*IOSR Journal of Computer Engineering (IOSR-JCE)*
*e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 6, Ver. II (Nov.- Dec. 2017), PP 68-75*
*www.iosrjournals.org*

# Ontology Based Searching For Optimization Used As Advance Technology in Web Crawlers

Pankaj Pratap Singh[1], Palak Agarwal[2]
*[1]Supervisor, Department CSE IIMT Engineering College Meerut, Uttar Pradesh*
*Email:pankajpratapsinghcs@gmail.com*
*[2]M.Tech Scholar Department CSE, IIMT Engineering College Meerut, Uttar Pradesh*
*Email:palakagarwal1508@gmail.com*

***Abstract:*** *As Web is massive, much less coherent, changes more rapidly, and is spread over geographically distributed computers. This requires new information retrieval techniques, or extensions to the old ones, to deal with the gathering of the information, to make index structures scalable and efficiently updateable, and to improve the discriminating ability of search engines. A crawler solves the Resource Discovery Problem in the context of WWW by retrieving information from remote sites using standard web protocols. The importance of a page for a crawler can also be expressed as a function of the similarity of a page to a given query. Web crawlers that attempt to download pages that are similar to each other are called focused crawler or topical crawlers.*

Date of Submission: 13-11-2017 Date of acceptance: 30-11-2017

## I. Introduction

### 1.1 Search Engine

A search engine is an information retrieval system designed to help to minimize the time required to find information over the vast Web of hyperlinked documents. It provides a user interface that enables the users to specify criteria about an item of interest and searches the same from locally maintained databases. In the case of text search engines, the search query is typically expressed as a set of words that identify the desired concept that one or more documents may contain. Some text search engines require users to enter two or three words separated by white spaces for the search of required information contained in text documents, pictures files, sounds files etc.

### 1.2 ELEMENTS OF A WEB SEARCH ENGINE

The various elements of a Web search engine consists of following main components:

- Crawler Module: As compared to traditional document collections which reside in physical warehouses such as the college's library, the information available on WWW is distributed over the Internet. In fact, this huge repository is growing rapidly without any geographical constraints. Therefore, a component used crawler is employed by the search engine which visits the Web pages, collect them and categorize them.
- Page Repository: The downloaded Web pages are temporarily stored in a local storage of search engine, called page repository. The new pages remain in the repository until they are sent to the indexing module, where their vital information is used to create a compressed version of the page.
- Indexing Module:The indexing module takes each new uncompressed page from the page repository extracting suitable descriptors, creating a compressed description of the page. The compressed version of the page is stored in the database, accessible through appropriate interface. Thus, the indexing module is like a black box that takes the uncompressed page as input and outputs a compressed version of the page.

### 1.3 TYPE OF DATA RETRIEVED BY SEARCH ENGINE

- Distributed data: Data is distributed widely over the WWW It is located at Different sites and platforms. The communication links between computers vary widely. Also, there is no topology for data organization.
- High percentage of volatile data: Documents can be added or removed easily in the World Wide Web. These Changes to the documents are usually unnoticed by users.
- Large volume: The growth of data over the WWW is exponential. It poses scaling issues that are difficult to cope with.
- Unstructured and redundant data: The Web is not exactly a distributed hypertext. It is impossible to organize and add consistency to the data and the hyperlinks. Web pages are not well structured. Semantic redundancy can increase traffic.

- Quality of data: A lot of Web pages do not involve any editorial process. That means data can be false, inaccurate, outdated, or poorly written.
- Heterogeneous data: Data on the Web are heterogeneous. They are written in different formats, media types, and natural languages.
- Dynamic data: The content of Web document changes dynamically. The content can be changed by a program such as hit counter that keep tracks of number of hits.

## 1.4 THE CRAWLER

- As compared to traditional document collections which reside in physical warehouses such as the college's library, the information available on WWW is distributed over the Internet. In fact, this huge repository is growing rapidly without any geographical constraints. Therefore, a component called crawler is employed by the search engine which visits the Web pages, collects them and categorizes them. The crawler retrieves web pages commonly for use by a search engine. It traverses the web by downloading the documents and following embedded links from page to page. Though, crawlers are mainly used by web search engines to gather data for indexing, other possible applications include page validation, visualization, update notification, mirroring and personal web assistants / agents etc. Formally, crawlers may be defined as *"Software programs that traverse the World Wide Web information space by following the hypertext links extracted from hypertext documents".*

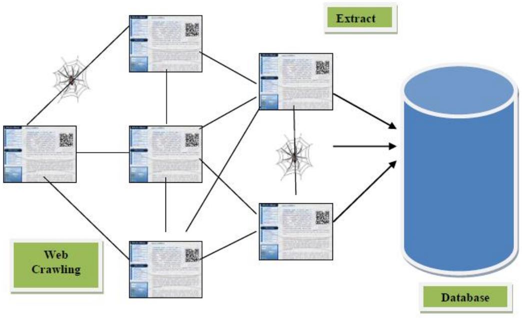


***Figure 1****: Web-Crawler*

### 1.4.1 WEB CRAWLER ISSUES

(a) Input: Number of starting (seed) URLs and (in the case of focused crawlers) the topic descriptions are inputted into crawlers. It can be the description of a list of keywords for classic and semantic focused crawlers.
(b) Page downloading: Extracted pages of the downloaded links are placed in a queue.Queue entries are reordered in a focused crawler by applying content relevance or importance criteria or links may be excluded for further expansion(generic crawlers may also apply importance criteria to determine pages that are worth crawling and indexing).
(c) Content processing: Downloaded pages are lexically analyzed and reduced into term vectors. According to VSM each term vector is denoted by its term frequency-inverse frequency vector (tf-idf). Here we used precompiled idf weights, provided by the IntelliSearch.
(d) Priority assignment: Extracted URLs from downloaded pages are placed in a priority queue where priorities are considered based on the type of crawler and user preferences. It can vary from simple criteria to more involved criteria (e.g. criteria determined by a learning process) i.e. page importance or relevance to query topic (computed by matching the query with page or anchor text)
(e) Expansion: URLs are selected for further expansion and steps (b)–(e) are repeated until some criteria (e.g. the desired number of pages have been downloaded) are satisfied or system resources are exhausted.

### 1.4.2 TYPES OF CRAWLERS

There are various types of crawler available in the market these days, out of which few of the most common one are:

### 1.4.2.1 FOCUSED CRAWLER

The importance of a page for a crawler can also be expressed as a function of the similarity of a page to a given query. Web crawlers that attempt to download pages that are similar to each other are called focused crawler or topical crawlers. The concepts of topical and focused crawling were first introduced by Menczer and by Chakrabarti et al. The main problem in focused crawling is that in the context of a Web crawler, we would like to be able to predict the similarity of the text of a given page to the query before actually downloading the page. A possible predictor is the anchor text of links; this was the approach taken by Pinkerton in the first web crawler of the early days of the Web. Diligenti et al. propose using the complete content of the pages already visited to 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.

There are two main issues regarding the focused crawling discussed as follows:

- The crawlers need to identify from a list of unvisited URLs the ones most likely to contain relevant information.
- The crawlers should avoid irrelevant or bad quality documents by determining the quality and reputation of each document.

### 1.4.2.2 PARALLEL CRAWLER

A parallel crawler is a crawler that runs multiple processes in parallel. The goal is to maximize the download rate while minimizing the overhead from parallelization and to avoid repeated downloads of the same page. To avoid downloading the same page more than once, the crawling system requires a policy for assigning the new URLs discovered during the crawling process, as the same URL can be found by two different crawling processes.

### 1.4.2.3 ADAPTIVE CRAWLER

Adaptive crawler is classified as an incremental type of crawler which will continually crawl the entire web, based on some set of crawling cycles. The adaptive model used would use data from previous cycles to decide which pages should be checked for updates. Adaptive Crawling can also be viewed as an extension of focused crawling.

### 1.5 SEMANTIC NETWORK

A semantic network is a graph of the structure of meaning. Specifically, "It is a graphical notation for representing knowledge in patterns of interconnected nodes and arcs". The nodes represent the concepts and the arcs are the interrelationship between every two nodes. It provides a convenient approach to visualize a knowledge base. Semantic network has been applied for many ontology development projects. It is believed that semantic network is the most appropriate representation method for capturing and encapsulating the massive amounts of semantic information in an intelligent environment.

### 1.6 ONTOLOGY

The term ‘ontology’ is derived from the Greek words “onto”, which means being, and “logia”, which means written or spoken discourse. In computer science, ontology formally represents knowledge as a set of concepts within a domain, and the relationships among those concepts. It can be used to reason about the entities within that domain and may be used to describe the domain. Ontology are the structural frameworks for organizing information and are used in artificial intelligence, semantic web, systems engineering, software engineering, biomedical informatics and information architecture as a form of knowledge representation about the world.

## II. Problem Statement

A critical look on the available literature reveals that the existing work needs to include the following issues:

- There is a need of search engine which cover the two major issues of information retrieval i.e. Polysemy and Synonymy that to simultaneously. Polysemy refers to words with multiple meanings, i.e. how the same phonological form (word) has different semantic mappings (meanings). If the two meanings are unrelated, as in the word pen meaning both writing instrument and enclosure, they are considered homonyms. Synonymy refers to multiple words having the same meaning. As the name implies, synonyms are words that mean the same or have similar meanings in context. Synonyms are used in a variety of situations not only for variety, but to express thoughts or ideas in another, often more emphatic manner.
- To make web searching specific and fast, an appropriate ontology construction plays the most important role as the ontology serves as a starting edge structure for knowledge representation, and the procedure of ontology construction is one of the most critical research topics in the ontology processing.

## III. Proposed Framework

Finding meaningful information among the billions of information resources on the web is a tedious task as the popularity of Internet is growing rapidly. The future of web is a structured semantic web in place of unstructured information present in the web nowadays. On semantic web, ontology is used to assign meaning to the content of the web. Keeping semantic net and ontology in our mind we include following points in our proposed solution:

- A structural approach for the unstructured knowledge over the internet.
- To build a bottom-up ontology model for the existing web.

- To propose a search algorithm in which we exploit our proposed ontology to produce result of a search query in shorter time.
- To propose a solution to the two major problems of information retrieval system (i.e. Polysemy and synonymy), that too simultaneously.
- To build a tree structure for our proposed ontology and better analysis of user search.
- To build a simulation model to prove that our findings have considerable impact on the current searching techniques.

### 3.2 Proposed Solution

Here we are giving the overview of all the three steps involved in our Proposed Solution.

#### 3.2.1 Ontology Construction

As the main objective of our research is to optimize the searching, by making changes in the way the user send his search keywords. Instead of searching in the whole web, our algorithm will search in the ontology built by us that is updated periodically. So before the actual web-searching starts, we should have a web-repository for the development of ontology (Structured knowledge about English word) in parallel.

For building ontology we are using XML (Extensible Markup Language) which is a platform independent plain ASCII text file used as data description language. We have decided to use XML as it could be easily integrated with any of the web development language and it is very easy to use. To build a dynamic XML file, which could be automatically updated we have used C#.net language provided by Microsoft.

#### 3.2.2 Building DOM tree using the Ontology

Based upon the keyword entered by the user, we will create a tree structure using ontology build in step one. For doing so, we will again use C#.net that will retrieve the keyword along with its multiple contexts and its related topics. Thereafter displaying them on a web page in graphical form for making it easier for the user to extract what is desired by the user.

#### 3.2.3 Pruning the Results

The main process on which our basic architecture relies to make the searching more focused and fast is pruning of the semantic network based on the relevance of context given by the user. Based on the relevance the network gets pruned displaying a specific semantic tree based result. For doing this we need a web repository from which result could be extracted, so we have used the web repository of Google, which is considered as the largest and fastest web repository. Using ASP.net we have customized the existing search technique to display more focused i.e. relevant results.

## IV. Pseudo code for Ontology Construction

```
        begin
for each document in web-repository
        begin
provide unique ID to document
        extract keywords
find multiple contexts for keyword using thesaurus
find synonyms, related words & properties for context
        provide keyword, contexts, synonyms & properties to Onto-Builder
        add/update Ontology
end
```

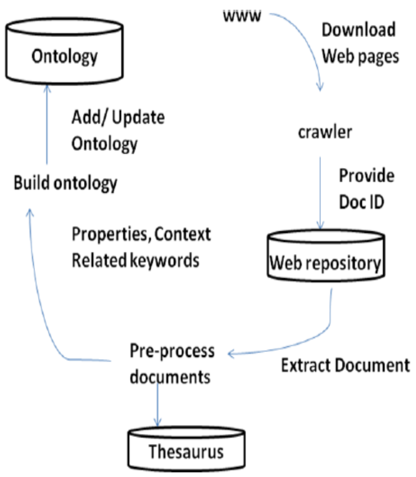


***Figure 2.*** *Committing of Ontology*

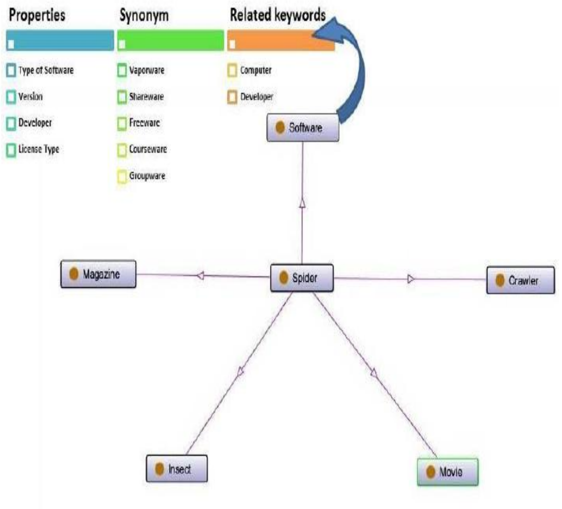


***Figure 3:*** *Diagrammatic representation of ontology stored at backend.*

Proposed architecture of Semantic net based ontology construction
1. Defining ontology in terms of documents. (done in previous chapter)
2. Integrating this ontology with the semantic nets so that a focused document group can be created.
3. Pruning this ontology using the concept of semantic tree for practical usage.
In this idea we proposed a concept of hybrid semantic networks which contains both of the above properties.

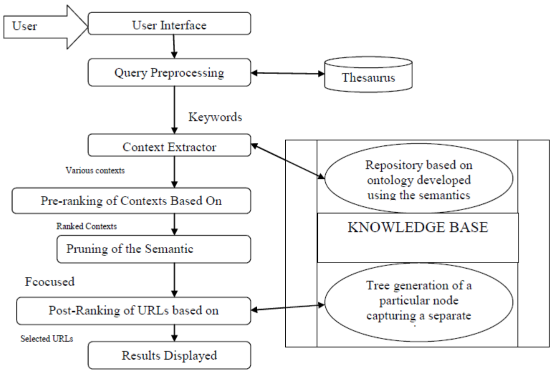


***Figure 4:*** *Proposed architecture of Semantic net based ontology construction*

## V. Implementation Of Semantic Tree

Algorithm to generate tree
Step 1. Retrieve the keyword to search
Step 2. Search the keyword in the Knowledge Base
Step 3. If found then
Step 4 Extract its context
Step 5 Foreach context repeat step 6 and 7
Step 6 Create node and branch
Step 7 Display its context in the form of a semantic net
Step 8 Make each node a hyperlink.
Step 9 Else
Step 10 Display the results in their relevance ratio
Step 11 End'

## VI. Architecture Of Searching

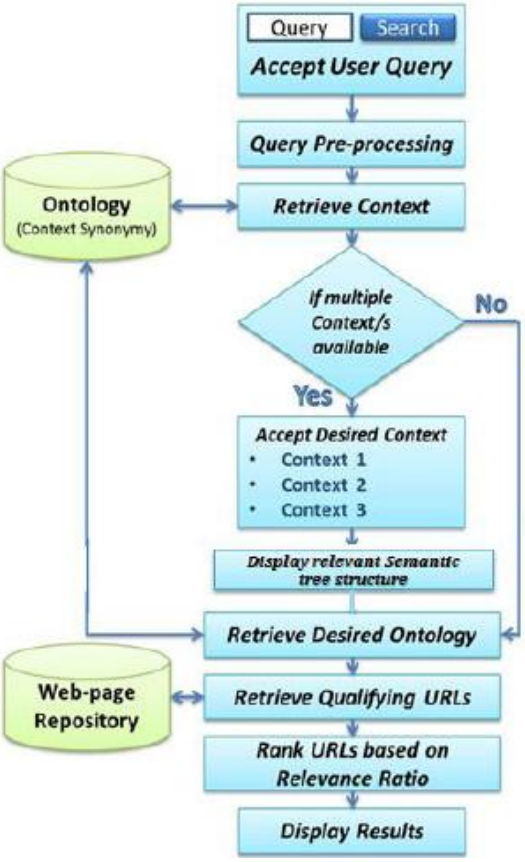
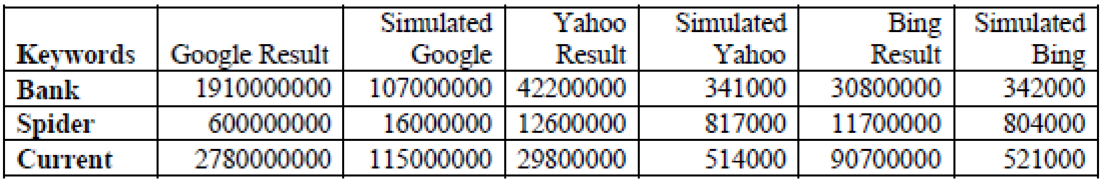


**Figure 5.** Architecture of Searching

## VII. Simulation And Result Analysis

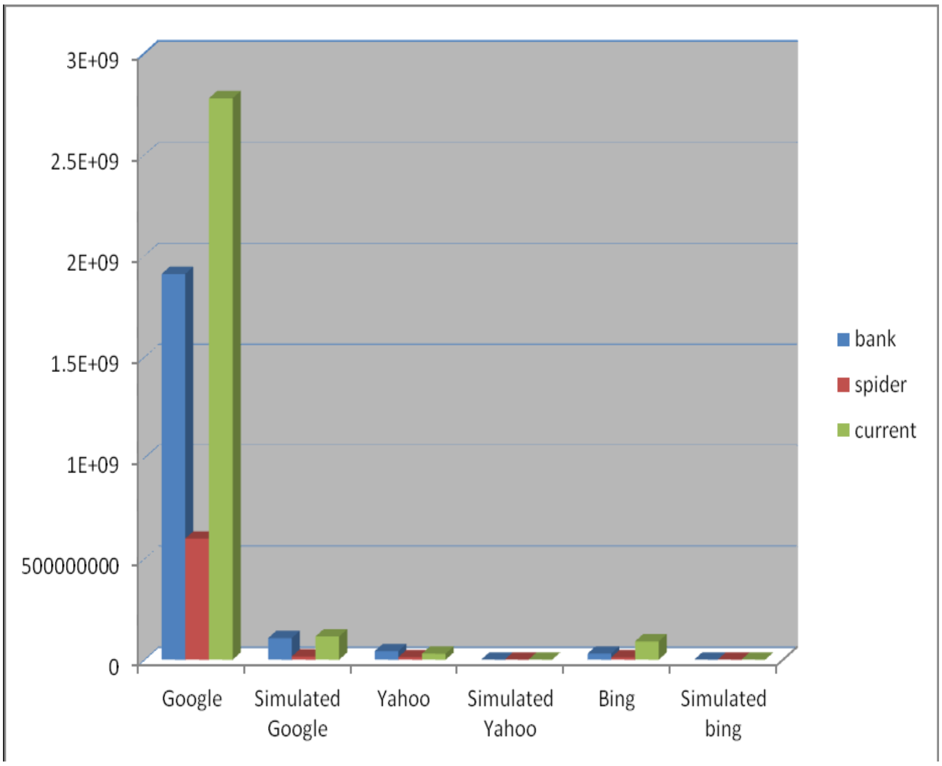
The model “Ontology based-Context synonymy web-searching” is implemented using Microsoft Dot.net. The frontend is designed in ASP.net, the scripting is done in C# and ontology is developed using XML as a Knowledge Base support to the complete system. In our simulation model, we have considered three words having multiple context, various properties and synonyms. Then after implementing our algorithm, we receive a confined search, which is sent to Google for further searching. Same process we did for Yahoo and Bing search engine.

We have considered three words (i.e. Spider, Bank & Bat) initially. All the three words have multiple contexts available. The table below 1 and the bar graph 1 shows the result analysis for these three words (i.e. Spider, Bank & Bat) over three leading search engines (i.e. Google, Yahoo & Bing) and the one of our implementation (Ontology based Context-Synonymy Web-Searching).



| Keywords | Google Result | Simulated Google | Yahoo Result | Simulated Yahoo | Bing Result | Simulated Bing |
|---|---|---|---|---|---|---|
| **Bank** | 1910000000 | 107000000 | 42200000 | 341000 | 30800000 | 342000 |
| **Spider** | 600000000 | 16000000 | 12600000 | 817000 | 11700000 | 804000 |
| **Current** | 2780000000 | 115000000 | 29800000 | 514000 | 90700000 | 521000 |

*__TABLE 1__ RESULT ANALYSIS OF KEYWORDS*



## VIII. Conclusion

Our proposed model helps in the providing the solution to the most critical problem of information retrieval, Synonymy and Polysemy. This study proposes the systematic methodology to develop the ontology in a bottom-up style from engineering documents, called DocOnto (Document-based Ontology). Our methodology is mainly composed of three phases such as defining ontology, integrating the ontology with semantic networks and pruning the ontology for practically usage. This ontology can be updated and generalized using much easier process and is less time consuming and has specific definition of each word in the form of attributes.

Looking into the future perspective of this project, we can extend this research by building the concept of learning (Supervised as well as Un-Supervised) in the semantic networks so that any new word which does not have any entry in the existing ontology can be added. The architecture can be updated so that if user enters any new word (non-existing), it is been recorded by the model in a separate table (un-supervised learning) and whenever the developer of the architecture is looking for updating in ontology that word has been retrieved by the developer and a new entry must be created in the existing ontology related to that word.

In addition to above ideas, we can extend the proposed algorithm by implementing it on parallel crawlers.

Advantages of the Work

- Reduces the number of results extracted.
- Through focused searching irrelevant results are pruned which reduces the time.
- Displaying the multiple contexts and its related topic on a web page in graphical form, making it easier for the user to extract what is desired by the user.
- The advantage of our ontology is that once build, it could be used by any search engine.
- To improve searching performance in terms of precision & relevance.

## References

[1]. A Novel Architecture of Ontology based Semantic Search Engine, International Journal of Science and Technology Volume 1 No. 12, December, 2012

[2]. Thomas R. Gruber, A translation approach to portable ontology specifications, KnowledgeAcquisition 5 (1993), no. 2, 199–220.

[3]. Sajendra Kumar, Ram Kumar Rana, Pawan Singh, “A Semantic Query Transformation Approach Based on Ontology for Search Engine”, International Journal on Computer Science and Engineering (IJCSE), May 2012. (688-693).

[4]. P. Jaganathan, T. Karthikeyan, Highly efficient architecture for scalable focused crawling using incremental parallel web crawler, Journal of Computer Science, 2014

[5]. Eakansh Manglik, Priyanka Sharma, Paramjeet Rawat, Nidhi Tyagi, Ontology based context synonymy web searching, Information systems and Computer Networks, 2013.

[6]. Nidhi Tyagi, Rahul Rishi and R.P. Aggarwal Context based Web Indexing for Storage of Relevant Web Pages, International Journal of Computer Applications, Volume 40 No.3, (Page No. 1-5), 2012.

[7]. S.Thenmalar and T. V. Geetha Concept based Focused Crawling using Ontology, International Journal of Computer Applications, Volume 26 No.7, (Page No. 29 32), 2011.

[8]. Nidhi Tyagi, Rahul Rishi and R.P. Aggarwal, Contextual Ontology: A Storage Tool for Extracting Context from Web Pages, International Journal of Computer Applications, Volume 56 No.7, (Page No. 30 34), 2012.
[9]. Learn XML http://www.w3schools.com/xml.
[10]. www.coursera.com
[11]. Client based architechture http://en.wikipedia.org.
[12]. Google custom search API , http://developers.google.com

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

Pankaj Pratap Singh Ontology Based Searching For Optimization Used As Advance Technology in Web Crawlers.” IOSR Journal of Computer Engineering (IOSR-JCE) , vol. 19, no. 6, 2017, pp. 68-75.