I.J. Information Engineering and Electronic Business, **2019, 5, 40-45**
Published Online September 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2019.05.06

An Efficient and Optimized Sematic Web Enabled Framework (EOSWEF) for Google Search Engine Using Ontology

Vipin Kumar, Arun Kumar Tripathi, Naresh Chandra
Department of Computer Applications, KIET Group of Institutions, Ghaziabad, India
Email: geniusvipin@gmail.com, mailtoaruntripathi@gmail.com, naresh.chandra@kiet.edu

Received: 03 July 2019; Accepted: 27 July 2019; Published: 08 September 2019

***Abstract*—**Remarkable growth in the electronics and communication field provides ubiquitous services. It also permits to save huge amount of documents on web. As a result, it is very difficult to search a specific and desired information over the Internet. Classical search engines were unable to investigate the content on web intelligently. The tradition searching results has a lot of immaterial information along with desired one as per user query. To overcome from stated problem many modifications are done in traditional search engines to make them intelligent. These search engines are able to analyze the stored data and reflects only appropriate contents as per users query. Semantic Web is an emerging and efficient approach to handle the searching queries. It gathers appropriate information from web pool based on logical reasoning. It also incorporates rule-based system. Semantic web reasonably scrutinizes webs contents using ontology. The learning process of ontology not only intelligently analyze the contents on web but also improves scrutinizing process of search engine. The paper suggests a new keyword-based semantic retrieval scheme for google search engines. The schemes accelerates the performance of searching process considerably with the help of domain-specific knowledge extraction process along with inference and rules. For this, in ontology the prefix keywords and its sematic association are pre-stored. The proposed framework accelerates the efficiency of content searching of google search engine without any additional burden of end users.

***Index Terms*—**Semantic web, Ontology, SWOOGLE, EOSWEF, Semantic search.

I. Introduction

Modern data storage technique allows storing of huge amount of internet data storage devices. Internet users upload enormous volume of data over Internet on hourly basis. The warehoused content on web is a pile of unrelated data. The search engines analyzes the contents of web warehouses and return meaningful data i.e. information to the user. Search engines allow to analyses and extract the appropriate contents from huge granary of webpages. Most common techniques for searching of correlated information from web warehouse are full-text-based [1,2] searching and syntax-based searching used by search engines.

Full-text based search technique allows web content searching based on texts and keywords. It matches each individual word specified in the source document and ranks the results algorithmically. In full-text based search, incoming documents are converted into plain text using a document filter, an index module stores a list of essential words from each document into a database, and then the database is optimized for quick lookups without storing the full text of each document. Once the user queries the system using a Web page, the Stop list module deletes words that are not useful for the search. To find more relevant documents, the system will add all possible synonymous terms, which will provide more finely tuned search results.

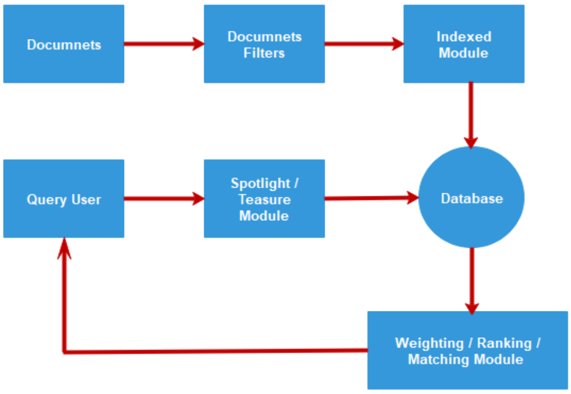
Documnets
Documnets Filters
Indexed Module
Query User
Spotlight / Teasure Module
Database
Weighting / Ranking / Matching Module

Fig.1. Full-text based searching procedure

After the processed query is compared to the stored index, the weighting and ranking factors for each document can be computed across categories to determine the most relevant documents. User search query may contain a single word or a collection of words, phrases etc. The group of words are joined together using basic logical query connectors i.e. AND/OR. When user queries, full text function accesses the optimized word index to identify which documents contain the requested terms. Records that match the query are returned, but records that do not match the query are not returned.

Full-text based searching technique is not efficient and undergoes through several discussed as follows:

- **Synonym Problem:** The synonym problem arises in full-text searching because any thought may be expressed by more than one way such as a person, place, or thing.
- **Harmony Problem:** In full-text based search, when there is more than one meaning for a single word or phrase is known as harmony problem. In other words, in harmony problem, the full-text search fetches numbers of words with several meaning rather than the one that searcher want.
- **False Cognate Problem:** When two or more words in different languages are spelled similarly, but they have different meaning in different languages. This type of problem is known as false cognate problem.
- **Spamming Problem:** When a search query results additional text along with text searched is known as spamming problem. It is also called as "keyword stuffing".
- **Variant Spelling Problem:** When two or more words have same meaning but different spelling is known as variant spelling problem. For example, color is spelled in American English while colour in British English but have same meaning.

II. Literature Review

Semantic web based search provides a better solution to handle above challenges. In sematic web, the data is organized and retrieved in more readable and understandable manner. W3C suggested a new model known as Resource Description Framework (RDF) [6] for semantic web. In sematic web, the knowledge is represented in the form of rich conceptual schemas called ontologies [7]. In other words, we can say Ontologies are the backbone of the Semantic Web. Ontologies are rich conceptual schemas that give formally defined meanings to the terms used in annotations, transforming them into semantic annotations. In the subsequent section, the paper discusses popular semantic web [3,4,5] based search engines.

2.1 Swoogle Search Engine Architecture

Swoogle (Semantic Web based google) [8] is one of the semantic web ontology based search engine. It is crawler-based indexing and retrieval system. The crawlers is used to discover RDF documents and HTML documents with embedded RDF content [4].

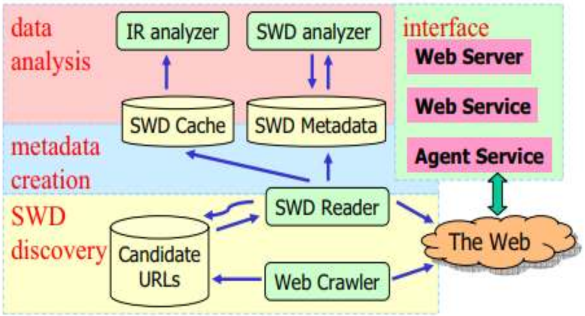

Fig.2. Swoogle's Architecture

Figure 2 shows basic architecture of Swoogle's. It is divided into four key modules named as SWD discovery, metadata creation, data analysis, and interface. The description and working of each components are as follows:

- The SWD discovery phase learns auspicious semantic web databases continuously from Web and keeps latest information about semantic web databases from candidate URLs.
- The metadata Phase is used to cache the snapshot of a semantic web [3,4,5] database and produces objective metadata about semantic web databases for creation of syntax and the semantic.
- The cached semantic web databases and newly created metadata are used by data analysis module to develop analytical reports, for example classification of semantic web ontologies and semantic web databases, rank-of semantic web ontologies, and the IR index of semantic web databases.
- The interface component is used to provide basic data services to Semantic Web community such as agents and humans. Swoogle suffers from poor indexing of documents and high latency in query.

2.2 Semantic Web Search Engine (SWSE) Architecture

Semantic Web Search Engine (SWSE)[9] is another popular semantic based search engine. It is developed by Digital Enterprise research Institute and includes crawling, indexing process and an interface for search to retrieve an information in traditional search engine. SWSE also works on RDF Web data. SWSE works on structured data. It returns data representing the real world entity instead of returning the link of documents. Figure 3 shows the architecture of SWSE and its components. The description and working of each components are as follows:

- The crawl module accepts a seed of URIs (Uniform Recourse Identifier) and generates number of clusters containing RDF documents from web database.

- ➢ Consolidation module searches the synonymous of the identifiers in the data and combines the data according to synonymous retrieved.
- ➢ The ranking components analyze the individual element in the data and assigns a rank/score representing the importance of each data.
- ➢ The reasoning module generates new data, which is implied by the inherent semantics of the input data.
- ➢ The indexing module generates a sequence for information retrieval by the user interface.

Subsequently, the query processing and user interface provide the services to the query over the index built in pervious step. Figure 3 shows semantic based SWSE architecture.

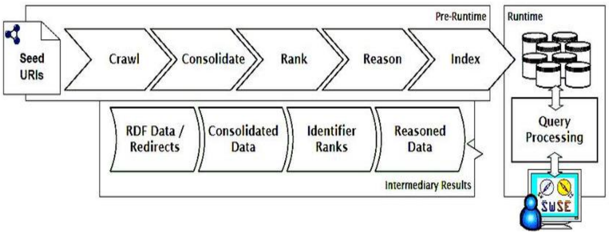

Fig.3. SWSE Architecture

Although, SWSE has similar architecture as in traditional search engines which contains crawling, ranking and ranking of data. On the other hand, to improve the efficiency it includes consolidation and reasoning module to handle RDF documents.

SWSE also suffers from limitations for example poor ranking of documents. It is due to preference of ranking stage before indexing stage.

III. Google Search Engine Architecture

Figure 4 shows the high-level architecture of Google along with its working system. C or C++ can be used as a platform to develop it. For development of its architecture the Linux or Solaris can be used as an operating system. Figure 4 shows high level architecture for Goggle search engine [10].

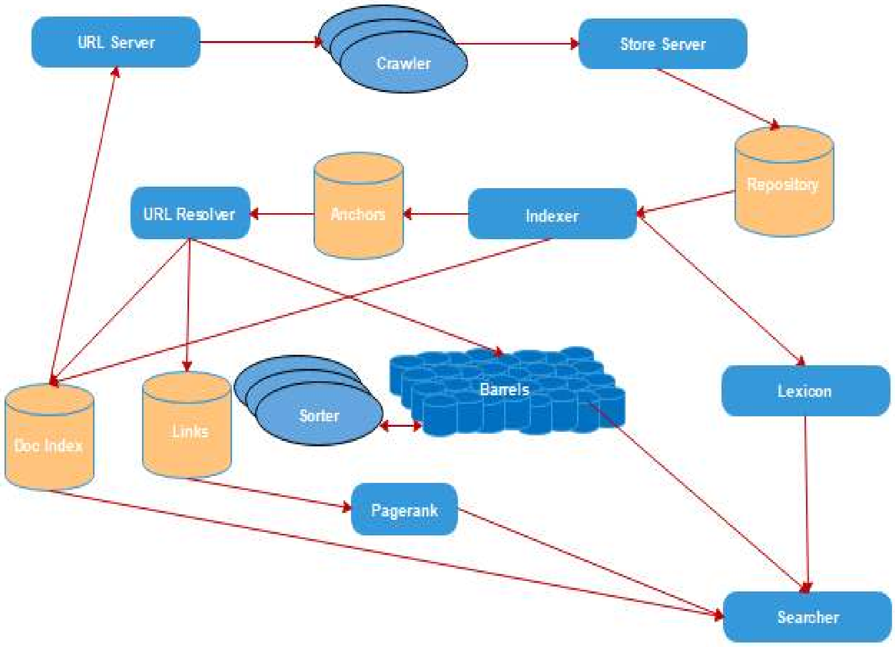

Fig.4. Google Search Engine Architecture

The working of Google search engine is explained as follows:

Step1: The web crawlers are responsible for downloading the web pages. For this, URL server sends lists of URLs to the crawlers.

Step2: Downloaded web pages by crawler are sent to the Store server.

Step3: Store server is responsible for compressing the fetched web pages and stores then at Repository.

Step4: Each fetched web pages is associated with a unique id known as docID.

Step5: The Indexer and sorter combines the indexing function and fetches the data from repository, uncompressed it and parses the document.

Step6: The documents are transformed into hits. Hit implies the set of words occurrences and after that it sorts all the links.

Step7: The indexer then distributes these hits into various set of barrels. The indexer parses all the links in web pages and stores important information in an anchor. The anchors contain information concerning to the links.

Step8: URL resolver fetches documents from anchor files and translates the relative-URLs into absolute-URLs, and forwards the absolute-URLs into docIDs.

Step9: URL resolver is also responsible for creating links database. The link database is responsible for calculating and assigning the page rank for all documents

Step10: Sorter pulls the data from barrels, resorts them by Word ID and off sets into an inverted index.

Step11: Dump Lexicon works the list with Lexicon and produces new lexicon to be used by searcher. The searcher is run by a web server and uses the lexicon built by Dump Lexicon together with the inverted index and the Page Ranks [11,12] to answer queries.

IV. EOSWEF FOR GSE FRAMEWORK

This paper proposes an efficient and optimized sematic web enabled using ontology framework for Google search engine .EOSWEF is designed in a modular fashion. Figure 5 represents the interconnection of various modules. The framework is designed to optimize the Google search engine. To achieve this, we have incorporated predefined semantic rules and logic in the google search engine. Along with this, we have also incorporated ontology as semantic knowledgebase. The framework is implemented using JSP and JENA. JENA is a free and open source Java framework. It is used to building of semantic web and linked data applications. The main function of JENA includes RDF API, query language, reasoning subsystem, persistent storage memory, ontology subsystem, and provides the appropriate interface. The implementation of EOSWEF is divided into number muddles and sub-modules. Each module, its implementation and working are discussed as follows:

- **User:** The user is an actor, who interacts with the EOSWEF using a web interface or mobile application. The user enters a query with interface or application and return the result back on same.
- **Semantic Application Module:** The graphical user interface (GUI application module takes input from interface and checks for semantics of input data. In the proposed framework the Graphical User Interface (GUI) application is designed with the help of JSP and JENA.
- **Keyword Parser Module:** The keyword parser takes input from pervious phase and divides it into number of smallest units as possible known as keywords or tokens. The keywords are categorized into two parts i.e. redefined prefix and user entered searching keywords. The predefined keywords have special meaning and also stored in ontology manager. A list of prefix keywords are listed in Table 1. It can be understand by following searching example:

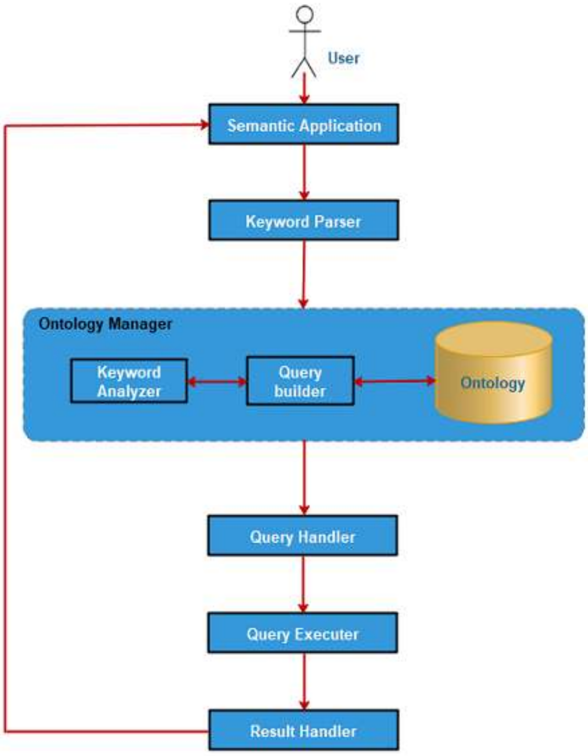

Fig.5. EOSWEF Architecture

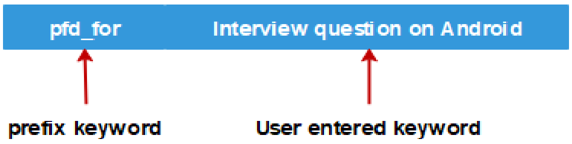

Fig.6. Syntax for prefix keyword and user keywords

The keyword parser segregates the prefix keywords and user entered searching keywords. During paring keyword parser module removes all field delimiters and includes + sign between each user entered keywords except proposition words .For example:

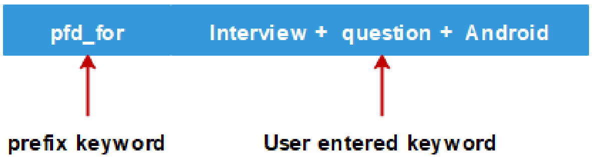

Fig.7. Example of prefix keyword and user keywords

- **Query Executor Module:** The basic task of query executor module is to execute the final semantic query using of google search engine in backend and forwards the generated results to the next module.
- **Result Handler Module:** It is responsible to convert the result generated by query handler module into user’s requested format and send back to the application interface to display the user.

Table 1. Prefix keywords and list of file to search

S. No.	Prefix Keywords	File Types
1	text_for	For Text files
2	video_for	For Video files (MP4,AVI, etc.)
3	music_for	For Music files (MP3, WAV etc).
4	movie_for	For Movie files (MP4, MKV etc.)
5	image_for	For Image files (JPEG, BMP, etc.)
6	pdf_for	For PDF
7	paper_for	For Research Paper

V. Algorithm of Semantic Search

In keyword-based search, quality and efficiency of search result is serious issue. Most of the time in in traditional search the related or relevant pages are not properly organized. Moreover, keyword-based searching techniques are focuses on spelling of the word not on meaning of the word. Another problem in keyword-based searching techniques is that it do not automatically extract meaning from the relevant results of a query. This happens because initially web was designed for direct interaction of human beings; and it does not support machine-readable semantic annotations. The paper focuses on the first of all identification of prefix keyword and association of user entered keywords with prefix keywords.

The Semantic based search using ontology algorithm is discussed as follows:

Input: Search Query of User.
Output: Retrieved Semantic Search Information
Procedure:

1. User enter query in Sematic Application
2. Keyword Parser tokenized searching query keywords to number of terms.
3. Keyword Parser also remove the stop words from the query.
4. Keyword Parser also stem the word.
5. Finally, Keyword Parser get POS (Part of Speech) of the word from the query.
6. Keyword Analyzer expand the words by the hypernym and hyponym concepts in the WordNet.
7. Expand the words by the Ontology Manager (OM) as:
    a. Search for existence the word in the Ontology.
        i. If word is, found, in ontology then get the Hyponym, and send the word to Query Builder along with prefix keyword.
        ii. If word is, not found, in ontology get the Hyponyms Hypernyms, and add word to Ontology and send the word to Query Builder along with prefix keyword.
8. Send final query to Query Handler.
9. Query Handler semantically computes it and forwards it to Query Executor.
10. Query Executor finalized the original query and execute it.
11. Result Handler formats the obtained results and pass it to Semantic Application
12. At the last, Semantic Application display result to User.

VI. Result Analysis

In the analysis, we explored the hidden features of Google search engine and optimized them with the help of semantic knowledge with the help of ontology.

Table 2. Searching Comparisons

S. No.	Google Search (File Type)	Proposed Semantic Search (% for first ten pages of search)	Google Search (% for first ten pages of search)
1	text	80	91
2	Pdf	72	81
3	Image	40	45
4	Video	66	71
5	Music	80	82
6	Movie	35	37
7	Research Paper	33	34

The result analysis is one of the critical phase. The experiment is performed on Core-i5 machine with 4 GB RAM, 1TB hard disk having window10 operating system. The software specification includes JENA framework. The experiment is performed for 100 results. In general, Google ten results per page, hence we have considered first ten pages for comparison. Table 2 shows the comparison between Google search and proposed semantic search.

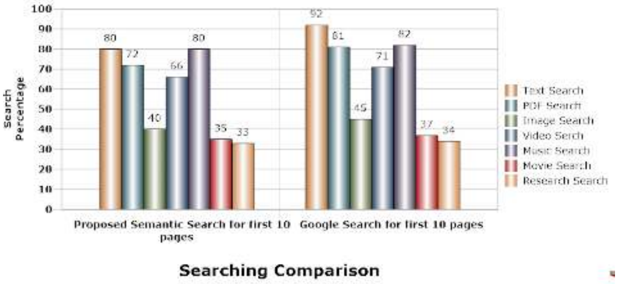

Fig.8. Comparative result analysis between Google search and proposed semantic search frameworks

VII. Conclusion

Extensive growth in the field of mobile and communication industry allows to access Internet from anywhere at any time. People can search the information of their interest from huge data warehouses. Initially paper discuss existing full-text based searching techniques and their drawbacks. Later on, existing semantic based searching techniques are discussed. These searching techniques are implemented in various search engines. Google is one of the most popular search engine. This paper proposes a new framework for google search engine. The paper enhances the searching capabilities of google search engine with the help of semantic web using ontology. The execution of google and proposed framework are compared. The results shows the proposed framework is more efficient and optimize than the existing framework of google. EOSWEF uses the hidden features of google search engine. Furthermore, artificial intelligence feature can added into it to make it more efficient.

REFERENCES

[1] J. Beal, “Weaknesses of Full text search”, The Journal of Academic Librarianship, vol. 34, Number 5, pp. 438-444, 2008.

[2] J. Beal,, and Technical Services”, vol. 34, Issues 2–3, pp. 74-82, 2010.

[3] Tim Finin , James Mayfield , Anupam Joshi , R. Scott Cost and Clay Fink, “Information Retrieval and the Semantic Web”, IEEE 8th Annual Hawaii International Conference on System Sciences, pp. 1-10, 2005.

[4] Liyang Yu, “Introduction to the Semantic Web and Semantic Web Services”, First Eition, Chapman and Hall/CRC, 2007.

[5] T. Berners-Lee, “Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web by its Inventor”, Harper: San Francisco, 1999.

[6] L. Chang, W. Haofen, Y. Yong and X. Linhao, “Towards Efficient SPARQL Query Processing on RDF Data”, Tsinghua Science and Technolgy, vol. 15, Issue 6, pp. 613-622, 2013.

[7] Seema Redekar, Vishal Chekkala, Siddhapa Gouda and Swapnil Yalgude, “Web Search Engine Using Ontology Learning” IJIRCCE, vol. 5, Issue 3, 5092-5097, 2017.

[8] Tim Finin, Yun Peng, R. Scott, Cost Joel, ”Swoogle: A Search and Metadata Engine for the Semantic Web", University of Maryland Baltimore County, pp. 652-659, 2011.

[9] Aidan Hogan and Andreas Harth and Jürgen Umrich, Sheila Kinsella, Axel Polleres and Stefan Decker, "Searching and Browsing Linked Data with SWSE: The Semantic Web Search Engine",Journal of Web Semantics, vol. 9, Issue 4, pp:1-55, 2011.

[10] L.A. Barroso, J. Dean, U. Holzle, “Web search for a planet: The Google cluster architecture”, IEEE Micro vol. 23 Issue. 2, pp. 22-28, 2003.

[11] M.M.EI-gayar, N.Mekky and A.Atwan: “Efficient proposed framework for semantic search engine using new semantic ranking algorithm” in IJACSA, vol 6, no. 8, pp. 136-143, 2015.

[12] M. P. Selvan, C. A. Sekar and P. A. Dharshini, “Survey on Web Page Ranking Algorithms”, International Journal of Computer Applications, vol. 41, No. 19, Published by Foundation of Computer Science, pp. 1-7, 2012.

Authors’ Profiles

Dr. Vipin Kumaris an Assistant professor & Assistant Dean (Skill Development) in KIET Group of Institutions, Ghaziabad, UP, India. He completed his Ph.D. (CS) in 2016 at NIMS University, Jaipur, India. His research interests include Semantic Web, Networking, & Block Chain.

Dr. Arun Kumar Tripathi received the B.Sc. (Electronics) degree from Dr. Hari Gour University Sagar and M. Tech. Dr. APJ Abdul Kalam Technical University in Computer Science and Engineering and completed Ph.D. from National Institute of Technology, Kurukshetra in the field of security in PMIPv6. He joined the KIET group of Institution, Ghaziabad in 2003 and presently working as Associate Professor. His area of interest is Mobile and Wireless Communication. He has published 38 papers in various International National conferences and Journals.

Naresh Chandra is working as an Assistant professor in KIET Group of Institutions, Ghaziabad, UP, India from 2008. He completed his MCA from M.M.M. Engineering College, Gorakhpur and M. Tech. (CSE) from Shri Venkateshwara University, Gajraula. (UP) and pursuing Ph.D. from Jaypee Institute of Information Technology, Noida. His areas of interests are Mobile Computing, Web Technology and Blockchain.

How to cite this paper: Vipin Kumar, Arun Kumar Tripathi, Naresh Chandra, "An Efficient and Optimized Sematic Web Enabled Framework (EOSWEF) for Google Search Engine Using Ontology", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.11, No.5, pp. 40-45, 2019. DOI: 10.5815/ijieeb.2019.05.06