ISSN 2320 - 2602
Volume 3, No.1, January 2014
International Journal of Advances in Computer Science and Technology
Available Online at http://warse.org/pdfs/2014/ijacst02312014.pdf

Intelligent Mechanism for Redundant Result Removal from Search Results

Aarti Singh
Associate Prof., MMICT & BM, MMU, Mullana, Haryana, India
Singh2208@gmail.com
Rajeev Soni
Research Scholar MMICT & BM, MMU, Mullana, Haryana, India
rajeevsoni.mca@gmail.com

ABSTRACT

Internet has become an inseparable part of our lives these days. Everyone is making use of internet for searching information. Search engines are an important component of our internet access, facilitating information retrieval from the web. Typically users submit a query containing combination of keywords and receive a list of web pages as result. However many web pages are being replicated on the web which appear redundantly in the search results, thereby reducing the search efficiency. This work aims to propose an agent based intelligent mechanism for detecting and removing redundant web pages from the search results.

Key Words: Agent Technology, Document Similarity, Document Shingle, URL, WWW.

1. INTRODUCTION

WWW has become an information repository for all of us and search engines are tools for extracting information from this repository. Information Retrieval is the activity of obtaining information resources relevant to an information need from a collection of information resources. Searches can be based on metadata or on full text indexing. Information Retrieval is the art of presentation, storage, organization of and accessing the information items [15]. The goal of any information retrieval system is to satisfy user's information needs.

However, many documents and web pages are being stored at multiple locations and are available in multiple versions or formats on the web. Present day search engines fetch all those redundant documents and index them in their indexes. As a result, while providing search results to the user queries these documents appear in the form of multiple links and when user clicks on those links same document or web page gets opened. This is a frustrating problem for the user since this leads to wastage of time and decreases user search satisfaction. This provided us the motivation to propose an intelligent mechanism for detecting duplicate documents listed in search engine index and removing it from the search results leading to improved search efficiency and user satisfaction. Rest of the paper is organized as: Section 2 provides review of relevant literature, Section 3 presents the proposed framework, and Section 4 concludes the work.

2. LITERATURE SURVEY

This section presents the work of eminent researchers in the field of information retrieval and agent technology.

Molina.et.al.in [1] has presented a mechanism to identify replicated documents from hyperlinked document collections. Zobel.et.al.in [2] has explored the syntactic techniques for detecting contents equivalence. Bernstein et.al in [3] has highlighted seven criteria depending upon research outcomes that should be measured.

Kitsuregawa et.al in [4] has described criteria for measuring the certainty that a newly crawled page appeared between the previous and current crawls. Kazar et.al in [5] proposed an agent based architecture, for searching information from distributed heterogeneous sources.

Clark et.al.in [6] has presented a general framework for information retrieval from WWW, extracting information from heterogeneous information sources that exist in distributed environment. Brewing ton et.al in [7] describes the strength of mobile agents in distributed information retrieval. Molina et. al in [8] has presented an architecture for the incremental crawlers, that can improve the freshness of the collected web pages in the index. Rajesh war et.al in [9] highlighted the security issues associated with

mobile agents. Molina et.al in [10] suggested a mechanism for detecting copies of existing documents, based on comparing the word frequency of the new document against those of already registered documents. Broder. et.al [11] has presented a mechanism to create a cluster of all syntactically similar documents.

Siddhartha et. al [12] presented SpotSigs, an algorithm for extracting and matching signatures for near duplicate detection in large Web crawls. Prasanna Kumar et.al.in [13] presented a survey of the existing literature in duplicate and near duplicate detection from web documents while web crawling. Bar Yossef et al [14] presented Dust Buster algorithm, for uncovering DUST (Different URLs with Similar Text). Jyodip Datta [15] presented a report on ranking in Information Retrieval focusing on various aspects and models of information retrieval.

Ahmad M.Hasnah [16] proposed a novel data reduction algorithm employing the concept analysis which can be used as a filter in retrieval systems to eliminate redundant documents. It can be used to reduce the size of stored information in databases or data collections.Dean.et.al.in [17] presented two algorithms to identify related web pages. Both algorithms use only the connectivity information in the web not the content of pages or usage information.Molina.et.al.in [18] describe the order in which a crawler should visits the URL and elaborated important metrices along with ordering schemes. Shivkumar et.al. in [19] presented a technique for computing pair wise document overlap.Molina.et.al.in [20] proposed a copy detection mechanism .

From the above literature review it is clear that replicated documents in the WWW are a point of concern and many researchers have proposed mechanisms to detect them; however no such mechanism has been adapted as the standard till date. Researchers have also highlighted that mobile agents can be useful in distributed information retrieval. Being intelligent they have already been employed widely in web based applications and have proved promising solutions. This provided us the motivation to propose an agent based framework to detect duplicate documents from the index maintained by the search engine, so that same documents are not returned to the user as separate links. Next section elaborates our proposed framework.

3. PROPOSED WORK

Before explaining the proposed mechanism, traditional information retrieval system employed in present search engines needs to be understood. Figure 1 given below provides the high level view of the traditional search engine architecture.

Search Engines (SE) are very efficient tools to retrieve information from web pages. A SE comprises of a crawler that fetches web pages from World Wide Web (WWW) for later use. Crawler also known as Web Crawler (WC) is a program that automatically traverses the web by downloading the documents and following links from one page to the other page. WC are also known as robots, worms and spiders etc. SE and many web sites make use of crawling as a means of providing the latest data. WCs creates a copy of all the visited web pages in document repository of SE, which is then indexed by its indexer component. Indexer reads the documents repository and converts each document into a set of word occurrences called hits. The hits record the word position in a document. Then for efficient retrieval of web pages, an inverted index containing list of web pages based on frequency of key words is being maintained.

This system revolves around the inverted index created and maintained by the indexer component. However in WWW many documents and web pages are being replicated [3] due to many reasons such as: same text with different dates available through multiple URLs, versions created for different delivery mechanisms such as HTML/PDF, reuse and republication of text, syndicated news articles delivered in different venues, policies and procedure for the same purpose in different legislatures, revisions and versions of documents. These replicated documents or pages available from different sources are listed as separate web pages in the inverted index under same category and are also returned as separate documents and links in the search results. These are referred to as redundant results here. Redundant results reduce search efficiency of the search engine since sometimes many results referring to same document/page is being returned from multiple sources. Moreover this leads to dissatisfaction of the user because of wasted time and efforts.

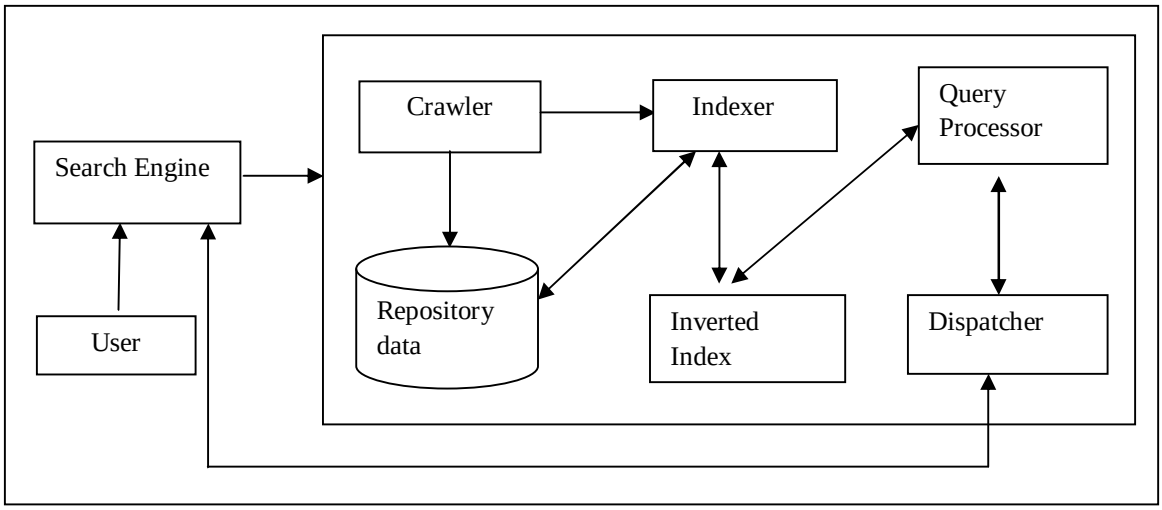

Figure 1 : Search Engine based Information Retrieval System

Literature reveals that researchers had realized this problem and proposed mechanisms for detecting similar documents on the web [1, 2, 8, 10, 11], however till date no mechanism has been adopted as standard and users still suffer from redundant results by the SE. Literature review also highlighted this fact that mechanisms for detecting content similarity lack scalability, considering the ever increasing size of the WWW, number of documents and web pages keeps increasing exponentially. Detecting similarity of incoming documents with all earlier existing documents even in One SE index is a challenge in itself. Some researchers [5, 7] have used intelligent agents for extracting information from distributed heterogeneous sources on the web. Intelligent agents being autonomous components possessed with mobility, learning ability, cooperation, reactiveness and pro-activeness in their actions can help automate complex tasks.

They have widely been employed in web based applications such as e-commerce, semantic web applications, wireless sensor networks, wireless communications etc. And have proved to be beneficial. This provided us the motivation to design redundant document removal layer in the existing search engine architecture which will detect similar contents available from different sources, link them and will eliminate them from search results being returned to the end user. This layer comprises of intelligent agent for automation of this complex task. Next section provides details of our proposed mechanism.

3.2 PROPOSED FRAMEWORK

In this section we describe our proposed framework, in which we determine similarity of two documents, for this purpose we proposed a redundant document detection and removal framework (RD2RF) which is embedded as a layer between existing components of a search engine as shown in Figure 2 below.

This framework comprises of query processor agent (QPA), shingle generation agent (SGA) and shingle comparison agent (SCA) respectively. Figure 3 given below provides the high level view of this layer. Composition of the agents is as follows:

Shingle Generation Agent (SGA): This agent is responsible for taking documents or web pages from the inverted index and creating their shingle set. Where a shingle is a collection of n words from the documents [11] and shingle set of a document uniquely identifies a document.

Shingle Comparison Agent (SCA): This agent is responsible for comparing shingle set of one document with other documents of the same category in order to find similar documents and subset or supersets of existing documents. SCA groups similar documents and their subset and supersets together.

Query Processor agent (QPA): This agent is responsible for removing redundant results from the search results being returned to the user. QPA retrieves the suitable URLs corresponding to the query from inverted index and lists redundant URLs under one link to the user.

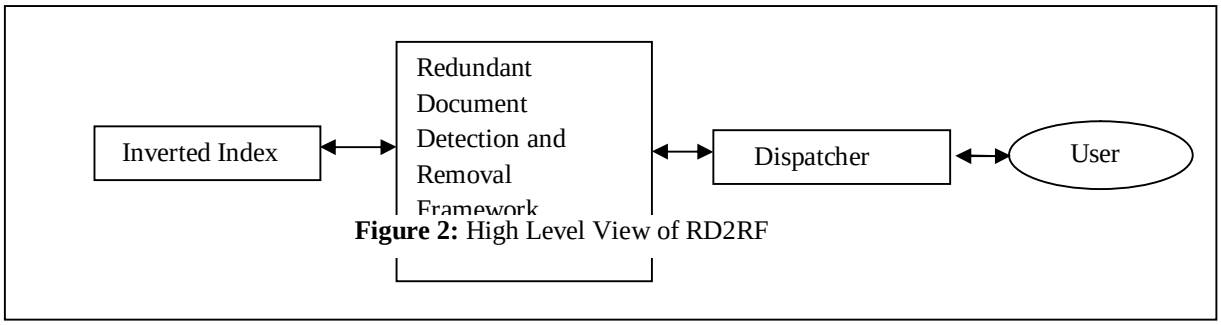

Figure 2: High Level View of RD2RF

Figure 3 given below provides detailed view of RD2RF

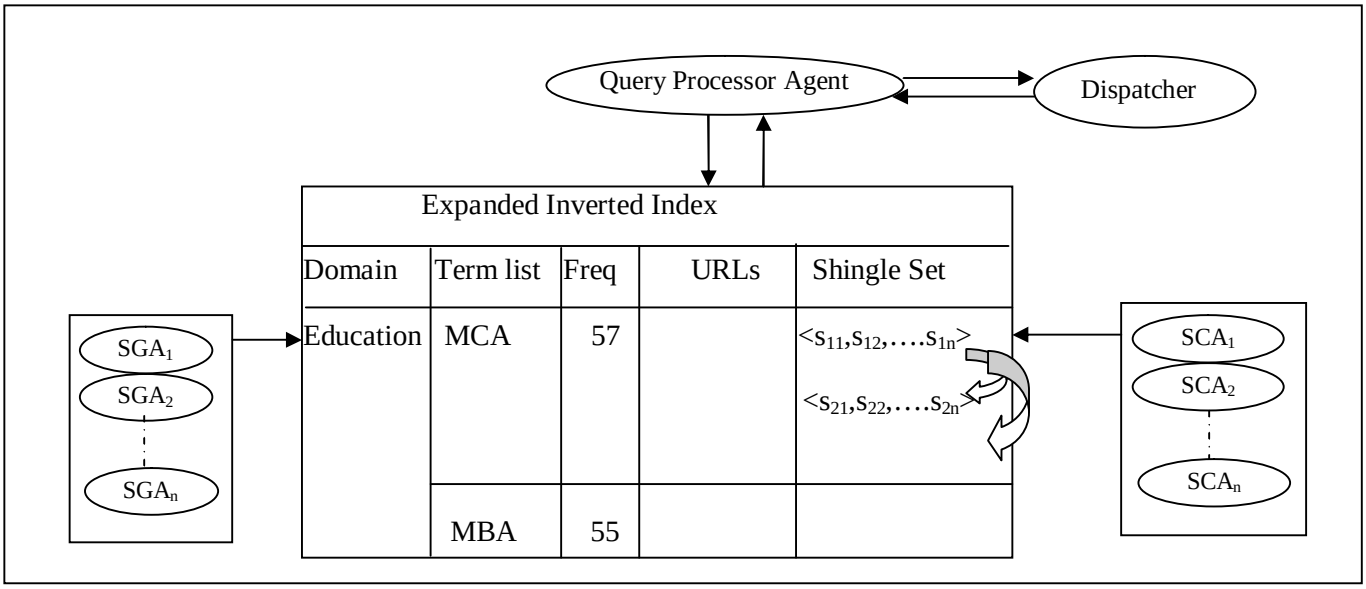

Figure 3 : Detailed View of Redundant Document Detection and Removal Framework

One SGA of RD2RF is being dedicated to one category of inverted index considering the large number of URLs available in each category. SGA assigned to each category is responsible for picking up listed URLS one by one and creates its shingle set and places it in the inverted index. One SCA is also dedicated to each category in inverted index responsible for comparing shingle sets and finding the URLs containing similar contents. SCA picks shingle set of first URL in a category and compares it with shingle sets of all other URLs one by one. For estimating similarity of documents it uses similarity and containment measures defined by Broder et.al [11]. Using shingle set of two documents their similarity is defined as number of distinct shingles appearing in both document divided by the total number of shingles in two documents, equation (1) given below provides the same:

Similarity=

$$\frac{Shingle\_set(URL_i) \text{ I } Shingle\_set(URL_j)}{\sum Shingle\_set(URL_i) + Shingle\_set(URL_j)} \quad \text{.....(1)}$$

If similarity value is greater than 0.75 documents are considered as similar.

SCA also calculates containment [11] of a document in other document so as to find subsets of an already existing document for grouping them together.

Containment=

$$\frac{Shingle\_set(URL_i) \text{ I } Shingle\_set(URL_j)}{Shingle\_set(URL_i)} \quad \text{.....(2)}$$

In first pass SCA picks one URL and calculates its similarity and containment value with rest of the URL in same category. In second pass SCA groups all documents having similarity value more than 0.75 and containment value 1 with the URL under consideration. Separate entries for similar or contained documents will be removed so as to reduce size of the list and not to return them as separate links in query results.

Now whenever a query is received from user, dispatcher passes it to QPA which looks in the inverted index to find relevant results. URLs available from multiple sources would be returned as one link only, headed by the first URL in the group and remaining links would be displayed with indication that they contain same document. Any user interested in exploring all links may do so, but for most of the users this information in search results would save lot of time and efforts.

Next subsection provides flow diagram and algorithms of various agents involved

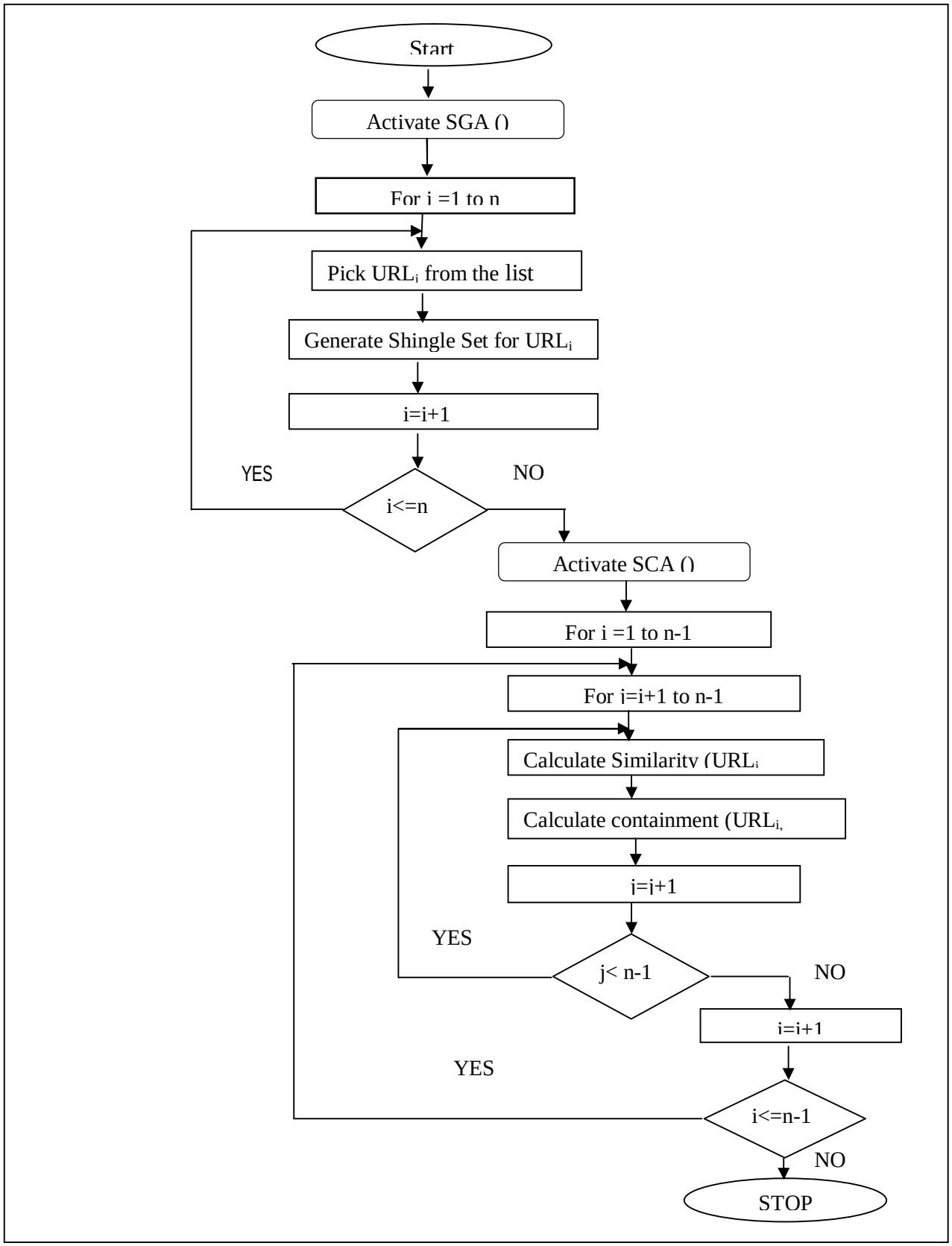

Figure 4: Flow Chart of Proposed Work

Algorithms of various agents are given in figure 5 (a), 5(b) and 5(c) respectively

```
Shingle_ Generating_ Agent ()
Input:  URL from Inverted Index;
Output: Generate shingle set
Action: activate, sleep;
{On input
Activate ();
for i=1 to n
{
Generate shingle_ set (URL_i);
Update inverted_index;
i=i+1;
}
Sleep ;}
```

Figure 5(a) : Shingle_Generating_Agent()

```
Query_Processor_Agent ()
 input: query from user;
  Output: relevant URLs;
{On input
Activate ()
{
   Search inverted index for keywords;
   Return (List_of_URLs);
}
Sleep ;}
```

Figure 5(c): Query_Processor_Agent

```
Shingle_Comparison_ Agent ()
Input:  Shingle set ;
Output: similar documents;
Action: activate, sleep;
{On input
Activate ();
for  i= 1, n-1
{    for j= i+1, n-1
    {Similarity= S(URL_i) ∩ S(URL_j) / ∑(S(URL_i) + S(URL_j)) ;
          If (similarity >0.75)
      {
        S(URL_i )≡S(URL_j);
        Group (URL_j, URL_i);
      }
      Containment = S(URL_i) ∩ S(URL_j) / S(URL_i) ;
    if(containment==1)
  {URLj ⊂ URL_i;
    Group (URL_j ⊂ URL_i);
   }
   Updated inverted index;
j=j+1;
}
i=i+1;
}
Sleep;
}
```

Figure 5(b): Shingle_Comparison_Agent

4 CONCLUSIONS

Web documents are being replicated on different servers. In many cases, these documents are near copies of other documents. These replicated documents get indexed in search engine indexes as separate documents and are listed in the search results multiple times. This replication reduces search efficiency and leads to dissatisfaction in end users. This work has proposed an agent based intelligent mechanism for detecting and removing redundant documents from search results. Being agent based this mechanism is scalable and can work well with large indexes available with search engines. It will lead to improved search efficiency and user satisfaction by providing unique search results.

REFERENCES

[1] Narayanan S, Hector G-M, "Finding Replicated Web Collection", Published in SIGMOD Conference 2000: Pages 355-366.

[2] Yaniv B Justin Z, "Redundant Documents and Search Effectiveness", In Proceeding of the 14th ACM International Conference on Information and Knowledge Management page 736-743

[3] Yaniv B, Justin Z, "The Case of the Duplicate Documents Measurements, Search and Science", In Proceedings of the 8th Asia-Pacific Web Conference on Frontiers of WWW Research and Development pp, 26—39.

[4] Masaru K, Masashi T, "What's really on the web? Identifying New Pages from a Series of Unstable Web Snapshots", In Proceedings of the 15th international conference on World Wide Web, WWW 2006, pp 233-241.

[5] Arvind A, Jungho C , Hector G-M, Andras P and Sriram R, "Searching the Web", Published in Journal ACM Transactions on Internet Technology, Vol. 1, No. 1, August 2001, Pages 2–43.

[6] Keith L.C and Vasilions S.L, "A Multi Agent System for distributed Information Retrieval on the World Wide Web", In Proceedings of the 6th Workshop on Enabling Technologies (WET-ICE '97), Infrastructure for Collaborative Enterprises, 18-20 June 1997,pp 87-93.

[7] Brain B, Robert G,Katsuhiro M, David-K,George C and Daniela R, "Mobile Agents in Distributed Information Retrieval" ,Published in Book Intelligent information Agents(1999). Chapter 15, pages 355-395.

[8] Junghoo C and Hector G-M, "The Evolution of the Web and Implications for an Incremental Crawler", In Proceedings of the Twenty-sixth International Conference on Very Large Databases, pages 200-209, New York, ***2000*** Available at http://www-diglib.stanford.edu/cgi-bin/get/SIDL-WP-1999-0129.

[9] B.Rajeshwar, A.Saravanar and G.Geetha, "Secure Information Retrieval Using Mobile Agents". International Conference on Computing and Control Engineering (ICCCE'2012), 12-13 April 2012.

[10] Narayanan S, Hector G-M, "Scam A Copy Detection Mechanism for Digital Documents". In proceeding of 2nd International Conference in Theory and Practice of Digital Libraries (DL 1995), June 11-13, 1995, Austin, Texas

[11] A.Z. Broder, S.C. Glassman, M.S. Manasse, and G. Zweig, "Syntactic Clustering of the Web". In Proceedings of the 6th World Wide Web Conference, pages 1157–1166, 1997.

[12] Martin T, Jonathan S and Andreas P: "Spotsigs:Robust and Effiecent Near Duplicate Detection in Large Web Collections", In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Singapore, pp. 563-57.

[13] J Prasanna K, Govindarajul P, "Duplicate and Near Duplicate Documents Detection: A Review". Published in European Journal of Scientific Research ISSN 1450-216X Vol.32 No 4(2009), pp.514-527.

[14] ZivBar-Y,Idit K and Uri S, "Do Not Crawl in the DUST: Different URLs with similar Text", In Proceedings of the 15th International Conference on World Wide Web .Pages 1015-1016.

[15] Jyodip Datta, "Ranking in Information Retrieval". April 2010. Available at http://www.cse.iitb.ac.in/internal/techreports/reports/TR-CSE-2010-31.pdf

[16] Ahmad M. Hannah., 2006,"A New Filtering Algorithm for Duplicate Document Based on Concept Analysis",Published in Journal of Computer Science, Vol. 2, No. 5, pp. 434-440.

[17] Jeffrey D, Monika R.H ," Finding related pages in the World Wide Web",In Proceeding of the 8th International World Wide Web Conference (WWW), PP 1467-1479.

[18] Junghoo C, Hector G-M and Lawrence P, "Efficient Crawling Through URL Ordering", In Proceeding of 7th World Wide Web Conference, PP 161-172.

[19] Narayanan S and Hector G-M, "Finding Near Replicas On The Web", Published in International Workshop on the Web and DataBase'98 Valencia, Spain March 27-28 ,1998 PP 204-212.

[20] Narayanan S,Hector G-M, "Building a Scalable and Accurate Copy Detection Mechanism" ,Published In Proceeding of Ist ACM Conference on digital Libraies Pages 160-168 ,Bethesda Maryland,March 1996

.