & Singh, P. (2004) ConceptNet: A Practical Commonsense Reasoning Toolkit. *BT Technology Journal, To Appear*. Volume 22, forthcoming issue. Kluwer Academic Publishers.

[4] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web* (WWW '07)

[5] Kim, S. J. and Lee, S. H. "An improved computation of the PageRank algorithm" in Proc. of the European Conference on Information Retrieval (ECIR', 2002, pp. 73—85).

[6] Carlos Castillo, Mauricio Marin, A. R. "Scheduling Algorithms for Web Crawling".

[7] Debashis Hati, Biswajit Sahoo, A. K. "Adaptive Focused Crawling Based on Link Analysis," *2nd International Conference on Education Technology and Computer (ICETC)*, 2010.

[8] Mehdi Ravakhah, M. K. "Semantic Similarity Based Focused Crawling" *'First International Conference on Computational Intelligence, Communication Systems and Networks'*, 2009.

[9] Serge Abiteboul, Mihai Preda, G. C. "Adaptive On-Line Page Importance Computation," *Second International Conference on Emerging Trends in Engineering and Technology, ICETET-09* (), 2009.

[10] Fox, C. "A stop list for general text," *SIGIR Forum* (24:1-2), r 90, pp. 19--21.

[11] Ho, T. K. "Stop Word Location and Identification for Adaptive Text Recognition, Int'l," *J. of Document Analysis and Recognition* (:3), 2000, pp. 16--26.

[12] Asuncion Honradot, Ruben Leon, R. O. D. S. "A Word Stemming Algorithm for the Spanish Language,".

[13] Hull, D. A. and Grefenstette, G. "A Detailed Analysis of English Stemming Algorithms", *Technical report, Xerox Research and Technology*, 1996.