ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)
Vol. 2, Special Issue 10, March 2016

MULTI-WORD CRAWLER HARVESTING FOR DWI

NANDHINI.G[*1], MURALI.D[#2], KUMARESAN.A[#3]
[*1] PG Scholar, Department of Computer Science and Engineering,
[#2]Assistant Professor, Department of Computer Science and Engineering,
[#3] PhD Scholar, Department of Computer Science and Engineering,
S.K.P Engineering College, Thiruvannamalai, Tamil Nadu, India.
nandhinig27@gmail.com

ABSTRACT

As web grows advance, where users can request and exchange information with others. The web services used to search the relevant details due to more number of web resources searching multi-key word by efficiently harvesting deep web interfaces is the biggest issue. In proposed system, Multi-Word Crawler is retrieving the web pages with relevant data using two search concepts, multi-key words ranked search and the synonym based search. The system provides all the possible relevant links. This process will be achieved in two ways. In the first way, the query is submitted to the application will be preprocessed, after preprocessing root words will be taken and it finds the synonym, hypernym and hyponym then it is listed to the user, with the help of query searching all the possible links can be found related to the search. In the second way, Multi-word Crawler achieves the output. If any words are selected in the displayed list, then all the web site links, images and news feeds will be provided as final output to the user. The additional feature in the Multi-word Crawler is after achieving the output, the bookmark concept is included. The bookmarked link will be added to the application directly not to the browser, and then the bookmarked content will be globally visible.

***Index terms*: Multi-word Crawler, Synonym, Hypernym, Hyponym.**

1 INTRODUCTION

The web services are increasing and more popular in nowadays. It is difficult to find the data from deep web. The main objective of this project is to search the multi word data from deep web interface. It uses clustering for their process. The data are provided in the form of synonym, hypernym and hyponym to the user. This method provides the similar meaning, general meaning and specific meaning to the user searched data. The existing strategy is not meeting these requirements, it just classifies the data into two stages, and they are sites locating and in sites exploring [1]. In this project Text clustering is used to cluster the data according to the user input term. Firstly the structure preprocesses the set of papers and the user given terms. The high-dimensional text vectors dimensionality is reduce by using the appraisal of character. The purpose of this project is to cluster the text entry based on the user typed input, to develop the correct web search (ontology) and overcome the grouping of unrelated documents into the collect matching documents. It also aims to help web users position the best explore tools for their search wants, ensuring in earlier and more exact search results. The OTMM model is used with statistical method and also optimization models and it contains the ontology reference. The new techniques in each process are clustering by self-organized mapping (SOM) algorithm. SOM algorithm is a neural network model which has typical unsupervised learning and it clusters the input data using similarities. The Multi-Word Crawler approach is both effective and efficient.

2 RELATED WORK

The deep web consists of large volume of information; the previously proposed work has more number of tools and techniques. Some of the steps related to this work are given.

2.1 Generic Crawlers:

The Generic Crawlers are collecting the all searchable data that are related to input. It is not focused on the particular input given [10], [11], [12], [13], [14].

2.2 Focused Crawlers:

The Focused Crawlers selecting the links that lead to the document and it avoids the non-topic area. It consists of two types, FFC [4] and ACHE [8]. The Form-Focused Crawler (FFC) searches only the specific data according to user input [4] and the Adaptive Crawler for Hidden-web Entries is same as FFC, it also searching only the specific data [8]. The FFC searches the data using the three components such as form classifier, page and then link. ACHE is extending from the FFC with two more additional components such as adaptive link ranker and form filtering.

2.3 Smart Crawler:

Smart Crawler is a particular domain crawler [1]. It locates the relevant deep web content. Smart Crawler is an effective deep web harvesting framework that achieves the high efficiency and broad coverage for a Focused Crawler. Smart Crawler is divided the stages into two. The first stage is Site-locating and then the second stage is In-site exploring [1]. The site locating stage, performs the searching for center pages using site based technique with the help of search engines, it avoid visiting a more pages. Smart Crawler prioritizes the highly relevant data according to the topic by rank the websites. The in-site exploring, extracts the more relevant links by achieving the in-site searching fastly. The site-locating helps to achieve broad coverage of sites for the crawler and the in-site exploring efficiently perform searching for the data within the site. The link tree data structure is designed to achieve broad coverage to eliminate one sided visiting of relevant links from the hidden web. It first ranks the sites and then prioritizes the links within the site [1].

2.4 Web Crawler:

Web Crawler is a script of program or software, it browses the web into two manners they are systematic manner and automatic. The web structure is graphical. The architecture of web crawler consists of three main components: frontier, and then page downloader and last component are web repository. The frontier stores the URL’s list related to topic to visit the data [17]; the page downloader is used to download the web pages and the web repository receives the web pages from crawler and stores the web pages in the database [19]. The web crawler is classified into four types namely, Focused Web Crawler, Incremental Crawler, Distributed Crawler and Parallel Crawler.

Focused Web Crawler

The Focused Web Crawler is also known as Topic Crawler. Focused Web Crawler downloads the pages that are related to each other. It collects the documents that are specific and relevant to the user input data [17], [18].

Incremental Crawler

Incremental Crawler is a traditional crawler, the collected data is refreshed using this crawler, and it replaces the old documents continuously with the downloaded new documents [20].

Distributed Crawler

Distributed web crawler is the distributed computing technique. More number of crawlers is working together to distribute in the process of web crawling, to have the more web data by broad coverage [21].

Parallel Crawler

In Parallel Crawler, Multiple crawlers are run in parallel. It consists of C-procs a multiple crawling processes. It also depends on the freshness of page and selection of page [22].

3 DESIGN

3.1 SYSTEM ARCHITECTURE

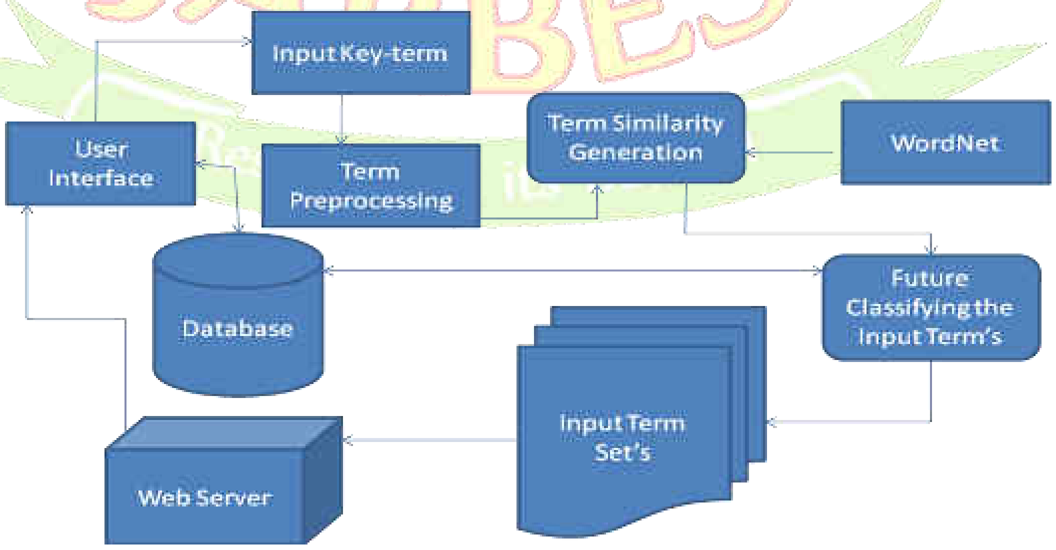


Fig.1: The System Architecture of Multi-Word Crawler.

- The architecture shows that the user interface gets the input from the user. The user classes and characteristics are:

  User: User can enter the search space and they can search the keyword and get the more relevant result based on ontology technique.

  WordNet: This is the tool for ontology clustering process, use this tool for NLP to generate the similar meaning words for user input.

  Deep Search: The system searches the user input data and also the word has similar meaning and provide the relevant data for search result.

- The input key term is preprocessed with term similarity generation using wordnet tool.The input terms are classified with the help of database. The input data sets finally go to the web server after processed. The web server provides the final result to the user.

4 MULTI-WORD CRAWLER SYSTEM

Multi-Word Crawler system proposes a practically flexible and efficient searchable scheme which supports two types of search concept, multi-keyword ranked search and the synonym based search. To address the multi-keyword search with ranking the result, the Vector Space Model (VSM) is used. It builds the document index. Each and every document is expressed as a vector and the dimension value of each document is the Term Frequency (TF) weight of its related keyword. A new vector is also developed in the phase of query. The new vector has also the same dimension with the document index and their each dimension has value denoted as Inverse Document Frequency (IDF) weight. The cosine measure is used to find the similarity of document to the search input. The search efficiency is improved using a binary tree which is balanced with tree-based index structure. The document index vector is used to construct the searchable index tree. The related documents can be easily found by the tree traversing.

4.1 ADVANTAGES:

- It provides the searchable data into three forms, namely, synonym, hypernym and hyponym.
- Easier to find the data from deep web and delivers quickly to the user.
- Bookmarked link is globally visible.

5 IMPLEMENTATION

In the implementation stage the project is turned out from theoretical design to the working system. It is the most critical stage for achieving the new successful system, and provides the confidence to the user that the new system will effectively work. The implementation stage involves planning, investigation of the old system and its details on implementation, and designing of these methods to achieve the changes and evaluating the changeover methods. It is the process of converting a design of new system into operation process. It is the phase that focuses on training for user, preparation of site and conversion of file for developing a new system. In this section the techniques and modules for the proposed approach are explained for implementation.

5.1 STEPS TO BE FOLLOWED

5.1.1 User Interface
- Search space
- User Input

5.1.2 Data Preprocessing
- Stop word Removal

5.1.3 Ontology Clustering
- Multi-term Search

5.1.4 Cluster the Most Relevant Content

5.1.5 Universal Bookmarking

5.1.1 USER INTERFACE

User interface is a collection of techniques and mechanism to interact with something. It is a subset of a field of study called HCI (Human-Computer Interaction). HCI is the process of planning and design that how the user and system work together to satisfy the user needs in an effective way.

- **Search space**
  - ➢ After the user login process, the web user enters into the search page.
  - ➢ It is the place for user to search the content from the web server.
  - ➢ This Search Space provides the interface for user and the system.

- **User Input**
  - ➢ Getting the input query for search process from the user.

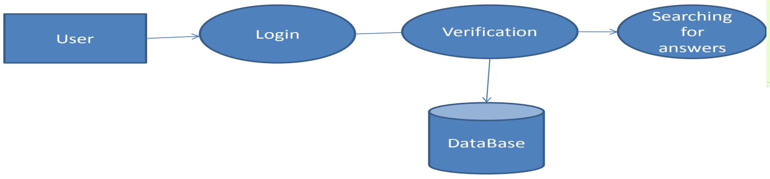


Fig.2: User Interface

5.1.2 DATA PREPROCESSING

- **Stop Word Removal**

Stop words are the one type of words that are filtered out before, or after, processing of data. It is controlled by the input of human and it is not automated. Some of the short function words are *at*, *which*, *on*, *is* and *the*.

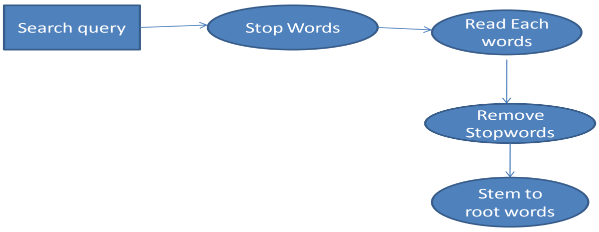


Fig.3: Data Preprocessing

5.1.3 ONTOLOGY CLUSTERING

- **Nym's Group:** Words stopping with **nym's** are used to describe the different classes of words, and their relationships in between the words.
  - **Hypernym:** It provides the general meaning for the word than another.
  - **Hyponym:** It provides the specific meaning for the word than another.
  - **Synonym:** It provides the two or more words have same or similar meaning.
- **Text Analysis:**
  - It analyses the text to find the data from the deep web.
  - The results are provided to the user in the form of sentence and document.
- **Multi-term Search**

  Get the multi-term input from the user and it will search the keyword one by one and get the relevant content from the web servers. The system get the search result deeply from the search engines and its search the terms randomly till last key term in that multi-term list.

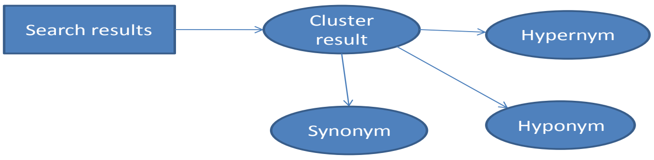


Fig.4: Ontology Clustering

5.1.4 CLUSTER THE MOST RELEVANT CONTENT

Cluster the sentences by keywords. Thus, the keywords that represent the input are identified within the text collection due to clustering. From the multi-term search result, cluster the more relevant content based on the relationship user input term. And classify the cluster and give the final output like most relevant content comes first and outcomes next to the output screen.

5.1.5 UNIVERSAL BOOKMARKING

From the search result user can open any result page link from the list of result, if the user feel any of page contains the best answer of them point of view they can bookmark the page link, system will store the input query and the bookmarked link in the database. In future, if the user going to search the same query input or similar query input in the search after preprocessing and before ontology, system going to check that query or related to that query is available in the bookmark list, if the query is match in bookmark user get the bookmarked page links directly, here going to reduce the time and provide trust content.

6 RESULTS

6.1 OUTCOMES OF STEPS

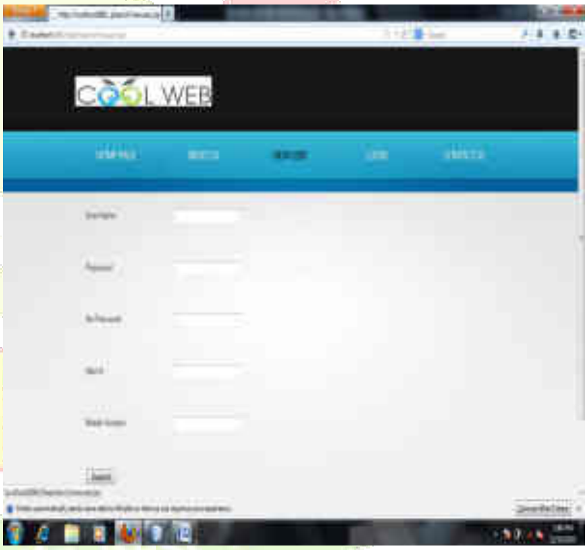
Test	**Requirement** **or Purpose**	**Action / Input**	**Expected Result**	**Actual Result**	**P/F**
1	Validating the user information	Click the login button	Valid user	Same as expected	Pass
2	Searching for keyword	Submit Query	List of Meanings will be displayed	Same as expected	Pass

3	Clustering	Submit Query	Meanings are clustered	Same as expected	Pass
4	Searching for results	Click on the “search button”	Search results will be clustered	Same as expected	Pass

6.2 SCREENSHOTS

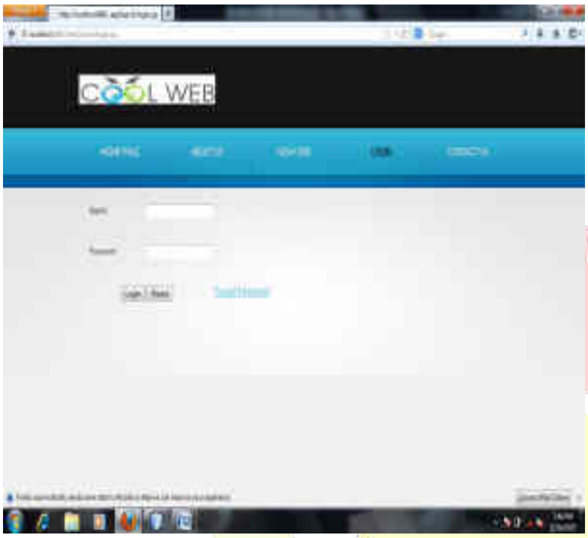
6.2.1 HOMEPAGE

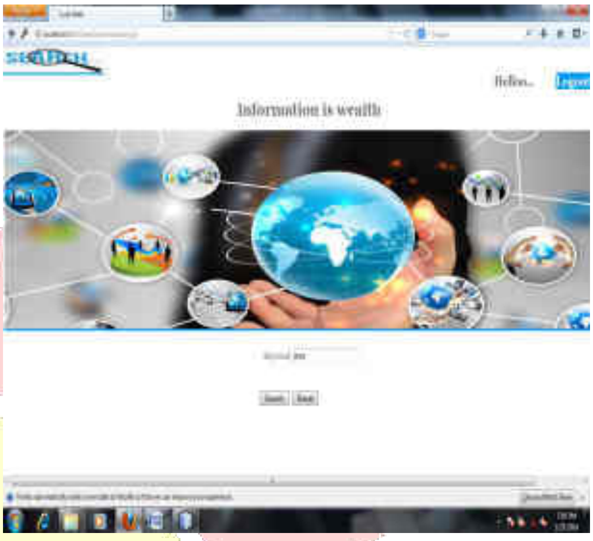
6.2.2 NEW USER



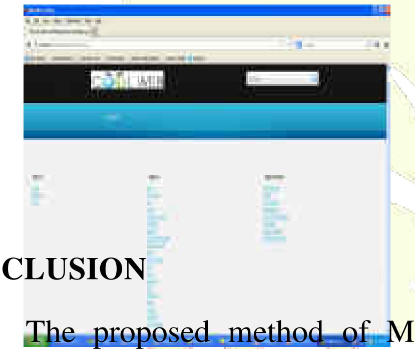
6.2.3 LOGIN

6.2.4 DATA PREPROCESSING

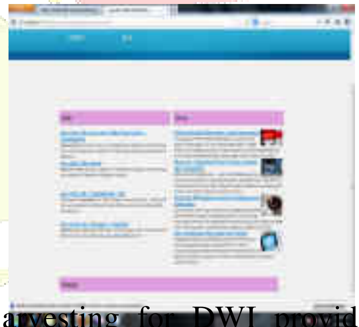




6.2.5 ONTOLOGY CLUSTERING



6.2.6 RELEVANT CONTENT



CONCLUSION

The proposed method of Multi-Word Crawler harvesting for DWI provides the Synonym, Hypernym and Hyponym for the query searched by the user. This method has the benefit that it extracting the data records and providing the alignment options for the data. This method is efficient and effective for clustering the research proposals with English texts. It supports two types of search such as multi-keyword ranked search and synonym based search. The bookmark concept is included that is the bookmarked link will be added to the application so that the bookmarked content will visible globally. The new techniques used in data extraction from deep webs are improved in this system to achieve the efficiency.

REFERENCES

[1] Feng Zhao, Jingyu Zhou, Chang Nie, Heqing Huang, Hai Jin. SmartCrawler: A Two-stage Crawler for Efficiently Harvesting Deep-Web Interfaces. IEEE Transactions on Services Computing Volume: PP Year: 2015.

[2] Cheng Sheng, Nan Zhang, Yufei Tao, and Xin Jin. Optimal algorithms for crawling a hidden database in the web. Proceedings of the VLDB Endowment, 5(11):1112–1123, 2012.

[3] Balakrishnan Raju, Kambhampati Subbarao, and Jha Manishkumar.Assessing relevance and trust of the deep web sources and results based on inter-source agreement. ACM Transactions on the Web, 7(2): Article 11, 1–32, 2013.

[4] Luciano Barbosa and Juliana Freire. Searching for hidden-web databases. In WebDB, pages 1–6, 2005.

[5] Wensheng Wu, Clement Yu, AnHai Doan, and Weiyi Meng. An interactive clustering-based approach to integrating source query interfaces on the deep web. In Proceedings of the 2004 ACM SIGMOD international conference on Management of data, pages 95–106. ACM, 2004.

[6] Panagiotis G Ipeirotis and Luis Gravano. Distributed search over the hidden web: Hierarchical database sampling and selection. In Proceedings of the 28th international conference on Very Large Data Bases, pages 394–405. VLDB Endowment, 2002.

[7] Soumen Chakrabarti, Kunal Punera, and Mallela Subramanyam. Accelerated focused crawling through online relevance feedback. In Proceedings of the 11th international conference on World Wide Web, pages 148–159, 2002.

[8] Luciano Barbosa and Juliana Freire. An adaptive crawler for locating hidden-web entry points. In Proceedings of the 16th international conference on World Wide Web, pages 441–450. ACM, 2007.

[9] Sriram Raghavan and Hector Garcia-Molina. Crawling the hidden web. In Proceedings of the 27th International Conference on Very Large Data Bases, pages 129–138, 2000.

[10] Kevin Chen-Chuan Chang, Bin He, and Zhen Zhang. Toward large scale integration: Building a metaquerier over databases on the web. In CIDR, pages 44–55, 2005.

[11] Denis Shestakov. Databases on the web: national web domain survey. In Proceedings of the 15th Symposium on International Database Engineering & Applications, pages 179–184. ACM, 2011.

[12] Denis Shestakov and Tapio Salakoski. Host-ip clustering technique for deep web characterization. In Proceedings of the 12th International Asia-Pacific Web Conference (APWEB), pages 378–380. IEEE, 2010.

[13] Denis Shestakov and Tapio Salakoski. On estimating the scale of national deep web. In Database and Expert Systems Applications, pages 780–789. Springer, 2007.

[14] Shestakov Denis. On building a search interface discovery system. In Proceedings of the 2nd international conference on Resource discovery, pages 81–93, Lyon France, 2010. Springer.

[15] Jayant Madhavan, David Ko, Łucja Kot, Vignesh Ganapathy, Alex Rasmussen, and Alon Halevy. Google's deep web crawl. Proceedings of the VLDB Endowment, 1(2):1241–1252, 2008.

[16] Olston Christopher and Najork Marc. Web crawling. Foundations and Trends in Information Retrieval, 4(3):175–246, 2010.

[17] Gautam Pant, Padmini Srinivasan, "Learning to Crawl: Comparing Classification Schemes", ACM Transactions on Information Systems, Vol. 23, No. 4, October 2005, Pages 430–462.

[18] Wenxian Wang, Xingshu Chen, Yongbin Zou, Haizhou Wang, Zongkun Dai, "A Focused Crawler Based on Naive Bayes Classifier", Third International Symposium on Intelligent Information Technology and Security Informatics, IEEE Conference Publications,2010

[19] Jun Hirai Sriram Raghavan Hector Garcia-Molina Andreas Paepcke, "WebBase : A repository of web pages" , available: http://ilpubs.stanford.edu:8090/380/1/1999-26.pdf

[20] Junghoo Cho and Hector Garcia-Molina. 2000a. "The evolution of the web and implications for an incremental crawler", In Proceedings of the 26th International Conference on Very Large Databases.

[21] Vladislav Shkapenyuk Torsten Suel "Design and Implementation of a High-Performance Distributed Web Crawler", CIS Department Polytechnic University Brooklyn, NY 11201.

[22] Junghoo Cho, Hector Garcia-Molina, "Parallel Crawlers", WWW2002, May 7–11, 2002, Honolulu, Hawaii, USA, ACM 1-58113-449-5/02/0005.

[23] Balakrishnan Raju and Kambhampati Subbarao. Sourcerank: Relevance and trust assessment for deep web sources based on inter-source agreement. In Proceedings of the 20th international conference on World Wide Web, pages 227–236, 2011.