



A Study on Different Search Operations on Strings

Ritu Arora*, Mr. Gurdev Singh**

*M.Tech Scholar in JIET, Jind, Haryana, India.

**Assistant Professor & HOD CSE JIET, Jind, Haryana, India

Abstract: Search is the foremost and the most frequent operation performed on any dataset. The search becomes more critical when it is performed on the text. Text documents having the challenge in terms of organization as well no index or the key based storage specification. In this paper, some of the work done earlier in the area of search optimization is described in detail. This paper has categorized all the search algorithm based on the partial search, full text search etc. The analytical study of these all methods is also presented so that the effective selection of the adapted algorithm can be performed.

Keywords: Search, Optimization, Complexity, Reliability.

I. INTRODUCTION

Search is the act of seeking for more valuable information from available dataset. The outcome of search forms a list of data values or documents that match the predefined matching criteria. The searching mechanism should support the consequent retrieval process, which is a process to access more detailed information from repositories and result in the use these search vectors for different purpose in future. Search and retrieval concern the component precision: the desired keyword is first specified, and components should not be retrieved unless they are absolutely relevant.

The search effectiveness of the search system is based on two main vectors called the reliability and the complexity. The reliability is measured in terms of retrieval or the search of all possibility of the search component and the complexity is defined in terms of efficiency. How effectively the search can be performed over the dataset is defined in terms of efficiency. Higher the complexity of the search algorithm, lesser the efficiency of the system. The fundamental problem in search algorithm is how to organize collections of available dataset so that search and retrieval process becomes effective. Search process can be beneficial if it is easier to represent and feature analysis can be effectively. Search algorithms have mainly two problems. The first one is classification of dataset components so that easy categorization of multiple queries can be performed. Second is the identification of appropriate search word in in the library so that effective search will be performed. A general problem for a user is component searching from repository and hence retrieval.

Searching and retrieval of keyword from a text dataset is one of the major phenomenon associated with different applications. The good search enables quick retrieval of information. Search mechanism help users to localize relevant component from digital library. Software repository provides fast access to exact information which is looking for if its mange all components effectively and up-to-date [23]. The success of a searching in software repository relies on the implementation of a powerful searching technique. The retrieval mechanism should be flexible and should provide users to manipulate the search process and results by retrieving search history, adjusting search strategies, editing and sorting search results. In the case of component retrieval, evaluation is often focused on the effectiveness of a result set in a specific search. For retrieving components browsing and searching are two major paradigms. When appropriate search keywords are unavailable to users searching is become more popular [25]. Retrieval technique is concerned with the matching of a query and the representations of components for the purpose of identifying and/or retrieving those components that are relevant to that query.

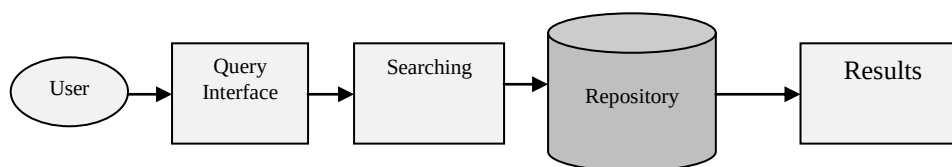


Figure1: A generic view of the retrieval process

Representations and retrieval techniques are interrelated. Information storage and retrieval is seems to be quite a simpler process. But in actual it is a quite complicated process. Usually a user either does not have the time or does not wish to spend the time reading the entire document collection, apart from the fact that it may be physically impossible for the user to do so. So, the term retrieval model is used. To make some difference one could roughly say that a retrieval technique is mostly concerned with the matching of the component with the

query representations. But a retrieval model is going a little beyond this, also taking into account the way of representing the components and the indexing process. Most of the research in component retrieval is depends on the different aspects and there exist a large number of retrieval models or techniques with different approach.

In this paper, a study on different search approaches is defined and explored. In this section, the basic requirement and components of a search algorithm is defined. The section includes the study on different parameters taken to compare the search algorithms. In section II, the work done by earlier authors in this area is discussed. In section III, some of the search approaches and their categorization is explored. In section IV, the conclusion driven from the work is also discussed and presented.

II. LITERATURE SURVEY

Lot of work is already done in the area of string matching. In year 2001, Gonzalo Navarro has defined a tour analysis based approach to perform the string matching. Author defined a study on different approaches on string matching under the error analysis. Author defined the experimentation to perform the performance analysis under different algorithms so that effective search over the web can be performed. Author defined the work in terms of analytical study on these algorithms and identify the best search algorithm. Author also discussed the associated problems with search algorithm and open area in this direction [1]. Another work on string matching was defined by Xinming Chen in year 2011. Author presented an improved Suffix Tree based approach to perform the string matching without the inclusion of buffer as the storage block. Author defined the pattern analysis over the sequence pattern generation and assembly. Author defined the algorithmic approach associated with AC algorithm so that effective search patterns will be identified [2]. A work based on hashing was proposed by Yang Xu in year 2011. Author defined a multi-dimensional analysis approach using hashing algorithm to perform the automation under the hash table transition. Author defined a collision analysis approach to perform matching based on each input character so that search can be optimized [3]. A work on string matching architecture based for intrusion detection was proposed by Lin Tan in year 2005. Author defined an optimized search algorithm to improve the searching over the database. Author defined the rule based analysis so that effectiveness based on the rule based match will be obtained[4].

A work on distance based search was proposed by Graham Cormode in year 2007. Author presented the edit distance analysis approach based on different string operation and encoding mechanism. In this approach, the distance based matching problem is defined to compute the distance between the pattern and the sub string. Author defined the substring moves based approach for the analysis[5]. A work on memory effective string matching algorithm was defined by Derek Pao in year 2010. Author defined the pipelining approach to reduce the system complexity. This approach was used to perform the matching process in linear time under the proportional length so that search will be optimized [6]. A review work was presented on different search algorithms were presented by Simone Faro in year 2013. Author defined the result based analysis on different algorithms under different application areas[7]. Another work on bit parallelism was defined by Heikki Hyyro in year 2005. Author defined a parallel bit analysis approach so that search patterns will be identified effectively. Author focused on short pattern based search [8].

III. STUDY ON SEARCHING TECHNIQUES

When studying information search techniques, everyone is aware that Internet is the richest information source that human has ever developed. There have been different strategies developed to support information retrieval for example, the catalogue or directory for a manually administered database, the automatic robot search engine to index Internet data, and the meta-search engine to scan search engines in parallel and merge the search results. Although strategies to search for information differ with each other, most of them follow the basic techniques: keyword search, full-text search, and classification-based search.

A) *Keyword search-*

It is an essential search technique used widely in library management systems. It allows users to find words and phrases located anywhere in the catalogue record. Keyword search would require information providers to provide appropriate keywords for each piece of information in the repository, which is a manual indexing process requiring skilful personnel.

B) *Full text search*

It is different from keyword search, as it automatically indexes all words in each document, and that avoids pitfalls in manual indexing work. Full text search presents a clear advantage that relevant documents are rarely overlooked due to its completed indexing of document. It is widely used in automated search engines for Internet information retrieval e.g. Alta Vista, HotBot, and so on. Meanwhile, it also means that irrelevant ones, that happen to contain certain relevant words out of context, can be retrieved if terms are not weighted according to significance.

C) *Classification based search*

This searching technique uses a number of predetermined perspectives for information classification and search. Yahoo is the primary classification based search engine on the web. There are several choices of classification based search by means of the different classification schemata. For example, faceted search uses faceted classification scheme; attribute-value based search uses an attribute-value based classification scheme; enumerated search uses a hierarchical enumerated classification scheme; and specification-based classification basically uses a feature-based classification scheme. Classification based search is widely used in the software component repositories.

In addition, with the development of hypertext techniques, one more search technique, Hypertext-based search: Hypertext-based search enables users move through a hypertext document by following links. It provides one of the newest forms to organize documents by using nodes and links. Nodes are associated with information blocks, and different types of links represent the different relationships between the source and destination nodes. Users are not constrained to the linear order of conventional documents any more, and navigate through documents at will, which may lead to a more flexible search process.

Search techniques continue to advance, integrating new and traditional searching methods. An effective search tool should accommodate continually expanding collections, a characteristic of most information repositories. We have reviewed four search techniques. The first three, keyword search, full text search, and classification based search, are index based search in a structured repository, while the last one, hypertext-based search, and navigates the repository by following the predefined links. The hypertext technique can be combined flexibly with other search techniques as a complement to empower information retrieval process. The automatic indexing feature of a full text search has proven that it can be widely used in text-intensive documents like articles and books. Meanwhile, as keyword search and classification based search predefine the keywords or catalogues used as an index to describe concepts relevant to the domain of discourse, they introduce semantic information absent in full text search. Each search technique has advantages and is appropriate in specific circumstances.

As component is itself collection of characters i.e. string. So string matching algorithm can be applied to search relevant component. Some technique are earlier developed to search a component which are based on the string matching algorithms are compared.

Name of Algorithm	No of Comparisons	Search Time Complexity	Order of Comparison	Main Features
Brute Force	No Pre-processing	$O(mn)$	Not relevant	Not an optimal one
KMP	$O(m)$	$O(m+n)$	Left to Right	Independent of alphabet size, decrease delay and decrease time of comparing
Rabin Karp	$O(m)$	$O(mn)$	Left to Right	Use hashing function, very effective for multiple pattern matching in 1D matching
Approximate string matching	-	$O(mn)$	Not relevant	First matching approximate then searching dictionary.
Hashing		$O(n)$ + time to compute $h(k)$		Good results than quick search, but suffered by collision problem
Suffix Tree	$O(m)$	$O(n)$		Suffix trees are much faster when the text is fixed and known first while the patterns vary.

IV CONCLUSION

In this paper, the study of the software reliability is been done under different models. These models basically explore the software reliability under the fault and failure analysis. The presented paper has given the better view of different reliability models analyzed by different authors.

REFERENCES

- [1] GONZALO NAVARRO, "A Guided Tour to Approximate String Matching", ACM Computing Surveys, Vol. 33, No. 1, March 2001, pp. 31-88. ACM 0360-0300/01/0300-0031
- [2] Xinming Chen, "AC-Suffix-Tree: Buffer Free String Matching on Out-of-Sequence Packets", 2011 Seventh ACM/IEEE Symposium on Architectures for Networking and Communications Systems 978-0-7695-4521-9/11 © 2011 IEEE

- [3] Yang Xu," A Multi-Dimensional Progressive Perfect Hashing for High-Speed String Matching", 2011 Seventh ACM/IEEE Symposium on Architectures for Networking and Communications Systems 978-0-7695-4521-9/11 © 2011 IEEE
- [4] Lin Tan," A High Throughput String Matching Architecture for Intrusion Detection and Prevention", Proceedings of the 32nd International Symposium on Computer Architecture (ISCA'05) 1063-6897/05 © 2005 IEEE
- [5] GRAHAM CORMODE," The String Edit Distance Matching Problem With Moves", ACM Transactions on Algorithms, Vol. 3, No. 1, Article 2, Publication date: February 2007 ACM 1549-6325/2007/02
- [6] DEREK PAO," A Memory-Efficient Pipelined Implementation of the Aho-Corasick String-Matching Algorithm", ACM Transactions on Architecture and Code Optimization, Vol. 7, No. 2, Article 10, Pub. date: September 2010. 1544-3566/2010/09-ART10
- [7] SIMONE FARO," The Exact Online String Matching Problem:A Review of the Most Recent Results", ACM Computing Surveys, Vol. 45, No. 2, Article 13, Publication date: February 2013 ACM 0360-0300/2013/02-ART13
- [8] HEIKKI HYYRO," Increased Bit-Parallelism for Approximate and Multiple String Matching", ACM Journal of Experimental Algorithmics, Vol. 10, Article No. 2.6, 2005, Pages 1–27. ACM 1084-6654/05/0001-ART2.6
- [9] Rajesh Prasad," Parameterized String Matching: An Application to Software Maintenance", ACM SIGSOFT Software Engineering Notes May 2010 Volume 35 Number 3 10.1145/1764810.1764829
- [10] LEENA SALMELA," Multipattern String Matching with q-Grams", ACM Journal of Experimental Algorithmics, Vol. 11, Article No. 1.1, 2006, Pages 1–19. ACM1084-6654/2006/0001-ART1
- [11] Rajesh Prasad," Multi-Patterns Parameterized Shift-And String Matching Algorithm with Super Alphabets", ICAC3'09, January 23–24, 2009, Mumbai, Maharashtra, India. ACM 978-1-60558-351-8
- [12] Mei-Chen Yeh," A String Matching Approach for Visual Retrieval and Classification", MIR'08, October 30–31, 2008, Vancouver, British Columbia, Canada. ACM 978-1-60558-312-9/08/10
- [13] Monther Aldwairi," Configurable String Matching Hardware for Speeding up Intrusion Detection", ACM SIGARCH Computer Architecture News Vol. 33, No. 1, March 2005.