International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

BARIUM SEARCH USING PYTHON

[1]Boini sri sushmitha
[1]UG, Dept. of Computer science Engineering, Mallareddy Engineering College for Women.

Abstract

With the growing amount of data in recent years, capturing large sets of data to identify the insights and visions of extracting data has become increasingly important. In this project, extracting keywords is one of the most important tasks while working with text data in the domains of text mining, information retrieval, and natural language processing. In order to achieve this, we propose our project, Barium Search, a standalone cross-platform application implemented in Python. The main aim of this project is to enable the system to go through the various files by using the word and prepare the list of files based on the search. Before accessing the data for the users, it will first encrypt the index information and save it to the database for their use. It can handle the information in multiple text file formats. This project will be able to provide users' required information at one particular place by using the word entered by the user during their search operation. This is a relatively new research topic, and many studies remain to be done.

Keywords:- LSR, NLP, Barium Search.

1. INTRODUCTION

The popularity of any search engine is contingent upon two parameters - how quickly it can return results and how relevant the results are. This is true for not just search engines but any scenario where large databases need to be scanned for fetching documents or data. In order to achieve the first parameter of faster retrieval, indexing mechanisms are used. Two types of indexing are possible forward and inverted. In forward indexing, documents are normally stored as a list of words. Whereas, in inverted indexing, the list of documents in which the given word appears is stored. The indexes must be stored in such a way that they can be accessed and retrieved quickly by search algorithms. Currently, the indexes are stored in the look up table in a random fashion without forming a logical sequence. Hence, iterating through the look up table for fetching an inverted index of a given word becomes a time consuming process. For example, in a table , if a search algorithm needs access to the inverted indexes for the word love, it will have to perform a linear search starting from the top of the table until it finds the entry for 'love'. Instead, some mechanism should be used to determine the position in the look up table where the word and its inverted index needs to be stored. Moreover, searching algorithms can apply the same mechanism to directly get the row number of the look up table where the indexes have been stored. In this paper, we propose using hashing as the mechanism for the same. Since tables are basically 2-dimensional arrays, direct access of a particular row is possible. Thus this technique eliminates the overhead of iterating through the look up table i.e. linear search for fetching the desired entry. In summary, we create a (logical) forward index via hashing to store the output of inverted index.

2. RELATED WORK

The existing search mechanisms were able to search only the files that are saved in the system. So there was no mechanism for searching the data inside the files as per the user search query. Most of time, users were not able to get their desired information and this section keeps users at their limit. Users were able to get information only if the file name is known.

Some special mechanism has been used to work on the concept of crawling which can be integrated with this project. As to work under real time situation, it will enable the system, to go through the various files by using the word and prepare the list of files based on the search. Before accessing the data to the users, it will first encrypt the index information and saves them to the local

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

database for their use. The proposed system will be able to retrieve the files containing the query word within less than one second.

3. IMPLEMENTATION

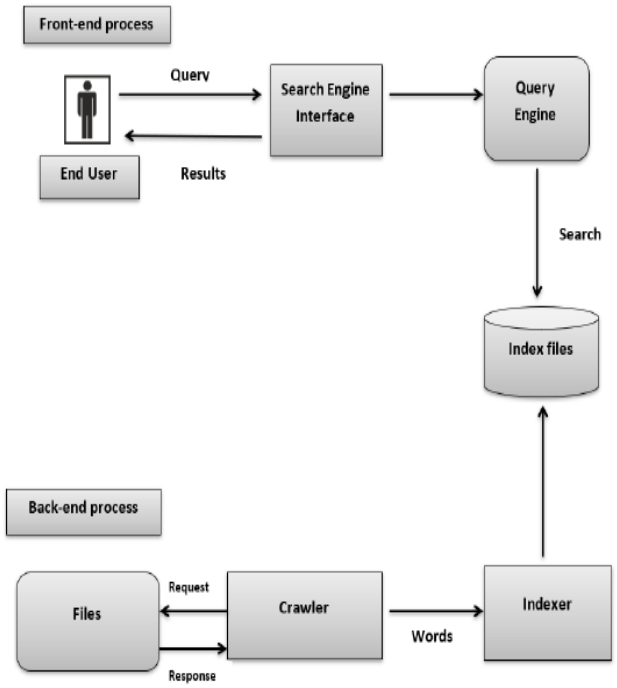

Fig 1: Architecture diagram

Dataset

A data set (or dataset) is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable, and each row corresponds to a given record of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Each value is known as a datum. Data sets can also consist of a collection of documents or files.

Fig 2: Dataset

Linear Search Algorithm

Searching is the process of finding some particular element in the list. If the element is present in the list, then the process is called successful, and the process returns the location of that element; otherwise, the search is called unsuccessful. Linear search is also called as sequential search algorithm. It is the simplest searching algorithm. In Linear search, we simply traverse the list completely and match each element of the list with the item whose location is to be found. If the match is found, then the location of the item is returned; otherwise, the algorithm returns NULL. It is widely used to search an element from the unordered list, i.e., the list in which items are not sorted. The worst-case time complexity of linear search is O(n).

Natural Language Processing

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.

4. EXPERIMENTAL RESULTS

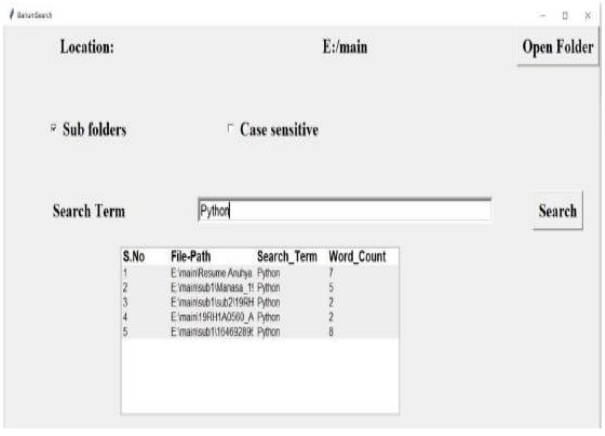

Fig 3: Search operation

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

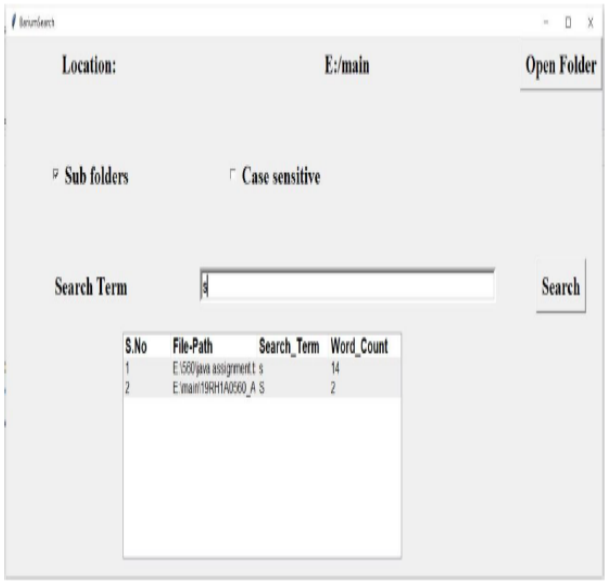

Fig 4: Search Operation (2)

5. CONCULSUION

Thus, a new approach for searching the data inside the files to display files that contains the searched term has been implemented in this project. This project also discusses about reading files of formats like .txt, .pdfs and .docx. Finally, considering the popularity of data extraction in many areas, the Barium Search application is provided. Future research might focus on included the image and other file formats too.

6. REFERENCES

[1] J. Giridharan and S. Vairavan, "Inverted index and interval lists for keyword search", 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), 2014.

[2] Hao Wu, Guoliang Li, and Lizhu Zhou, Ginix, “Generalized Inverted Index for keyword Search”, transactions on knowledge and data mining Vol:8 No:1 Year 2013.

[3] M. Singh and D. Garg, "Choosing Best Hashing Strategies and Hash Functions", 2009 IEEE International Advance Computing Conference,2009.

[4] N. Ram, R. Ranjan, S. Chakrabarti and D. Samanta, "Application of Data Structure in the field of Cryptography", in Proceedings of the International Conference, Computational Systems for Health \& Sustainability, R.V.College of Engineering, Bangalore,Karnataka,PIN560059,INDIA, 2015, pp.65-68.

[5] M. Ilic, P. Spalevic and M. Veinovic, "Inverted index search in datamining", 2014 22nd Telecommunications Forum Telfor (TELFOR),2014.

[6] H. Williams and J. Zobel, "Indexing and retrieval for genomic databases", IEEE Trans. Knowl. Data Eng., vol. 14, no. 1, pp. 63-78,2002.

[7] Harth A, Decker S. Optimized index structures for querying rdf from the web. InThird Latin American Web Congress (LA-WEB'2005) 2005 Oct(pp. 10-pp). IEEE.

[8] Brin S, Page L. Reprint of: The anatomy of a large-scale hypertextual web search engine. Computer networks. 2012 Dec 17;56(18):3825-33.

[9] Valduriez P. Join indices. ACM Transactions on Database Systems (TODS). 1987 Jun 1;12(2):218-46.

[10] Bayardo RJ, Ma Y, Srikant R. Scaling up all pairs similarity search. InProceedings of the 16th international conference on World Wide Web 2007 May 8 (pp. 131-140). ACM.