SEARCH ALGORITHMS – AN AID TO INFORMATION RETRIEVAL IN DIGITAL LIBRARIES

JIBAN K PAL **FALGUNI PAL**

Abstract

Information growth trends are global issues and are common to all. This paper presents the concept of 'digital library' emerged as a leading edge technological solution to the persistent problems in modern libraries. Convergence of digital library with the Internet phenomenally increases the availability of digital resources, across the globe. It examines the real situation, where the morass of web resources presents a formidable hurdle to effectively accessing the relevant information. However, in digital libraries it is more acute, as the documents in digital library environment are free-formatted, voluminous and multiple media types; virtually the situation demands for the efficient and effective indexing/ retrieval mechanism. This problem has led to the rise of efficient search algorithms, as a powerful tool for search and retrieval. Over the last few decades, a considerable number of search algorithms have been devised addressing the requirement of different applications in IR systems – like soundex, metaphone, phonex, stemming, and many more. These algorithms are deeply concerned with the process of sorting and searching, which provides an ideal framework for the application to information retrieval in digital collections as well as Internet. Here an attempt has been made to discuss about the most popular and useful search algorithms and their impact on precision of search results, in searching the digital archives.

Keywords: Information retrieval/ Search algorithms/ Digital library/ Search engine/Information filtering.

1. Introduction

Information growth trends are global issues and are common to all. In this information-age, magnitude of information generation and it's proliferation into many forms have created problems of storage, processing and dissemination of information. Libraries and other service sectors have tried to overcome these problems by adopting several modern techniques. Gradually, the concept of 'digital library' has emerged as a leading edge technological solution to the persistent problems of enhancing access, enduring archiving, and expanding the dissemination of information. However, the digital library with the convergence of Internet – offer us a powerful means of managing electronic

resources. The concept of 'Library without walls' has almost become a reality because of the Internet, which is the single greatest source of information in human history, and challenges traditional conceptions of information rights. Practically, there are enormous sources that are relevant to any user's query, but uncountable stacks of web resources (Cyber jungle) drastically creates a hurdle for accessing the information effectively and efficiently. In fact, a considerable noise exists in retrieval of information – as huge number of heterogeneous resources available in large cyberspace.

So, the Internet has wrought a dramatic change in information business and phenomenal increase of resources on the web demands for an efficient method of retrieving the information. Though, semantic based search engine and meta-search engine stimulates resource discovery on digital collections. In this context different search algorithms have dramatic effect on how the web is indexed and will improve the discovery of resources on the Internet. These algorithms differ in potentialities to meet users needs and it is very difficulty to establish a common algorithm for retrieving the digital information. No doubt, search-algorithms are inevitable component in any retrieval mechanism concerning the information from a digital collection. Moreover, the identification and retrieval of digital information is determined by the efficiency of the searchers, but the majority of searchers are not enough to express their information need into actual query. To the common user of web, the searching is confined to the search engine by employing a typical keyword search, but the indexes and spiders/ robots are too poor in translating the resources into representational keywords. If fact most of the Internet surfers do not know the absolute potential of it. Therefore, their searches result in retrieving a large number of pages, where a majority of the retrieval is practically irrelevant and almost all the relevant pages remain suppressed or invisible to the search mechanism. Silverstein et al. (1998) found that about 85% of users look only at the first screen of their search results.

Above discussion addressed for the emergence of efficient retrieval mechanism. In this regard manual techniques (human indexing and retrieval) are highly labor-intensive and limiting when large databases or dynamic pages are involved. The problems of traditional methods have led to the rise of interest in techniques for searching information by automatic means, which pose a challenge to traditional Information Retrieval (IR). But in automated retrieval there are no consistent indexing and filtering practices at hand to ensure the quality and credibility of searching.

2. Understanding Search and Retrieval

The 'Search' is a systematic examination of information in a database, aiming in view to identify the items or objects, which satisfy a particular preset criteria1. In other way, searching means the operation of locating a specific object in a given sequence of 'n' objects. Here the amount of time required to locate a particular object or field of data or piece of information from the store – is called 'search-time'. It is worth to mention that the sorting is an aid to searching. It is also necessary to say that each item contains within itself a piece of information is termed as 'key' and a given key for a search is refer to as 'search-key'. Technically, a library and information professional should be

aware to distinguish between two different tasks in processing information – storage process (entering a key into the store, using specific storage algorithm) and retrieval process (accessing a key from the store). In retrieval process one can search to match for a given key using particular retrieval algorithm.

Now, the term Information Retrieval [read as IR, coined by Calvin Mooers in 1950] is basically concerned with the structure, analysis, organization, storage, searching, and dissemination of information. Landcaster opines that six basic sub-systems constitute the proper functioning of an IR System; these are document selection, indexing, vocabulary, searching, user system interface, and matching. Again these subsystems may be coupled into two broad categories, such as 'subject or content' – implies analysis, organization, and storage of information, and 'search strategy' – implies user query analysis, search formula creation, and actual searching.

3. Algorithm: Background Concept

Before pursuing specific discussion a brief consideration of the term would be useful. The word 'algorithm' itself is quite interesting and named after a famous Iranian mathematician, Abu Ja'far Mohammed ibn Musa al-Khowarizmi (c. 825 A.D.)2. An examination of the latest edition of Webster's Dictionary defines as – "any special method of solving a certain kind of problem". But this term has taken on a special significance in computer science, where it has come to refer to 'a computable set of steps to achieve a desired result'3 or 'a detailed sequence of actions to perform to accomplish some task'. Technically, an algorithm must reach a result after a finite number of steps, thus ruling out 'brute force search methods for certain problems, though some might claim that brute force search was also a valid (generic) algorithm. The term is also used loosely for any sequence of actions (which may or may not terminate)4. Therefore, an Algorithm is a procedure to solve a problem in an orderly, finite, step-by-step logical and straightforward manner5. or "A procedure consisting of a finite set of unambiguous rules which specify a finite sequence of operations that provides the solution to a problem or to a specific class of problems" – is called an algorithm6.

Algorithms are deeply concerned with the process of sorting and searching, which provides an ideal framework for the application to information retrieval. However, search-algorithm looks for an object or value or item in a data structure. There are dozens of algorithms, algorithmic techniques, data structures, and approaches. Here an attempt has been made to describe a few important search-algorithms, which are highly useful and popular to retrieve the digital information in libraries. These are as follows,

- Soundex Algorithm
- Metaphone Algorithm
- Phonex Algorithm
- Stemming Algorithm

3.1 Soundex Algorithm

Soundex method was originally developed by Margaret K. Odell and Robert C. Russell. They patented the original soundex algorithm in 1918 [cf. U.S Patents 1261167 (1918), 1435663 (1922)]. As the name suggests, this method is based on the six phonetic classifications of human speech sounds, which in turn are based on the movement of lips and tongue to make those sounds. These categories may be represented as – Bilabial (Sound produce with both the lips); Dental (Articulated with the tip or blade of the tongue against or near the upper front teeth); Labiodentals (Utter with the participation of the lip and the teeth, the labiodentals sounds ' f ' and ' v '); Alveolar (Articulated with the tip of the tongue touching or near the teeth ride); Velar (Formed with the back of the tongue touching or near the soft palate, the velar ' k ' of 'kill' cool); Glottal (the interruption of the breath stream during speech by closure of the glottis).

Technically the problem exists concerning those terms that are often misspelled. Even, this problem has received considerable attention in connection with the airline reservation systems, library cataloguing systems, census operations, database designing and in other applications involving people's names when there is good chance that the name will be misspelled due to poor hand-writing or voice transmission or spelling choice, etc. Suppose, the names, having variable length, can have strange spellings, or have multiple spellings, and sometimes they are not unique, especially across different cultures or national boundaries. Therefore, to cope with this persistent problem, one has to consider the phonetic algorithms that can find similar sounding terms or names. Such families of algorithms are generically called as 'Soundex', after the first patented version. The goal is to transform the argument into some code that tends to bring together all variants of the same name.

- Poor Precision
- Dependence on Initial Letter
- Noise In-tolerance
- Poor Handling of the Name Equivalence
- Produces Unranked, Unordered Returns
- Poor Handling of the Name Syntax Variation
- Poor handling of the abbreviations and initials
- Poor handling of Phonetic Variations (Silent Consonants)
- Poor Handling of the Name containing particles
- Differing Transcription Systems

Though, several problems exist with the soundex algorithm, still it is better, than the best. No single algorithm that relies on a single mapping of sounds to letters can be expected to perform well across multiple linguistic systems, especially not when some degree of translation has been involved. John Hermansen (1985) describes that a

fundamental problem for soundex and its derivatives is that they are applied as a universal name-search method. An algorithm designed largely for English names is less well suited to handle names with sound patterns and structures as diverse as Arabic, Chinese, Russian, to name but a few.

3.2 Metaphone Algorithm

The discussion cited above make the sense that one cannot rely solely on Soundex in an advanced name-search system. Therefore, the original Metaphone algorithm was designed as a replacement of Soundex algorithm, and was described in the Computer Language magazine (December, 1990). It returns a rough approximation of how an English word sounds and uses many common rules of English pronunciation that Soundex does not cover10. This algorithm reduces the word (key) to a 1 to 4 character code using relatively simple phonetic rules for typical spoken English. Metaphone ignores all the occurrences of vowels after the leading letter, but retaining the same when they come as an initial letter of a word. It then reduces the remaining alphabet to sixteen consonant sounds viz. B, X, S, K, J, T, F, H, L, M, N, P, R, O, W and Y. Where the 'sh' sound is represented by an 'X' and a zero is used as a representation of 'th' sound [as it is similar to the Greek Ø (theta)]. Again, if a letter is duplicated then it considers only one occurrence, unless it is a 'C'11.

3.3 Phonex Algorithm

The early operational success (through Metaphone, Hermansen model, NYSIIS model, etc.) also raises the question of whether the Soundex methodology can be adapted for a particular database to improve results? A. J. Lait & B. Randell (1996) set out to answer just this question after finding the Soundex recall rates, which were very much disappointing. They compared the performance of several name-matching algorithms, including the basic soundex method. Searches were conducted on a data set of 5600 unique surnames, chosen to represent names beginning with each letter of the alphabet at a frequency of occurrence reflecting actual alphabetic distributions of names, and including as well names of varying lengths. The study found that soundex (judged to be the best of the four algorithms compared) returned only 36.37% of actual correct matches, and that more than sixty percent of names that were correct matches for query names were not returned.

Using the same database against which Soundex was tested, they progressively altered the soundex code until the maximal rate of accurate returns was found, with the minimal increase in incorrect matches. The resulting algorithm was titled "Phonex", which seems to offer significant improvements over soundex12. They decided to call it "Phonex", as it derived from two basic methods on which it is based – Metaphone & Soundex. Phonex was able to return 51.79% of the correct matches in the database, as opposed to Soundex's 36.37%. While the Phonex is an improvement, it still leaves almost half of the correct matches undiscovered. Lait & Randell also note that their improved algorithm addresses neither corrupted data nor multi-ethnic data. Phonex algorithm converts each word into an equivalent four-character code using the following steps:

Step-I : Pre-process the name according to the following rules:

- Remove all trailing 'S' characters at the end of the name.
- Convert leading letter-pairs as follows; KN = N, WR = R, PH = F
- Convert leading single letters as follows; H = Remove; E, I, O, U, Y = A; K, Q = C; P = B; J = G; V = F; Z = S

Step-II : Code the pre-processed name according to the following rules:

- Retain the first letter of the name, and drop all occurrences of A, E, H, I, O, U, W, Y in other positions.
- Assign the following numbers to the remaining letters after the first;
  1 = B, F, P, V
  2 = C, G, J, K, Q, S, X, Z
  3 = D, T (if not followed by C)
  4 = L (if not followed by vowel or end of name)
  5 = M, N (ignore next letter if either D or G)
  6 = R (if not followed by vowel or end of name)

Ignore the current letter if it has the same code digit as the last character of the code.

Step-III : Convert to the form 'letter, digit, digit, digit' by adding trailing zeros (if there are less than three digits), or by dropping rightmost digits if there are more than three.

3.4 Stemming Algorithm

A stemming algorithm is an algorithm that converts a word to a related form, means that derives the root form of a word. One of the simplest such transformation is conversion of plurals to singulars, another would be the derivation of a verb from the gerund form (the "-ing" word). In Natural Language Processing (NLP), stemming is the process of merging or lumping together non-identical words, which refer to the same principal concept. Stemming is usually done by removing any attached suffixes, and prefixes from the index terms before assignment of the term. It is commonly accepted that removal of word-endings (often called as suffix stripping) is a good idea – the idea is to reduce the words to their word roots by automatic handling of word endings. Removal of prefixes can also be useful in some subject domains; chemistry is an obvious example (eg. Di-Oxide, Tri-Oxide, Cyclo-Buten, Cyclo-Penten, etc). Since the stem of a term represents a broader concept than the original term, then the stemming process eventually increases the number of retrieved documents (including both word roots & word derivations) and potentially improves the recall value. Therefore, the stemming of words is the common form of language processing in most of the Information Retrieval (IR) systems. It is an important feature supported by present day indexing and searching systems.

4. Implication of Search Algorithms in DL Software and Web Search Engines

4.1 Concept of DL

A 'digital library' (often used as 'electronic library' and 'virtual library') is not merely a collection of electronic information – it is an organized and digitized system of data that can serve as a rich resource for its user community. One of the earliest definitions stated that a Digital Library (DL) is a – "service - an architecture - a set of information resources, databases of texts, graphics, video, etc. - a set of tools & capabilities to locate, retrieve and utilize the information resources available In-spite of several definitions available, the Association of Research Libraries (ARL) defines DL with five basic elements that have been identified as common to these definitions18. So, the DLs are the logical extensions and augmentations of physical libraries in the electronic information society. As such, digital libraries offer new levels of access to broader audiences of users and new opportunities for the library and information science field to advance both theory and practice"19. As a concept 'digital library' has different connotations for different professional groups (Marchionini & Komlodi, 1998). For IT professionals it is a powerful tool and mechanism for managing distributed databases. To the Information Science community it represents a new means of extending and enhancing access to distributed/ remote information resources. The IR community perceives DL to be another extension of Information Retrieval Systems. While traditionally IRS has been focusing on retrieving document surrogates, what has changed today is perhaps the nature of 'documents' and their 'surrogates' (Belkin & Croft, 1992 ; Croft, 1996). For the LIS community it connotes a logical extension of what libraries have been doing since time immemorial (Garfield – 2000 ; Thorin & Sorkin – 1997). Thus, DLs are augmenting resources, enlarging the services and audiences of libraries.

4.2 Retrieval as the Basis of DL-Software

Roots of the modern DL may be traced back to the Information Retrieval Systems (IRS) of 1960s and the Hypertext Systems (HTS) of the 1980s20. Digital libraries have evolved based on the techniques and principles developed by early IR researchers such as Moores, Perry, and Taube, etc. Salton (in 1968) pioneered the era of automatic indexing and search mechanism. Thus DLs are on the solid foundations of more than three decades of research in IR. However, the DLs as we know them today were conceived and built only since the 1990s. In fact, building a digital library is expensive and resource-intensive. In designing a digital library, there is no decision more important than the selection of the 'software. There are no rows of comforting bookcases to show the size of DL collection. There is no friendly face to help guide the client to the right information. There is just the 'DL-Software'. When clients don't find what they need in the online catalog of a traditional library, they can browse the shelves. In a DL, the software is the catalog and the shelves, and of-course more than the traditional library. It has to be easy to use and it has to be effective.

So, one of the main technical issues at the heart of the developmental efforts of DLs is identifying the appropriate software, i.e. what are the software components? What software properly connects the repository, mechanisms for identifying and organizing the digital objects, access and search tools, interfaces and other pieces? This means

that the software must provide users with an effortless way to perform effective searches. If the clients are comfortable with Web search engines, the software you selected should include a search engine that provides relevancy rankings. So, in DL Software selection, always you have to ask - Can you perform sophisticated searches with your digital library software that produces highly relevant retrieval? Even, if you find a package that meets all of your other criteria but has an unsatisfactory retrieval system, you may need to investigate the search algorithm employing for IR. The software vendor, your own IT department or a consultant (when it is open source) may be able to design an overlay that will meet the needs of the end-users. Unfortunately, packages that can accommodate the speed and precision searches desired by information professionals frequently do not provide the iterative prompts needed by end-users.

Here we will discuss a number of most promising open sources DL Software's along with the underlying mechanisms in their searching. Additionally, a careful effort has been made to investigate the algorithms used in popular DL Software's and Web Search Engines for their retrieval mechanism

5. Algorithms vs. DL Software

A number of DL softwares came into their existence in distributed cyberspace and are being available freely (sometimes open source) for organizing the digital libraries in real practice. This new paradigm has become a critical component in the mission of almost all academic institutions or learned societies, and extending the learning process beyond the walls of classrooms. These DL softwares are Greenstone, Dspace, TeN-Acado, EPrints, FEDORA, CDS-Ware, VITAL, and so many. Here we are intended to make a clear representation for analyzing a few of those softwares in terms of their algorithmic appliances towards retrieval performance in discovering the objects from the repositories.

5.1 Greenstone

Popularly known as GSDL – is free, multi-lingual, open-source digital library software developed by the New Zealand Digital Library Project (http://www.nzdl.org/) at the University of Waikato, and distributed in cooperation with UNESCO and the Humanity Info NGO at Belgium21. It is basically a suite of software for building and distributing digital library collections. It provides a new way of organizing information and publishing it on the Internet or on CD-ROM. However, GSDL integrates functions such as metadata, full text search and retrieval, multilingual support, access management, multiple formats and freely available under the terms of the GNU General Public License. It runs on any versions of Windows, Unix, Mac OS-X and both source code and binaries are available for download.

Use of this package internationally is growing very rapidly and several activities on Greenstone are an indication of its growing popularity. Any DL software normally store the data in a DBMS, however, GSDL, uses its own database. The Greenstone search engine is tailored for extremely efficient storage, and can compress an index to a large

collection into only five per cent of the size of the original text22. Greenstone is built around the MG (Managing Gigabytes) search engine. MG or MG++ provides the underlying indexing and retrieval systems in Greenstone, which offers flexible stemming methods, weighting terms, term frequencies, and decompression of texts, etc. Therefore, interestingly it supports different stemming methods with all four combinations of stemming and case folding can be used in queries and even weightings can be applied to individual terms in a ranked query i.e. allows the documents with higher rated query-words nearer the top among the total returned-documents. Moreover it facilitates end users in displaying the term frequencies i.e. the number of times each query term appears in the collection can be displayed during the search. Greenstone can harvest documents over OAI-PMH to include them in a collection and uses various metadata formats like DC (qualified & unqualified), MARC, XML, AGLS, etc. GSDL query syntax involves the Boolean operators, Term modifiers (stemming, case-folding, term-weighting, case sensitive or insensitive, stemmed or un-stemmed), Proximity searching, Fielded searching, etc.

5.2 Dspace

Dspace is fairly powerful, freely available, open source software came into its existence through a joint project of MIT and HP Labs24. It is a digital library system that is designed to capture, create, store, index, preserve, retrieve, and redistributes multiple forms of digital objects. Dspace is a service model for open access and/or digital archiving for perpetual access, which can be customized or extend its capabilities. It is basically a platform to build an Institutional Repository (IR) and the collections are interoperable, searchable and retrievable on the Web. It has inbuilt mechanism to expose metadata using OAI-PMH protocol through OAICat (an open source OCLC product for harvesting metadata), which can be easily extendable to other metadata schemas by developing java programs. Dspace by default uses qualified DC set for furnishing metadata and exposes metadata using unqualified DC format for the purpose of OAI-PMH. Moreover, The recent versions (1.2.2 beta onwards) allow clienteles to define their own metadata formats by using XML input-forms, i.e, these versions allow users to extend to Non-DC formats. Expectedly future versions of DSpace may permit a more integrated use of specialized metadata. Keeping such an intention MIT's SIMILE project is investigating Semantic Web technologies to provide end-to-end support for capturing, navigating, browsing, and searching heterogeneous community-specific schemes, metadata, and content types. Perhaps the support for multiple metadata formats (like VRA core, IMS/ IEEE-LOM, etc.) may greatly enhance the use of DSpace for archiving the digital objects. In this regard Dr. Prasad has presented a detailed discussion in a user meet at Cambridge25.

5.3 Acado

It is a comprehensive software solution for e-learning and digital library initiatives, developed by the Transversal e Networks (TeN), an US Technology Group company29. The Novel concept of Acado was actually germinated at the prestigious institution at IIT, Kanpur and became into existence through Indian Institute of Information

Technology & Management - Kerala (IIITM-K)30 at Trivandrum as a part of the Industry incubation programme of the Institute. This web based e-learning and communication tool not only provides solution for digital library projects but also incorporates the technology for effective collaboration and groupware applications like email, synchronous chat, threaded message board, discussion forums, common web-space for uploading files for remote access, customized news, alerts, calendar and much more. This suite of software is already at work at some of the most reputed academic institutions in India like – IISc Bangalore, IIT Chennai, IIITM-K Trivandrum, Anna University, Ramanujan Computer Centre, College of Engineering at Trivandrum, etc. Many customers from corporate (like TCS, Wipro), R&D (like VSSC, ADA, NAL), and academic sectors (like IIT's, IIM's, IIIT's, ISB, etc.) in India are testimony for it. Among the prestigious institutions like IIM Calicut, IIM Ahmedabad, Cochin University of Science and Technology, RRL Trivandrum, etc has installed the same software for their digital library solution.

5.4 EPrints

GNU Eprints (http://software.eprints.org) is free open source software developed at the University of Southampton, designed to create a pre-print institutional repository for scholarly research, but can be useful for many other purposes. It is worth to mention here, Eprints archive of the Indian Institute of Science, Bangalore is the first OAI-compliant academic institutional repository in India, installed in late 200233. EPrints uses UTF-8 (also encoding of Unicode) to store all metadata. It has no trouble to handle more than one language at once. It is highly possible to configure a field to be multilingual. This is most useful feature for titles and abstracts. GNU Eprints has an unique search mechanism and a set of innovative features to allow end-users to perform multiple Web search operations. EPrints supports OAI-PMH that allows a specialized search-engine (a.k.a. Harvester) to perform the query and to obtain a list of all items in the archive, and dublin core metadata about each record. Configuring & registering the OAI interface will increase the visibility of the records, as one could be able to search across many OAI-compliant archives rather than having to search each archive in turn. Eprints' Tim Brody's Celestial (http://celestial.eprints.org/), a software that harvests metadata rom OAI-compliant repositories and re-exposes that metadata to other services. Tim Brody has also crated the Citebase (http://citebase.eprints.org/) that is a citation-link-based "google" for the OA literatures, ranking papers and authors by citation impact or download impact.

5.5 Harvest

It is an integrated set of tools to gather, extract, organize, and search information across the Internet. Basic objective of the Harvest is to provide a flexible and custom search system on the Internet that can be configured in various ways to create many types of indexes34. Harvest also allows users to extract structured information from various formats and build indexes that allow these attributes to be referenced during queries. Another important advantage of Harvest is that it allows users to build indexes using either manually constructed templates (for maximum control over index content)

or automatically extracted data constructed templates (for easy coverage of large collections), or using a hybrid of the two methods. Harvest is designed to make it easy to distribute the search system on a pool of networked machines to handle higher load. Basically, Harvest uses three search engines for the purpose of indexing and searching. By default, harvest uses GIMPSE (GLobal IMPlicit SEarch) index / search engine and rest of two WAIS (Wide Area Information Server) and SWISH (Simple Web Indexing System for Humans), as supportive index/search engine. They uses exact matching, similar matching, string search, etc. as technique for supporting to searching. The application of search algorithms in digital library environment makes the searching easy and meaningful. And improve the recall and precision in information retrieval.

5.6 CDS-Ware

CERN Document Server Software (http://cdsware.cern.ch) is free, open source software developed by CERN, the European Organization for Nuclear Research in Geneva. It is basically designed to run an preprint server, online library catalogue, or a document system on the web. CDS-Ware has several interesting features that are manifolds – viz. search engine redesigned to yield five times more search performance for larger sites (WebSearch, BibWords); fulltext indexation of PDF, PostScript, MS Word, MS PowerPoint and MS Excel files (WebSearch); integrated combined metadata or fulltext or citation search (WebSearch); multi-stage search guidance in cases of no exact match (WebSearch); OAI-PMH harvestor (BibHarvest); moreautomatic daemon mode of the indexer, the formatter and the collection cache generator (BibWords, BibFormat, WebSearch); etc.

Many other DL softwares like FEDORA (Flexible Extensible Digital Object Repository - http://www.fedora.info/index.shtml), VITAL (a DL tool developed by VTLS http://www.vtls.com.products/vital.shtml), etc. also have a good deal of algorithmic appliances for indexing and searching the archives.

6. Algorithms vs. Web Search Engine

Search Engine is a software program, which acts as a mediator between documents/ resources and the users/ searchers. There are various search engines through which one can identify websites, without having knowledge of website addresses, using keywords. 37.com (www.37.com) provides a list of most useful and evaluated search engines. Example of powerful search engines includes Google (www.google.com) Yahoo (www.yahoo.com), Ht://Dig (http://www.htdig.org/), etc. It is worth to mention here, Ht://Dig is developed at San Diego State University as a way to search the various web servers on the campus network i.e. a complete world wide web indexing and searching system for a domain or 'Intranet', so as to serve the search needs for a single company or campus, or even a particular sub section of web site. It supports the text base searching and uses the word matching algorithms and are supportive of several search algorithms (in different combinations) like – soundex, metaphone, stemming, synonyms, accent-stripping, sub-string & prefix, exact-match algorithms, etc. However, we may assume that the word Google and Yahoo need no further introduction as they have become the

far most utilized search engine worldwide. These two popular competitors are fighting in the real world using high-performance search-algorithms as their modern swords. When Google prefers to use 'PageRank' algorithm then Yahoo uses 'Inktomi' algorithm for the purpose of search and retrieval. As the name implies, the original PageRank algorithm was described by Lawrence Page and Sergey Brin. PageRank and Google are trademarks of Google Inc., USA. PageRank is protected by US Patent 6285999.

However, there are lots of controversies in application and performance of these two algorithmsAndy Beal reported that Tim Mayer of Yahoo Corporation confirmed that Yahoo is using a search technology that is not actually Inktomi. Beal speculates that Yahoo Search is an advanced form of Inktomi, as it doesn't match results at HotBot or MSN, though both of which serve Inktomi results. Later on Beal noted that the results on Yahoo's search were slightly different from other sites that use Inktomi – even one can do some searches to determine the distinct algorithmic flavor of Yahoo. Furthermore, Grehan (an author of Search Engine Marketing), wrote that Yahoo will "exclude Inktomi paid inclusion URLs from its main results". So many explanations are due in determining the search algorithm that exactly the Yahoo is using currently. Again the issue became interesting, when Sherman said, "while Yahoo and Google are likely using similar algorithms, one reason for the differences... Studying the relationship between 'keyword-density' and 'search-ranking' for those keywords revealed interesting clues. Their findings are really considerable – the first major difference "that jumps out in the Yahoo results is the preference Yahoo's algorithm seems to have for more words on a page. The average number of words on a page for Google was 943 while Yahoo's average word per page in the top 10 results was 1305". Secondly - "Yahoo had an average keyword density of 19.6% while Google's title keyword density is 16.9% for the results compiled". Again, "Link text results show that Yahoo prefers less link text words on a page and more keyword occurrences within those words". After all they concluded as follows – "While the keyword density is almost identical, Yahoo's is 3.4% compared to Google's 3.6%, Yahoo definitely seems to have a preference for pages with more bold text (92 words compared to 65 words) and more occurrences (1.7 repeats for Yahoo compared to Google's 0.7)". This issue still requires further study

7. Conclusion

Digital libraries present an exiting opportunity not only for LIS professionals but also for stakeholders, aggregators, publishers, technologists, authors, and even the end-users. Here the emergence of appropriate retrieval mechanism definitely a considering factor at the heart of the developmental efforts of digital libraries. Here the main technical issues persists under the purview of the retrieval mechanisms are – identifying and organizing the digital objects, indexing and searching tools, precision and relevancy of search results. To cope with those issues, digital librarians should investigate the efficient search algorithms for providing an effortless way to perform most effective search results from digital repositories. Therefore, designing and development of high-performance search algorithms has been a popular line of research in the digital library arena.

Therefore, search algorithms are vital toolkit for information retrieval in digital libraries, means for DL software. Virtually the LIS professionals, when selecting software for building their digital libraries, search algorithms should be kept into their consideration for such a selection. More clearly to say it is very much essential to consider the appropriate search algorithm in designing and selecting DL software or web search engine. Remarkably search-algorithms are 'digital finders' or might be termed as 'digital intermediaries', as they are basically responsible for mediating between the actual object and internal DBMS – as like as the role of human intermediaries those who bridges between the searcher and Database. Both the phenomena serve the same purpose (obviously in different platforms) of searching as well as displaying the objects to satisfy the end-user's query.

References

1. Dictionary of Algorithms and Data Structures, hosted by National Institute of Standards & Technology (NIST), visited on 12th January 2006. Source: http://www.nist.gov/dads/
2. FOLDOC: Free On-Line Dictionary On Computing, visited on 12th January, 2006 http://foldoc.doc.ic.ac.uk/foldoc/index.html
3. Singh, B. S. (2003): Search Algorithms (paper-E) in the DRTC Workshop on Digital Libraries – theory and practice, in DRTC, Bangalore.
4. Lang, Richard A.: SOUNDZ – The Compact Disc Search Engine, project submitted to School of Design Engineering and Technology, 2001, under the supervision of Dr. Peter Knaggs. Visited on 2nd Feb, 2004, Source: http://www.rlang.co.uk/projects/soundz/Soundz.pdf
5. Venkatalakshmi, K (2002): Developing a word stemming program using Porter's Algorithm. Project submitted under the guidance of of T B Rajashekar to NCSI, Bangalore.
6. Porter, M. F. (1980): An algorithm for suffix stripping. in Program, V.14, No.3, July, P.130-137, visited 2nd March, 2004, Source: www.tartarus.org/~martin/PorterStemmer/def.txt
7. Witten, Ian H. and Bainbridge, David and Doddie, Stefan (2001): Greenstone – open-source DL software, in Communications of the ACM, Vol.44, No.5, May, P.47 [also visit the URL at http://www.greenstone.org/
8. Personal communication (face-to-face) in a Tutorial on Building Digital Library using GSDL and Dspace Software, organized by the Indian Statistical Institute Library, held at Kolkata on December 11th 2006.
9. Prasad, A R D (2005): Using Multiple Metadata formats in DSpace, presented at an User Meet on 6-8th July, University of Cambridge, UK
10. The Apache Jakarta Project: Lucene. Last visited on 22nd Nov, 2006, Source: http://lucene.apache.org/java/docs/index.html

11. Prasad, A. R. D and Patel D (2005): Lucene Search Engine – an overview (Paper H) in DRTC-HP International Workshop on Building Digital Libraries using Dspace held on 7th – 11th March, at DRTC, Bangalore. (also visit the URL: http://lucene.apache.org/java/docs/queryparsersyntax.html

12. Gilleland, Michael: Levenshtein Distance, in Three Flavors. Visited on 12th March, 2005, Source: http://www.merriampark.com/ld.htm

13. TeN: Transversal e Networks, visited on 10th Nov, 2005, Source http://www.transversalnet.com

14. IIITM-K: Digital Library, Visited on 10th Nov, 2005 Source: http://www.iiitmk.ac.in/iiitmk/digitallibrary.htm

15. IIITM-K: ACADO – User Manual, Visited on 10th Nov, 2005 http://www.iiitmk.ac.in/help/user/chapter/html/Course1_lo.htm

16. Personal communication through E-mail: On Subject: Re: Acado DL Information, Dated on Tuesday, 23 Mar 2004 09:01:24 +0530 (IST) From: Chief Technology Officer (TeN) manjuls@transversalnet.com To: Jiban Krishna Pal <jiban@isical.ac.in>, Cc: jjoseph@transversalnet.com

17. Munshi, U. M: Establishing Institutional Repositories – an Overview, in Handouts of the Tutorial on Building Digital Library using GSDL and Dspace Software, organized by the ISI Library, held at Kolkata on December 11th 2006.

18. Yahoo vs. Google: algorithm standoff. Inside report of WebProNews, by Garrett French (a member of the World's Forum for e-Business Professionals), posted on February 23rd 2004, Source: http://www.webpronews.com/2004/0224.html

BIOGRAPHY OF AUTHORS

Jiban K. Pal (b.1972) holds B.Sc in Zoology, B.Ed, MLISc, certificate in Computing and Museum studies. Curretly Mr. Pal is working in the Periodicals Unit of the Library Documentation & Information Science Division of ISI. He has several published & communicated papers to his credit and participated in various national & international level workshops and conferences. He is the life member of BLA, IASLIC and ISI.

Email : jiban@isical.ac.in

Falguni Pal (b.1978), B.Com. Hons, MLIS, is presently working as the Librarian at Pailan College of Management & Technology, Kolkata since October 2005. Prior to her present attachment, she was the Librarian at the Bijoygarh Vidyapith, Kolkata for two years and Library Assistant at the Sivanath Sastri College, Kolkata for one year. She holds more than four years of professional experience and expertise in the field of Library & Information science. She is the life member of Bengal Library Association.

Email : phalgun_78@yahoo.com