

# Optimization of Search Results with De-Duplication of Web Pages In a Mobile Web Crawler

Monika<sup>1</sup>, Mona<sup>2</sup>, Prof. Ela Kumar<sup>3</sup>

Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University, Delhi

[monika09.1992@gmail.com](mailto:monika09.1992@gmail.com) , [monali.mona08@gmail.com](mailto:monali.mona08@gmail.com) , [ela\\_kumar@rediffmail.com](mailto:ela_kumar@rediffmail.com)

**Abstract**— Being in an information era, where search engines are the supreme gateways for access of information on web. The efficiency and reliability of search engines are significantly affected by the presence of large amount of duplicate content present on World Wide Web. Web storage indexes are also affected by the presence of duplicate documents over web which leads to slowing down of serving results with high costs. Search results which consist of a collection of identical data takes seek time of user for finding relevant information. Duplicate contents on World Wide Web are present in multiple instances like data present on mirror sites, search results showing different hostname in urls but containing same content and many more. In this paper for escalation of search results a technique is being proposed in a mobile web crawler which aims to eliminate duplicate web pages in a respective domain in order to improve search engine efficiency where user will be able to get relevant results with respect to its query without letting wastage of network bandwidth and decrease the load on remote host which serves such web pages in order to provide low storage cost in comparatively less amount of time. The proposed result optimization technique is supposed to enhance the search engine effectiveness to a large scale.

**Keywords**— Crawler, Duplicate Documents, Web index, mirror sites, url, world wide web, search engine.

## INTRODUCTION

World Wide Web is a complete information system on the internet which is having global database containing comprehensive indices of documents for providing relevant information with respect to users. There is a presence of enormous collections of duplicate documents over World Wide Web, these documents are being uploaded on daily routines. Duplicate web pages refer to the type of contents which are identical in nature but present in multiple exemplars. . It has been reported that about 10% hosts are mirrored to various extents in a study including 238,000 hosts[5]. A systematic identification and detection of duplicate documents over web is a major issue that has raised due to the uploading of billions of web pages in every seconds. Detection of duplicate web pages is an intrinsic problem for improving the performances of search engines.

According to Google matt's around 25-30% of content over web are duplicative in nature and its okay to have that percent of replicate documents [1].Problems arises when we found pages having exact same content corresponding with different hostnames or the web pages having similar hostnames but crawls more than one time which seeks time of user.For retrieval of any kind of information from the search engines we need a web crawler which navigates from one page to another through hyperlinks for collecting data from web pages. Crawler interacts with billions of hosts and retrieve the pages continuously to keep the index up-to date.[3] Web crawlers are basically used for crawling over hyperlinks in a web page and then store those pages into web indexes but if duplicate documents present on web pages, web crawler will automatically crawl each page and will store pages in web indexes which results in flooded web documents having same content which consumes additional network bandwidth along with more amount of time and will able to provide only less satisfactory results.

This paper aims to eliminate the presence of duplicate web pages in a mobile web crawler, as in mobile crawlers the method of selection and filtration of web pages can be done at servers rather than search engine site which can reduce network load caused by web crawlers[4].Detection of Duplicate web pages will be done by checking the URI(uniform resource identifier- which is used to identify any object over the web) patterns as URI patterns are the only way to communicate with hostnames. Thus, Elimination of unnecessary flooded data will save network bandwidth and decrease load on remote host.

This paper consists of literature survey in the upcoming section followed by proposed work, Methodology behind the work and wrapping it up with conclusion and outcomes of the implemented work.

## LITERATURE REVIEW

This section consists of literature survey on duplicate content present over web. As because of the presence of duplicate types of documents on the web there is a degradation of performance of search engines which affects ranking mechanism of search engines with respect to any related search query. There are several techniques which have been discussed for detection of duplicate documents in order to improve search engine optimization mechanism. *Detection and Elimination of Near-Duplicates* - Works on near-duplicates detection and elimination are many in the history. In general these works may be broadly classified into Syntactical, Semantic based and URL based approaches[6].

### 1. Syntactical Approaches-

One of the earliest was by Broder et al [6], proposed a technique for estimating the degree of similarity among pairs of documents, known as shingling, does not rely on any linguistic knowledge other than the ability to tokenize documents into a list of words, i.e., it is merely syntactic. In this, all word sequences (shingles) of adjacent words are extracted. If two documents contain the same set of shingles they are considered equivalent and can be termed as near-duplicates.

### 2. Semantic Approaches-

A method on plagiarism detection using fuzzy semantic-based string similarity approach was proposed. The algorithm was developed through four main stages. First is pre-processing which includes tokenization, stemming and stop words removing. Second is retrieving a list of candidate documents for each suspicious document using shingling and Jaccard coefficient [6].

### 3. URL Based Approaches-

A novel algorithm, Dust Buster, for uncovering DUST (Different URLs with Similar Text) was intended to discover rules that transform a given URL to others that are likely to have similar content. Dust Buster employs previous crawl logs or web server logs instead of probing the page contents to mine the dust efficiently. Search engines can increase the effectiveness of crawling, reduce indexing overhead, and improve the quality of popularity statistics such as Page Rank, which are the benefits provided by the information about the DUST [6].

DE-DUPLICATION of web pages using URL Based Approach includes:

#### URL Preprocessing:

Tokenization is performed on URLs to generated a set of <key, value> pairs.

Tokenization is of two types:-

- a) Basic Tokenization involves parsing the URL according to RFC 1738(formally defined relative and absolute URLs, refined the general URL syntax, define how to resolve relative URL to absolute form) and extracting the tokens[16].
- b) Deep Tokenization involves host-specific learning, host specific delimiters given the set of URLs from a host[16].

In this paper URL Based approach is being used to detection and elimination of duplicate documents over web. Several kind of work have already been done on URL based approach which includes-

In[10], web page detection is done using set of techniques to mine rules from URLs string without fetching content explicitly.it consists of mining of crawl logs.Crawl logs tracks information about the status of crawled content which allows to verify whether crawled content was added to the index successfully or not. For manipulating crawl log ,Crawl log filter object is used.[10].Clusters of similar pages are utilized which includes page clustering and URL clustering for extracting transformation rules for normalization of URL.In[11] DUST algorithm is used for discovering substring substitution rules, which are used to transform URLs of similar content to one canonical URL along with the involvement of session IDs. In DUST algorithm URLs are first tokenized based upon specified generic delimiters to form components. Components are then tokenized using website specific delimiters.In[12]for detection of duplicate documents two approaches are used: Charikar's finger printing technique for which hamming distances of different bits need to be calculated. Algorithmic technique for identifying existing f-bit fingerprint that differ from a given fingerprint in at most k bit-positions, for small k.In[13] "Slice & Dice" generation of web pages, finding techniques are used, where pages are automatically generated by stitching together phrases drawn from a limited corpus. These techniques have been applied on two data sets 151 and 96 million web pages respectively. On first data set Breadth First Control is used and on second data set HTML Pages chosen at random from a large crawl conducted by MSN research.Rabin printing approach is used in[13] according to which functions treats the bits of an input string as the coefficients of a Boolean polynomial. There are many different Rabin functions each of which is parameterized by a primitive polynomial over the ring of Boolean.In[5] Data Detection algorithm is used for de-duplication of web pages on usage set of data, Algorithm works offline on the basis of favored user queries found by pre-mining the logs with query clustering.In this method there will be a detection of duplicates and near duplicates in an offline mode, while there elimination can be performed online by the search engines.Query logs are pre-mined by applying the query clustering technique and discovered query clusters are in turn

utilized for finding duplicate page. In [14] in order to find near duplicates of and input web page from a huge repository TDW matrix based algorithm is used with three phases- Rendering, Filtering, Verification Which receives an input web page and a threshold in its first phase, prefix filtering and positioning filtering to reduce size of record set in second phases and returns optimal set of near duplicate web pages in the verification phase by using minimum weight overlapping method(MWO).

### PROPOSED WORK FOR IMPLEMENTING A MOBILE WEB CRAWLER

In the proposed work there is a implementation of the technique through which elimination of duplicate web pages in a mobile crawler will be done.

In fig.1, A local server machine is being setup using Mysql in Xampp ( a free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages) and web server software is loaded on it and our local server machine makes its services available to internet using 8000 port. Mobile web crawler is implemented using mobile agent that makes use of IBM Java Aglets for crawling. Mobile crawler allows search engine to send a representative of the search engine i.e. an aglet to the data source.

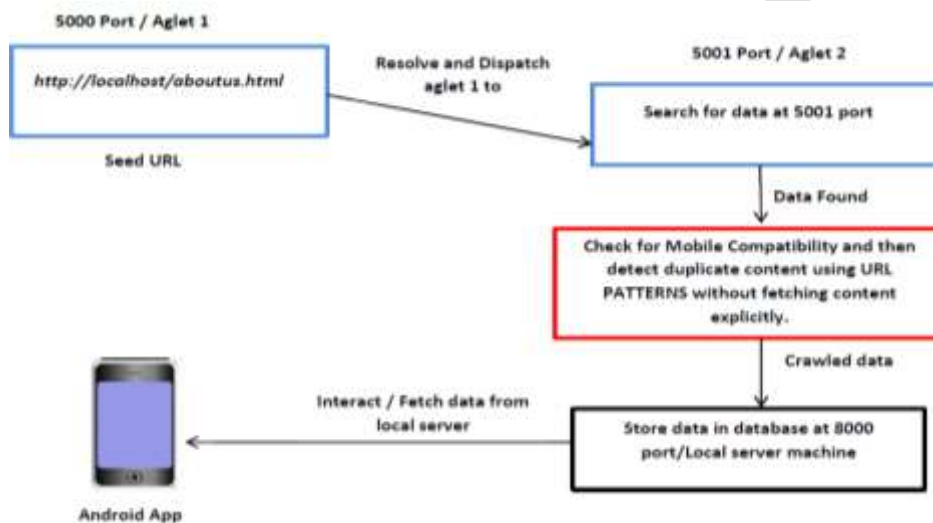


Figure 1. Architecture for Implementing Mobile Crawler

In our application the programmer instruct the crawler to migrate from web server at 5000 port to a web server at 5001 port in order to collect the relevant data. The purpose of this specialized mobile web crawler is to provide high quality searches in academic domain and provides the data to be viewed on mobile terminals i.e. data must be mobile compatible. Another feature of this mobile crawler is to check the presence of duplicate web pages using URLs patterns without fetching the content explicitly. The relevant data crawled by the mobile crawler depending on the crawler specifications is stored in the local server available at 8000 port. The data is for viewing on mobile terminals, an android mobile application working as a client fetch data from local server available to the internet at 8000 port.

### METHODOLOGY FOR DETECTION OF DUPLICATE WEB PAGES –

In this methodology Duplicate web page detection in a mobile web crawler will be done using set of techniques which analyses URI(is a compact sequence of character which identifies an abstract or physical resource[17]) Patterns in order to eliminate duplicate web pages without fetching the content explicitly, as URI are the only way to communicate with the domain names. Mobile crawler has been used as it is not possible to filter page in traditional web crawlers. For checking mobile compatibility for web pages VIEWPORT is used, which is use to distinguish size or screen of Desktop version from Mobile Version. After testing Mobile Compatibility of web pages, a technique is introduced to check Duplicate Web pages in Mobile crawler.

Detection and Elimination of duplicate HTML web pages in a Mobile web crawler will be done in following two manners:

1. If there will be a presence of same kind of hostnames in URLs, detection technique will detect them and mobile crawler will only crawl a single URL from them for maintaining the storage of web indexes and system load unnecessary.
2. If two different Hyperlinks are having the same content then system will crawl only a single URL from them which consume network bandwidth and will relate with customer satisfaction.

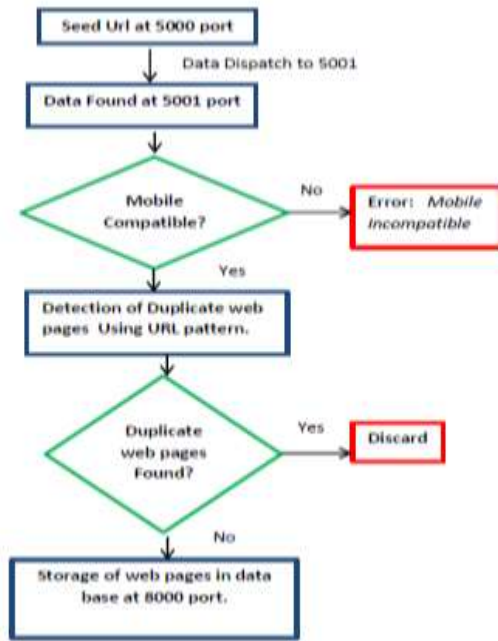


Figure 2: Workflow for Detection of duplicate documents in Mobile Web Crawler

In figure 2 there is work flow for reduction of duplicate web pages. Firstly seed URLs will be given at 5000 port and if data is found at 5001 port, Test of mobile compatibility will be done. If pages are found mobile compatible then they will further go for a test of duplicate web page detection using URL patterns, and if pages are not mobile compatible crawler will not Mobile crawl them and shows Mobile Compatibility error. After that Duplicate web page detection will be done using set of rules which discussed above and then there will be a storage of those web pages in database at 8000 port which is our local machine. The proposed Search results optimization technique is supposed to enhance performance and efficiency of a Mobile crawler to a large scale.

### Implementation and Results

In this Paper, the design and working of a web crawler for searching mobile compatible data is presented. This web crawler is implemented using mobile agent which is an autonomous software agent.

- First a local server machine is setup.

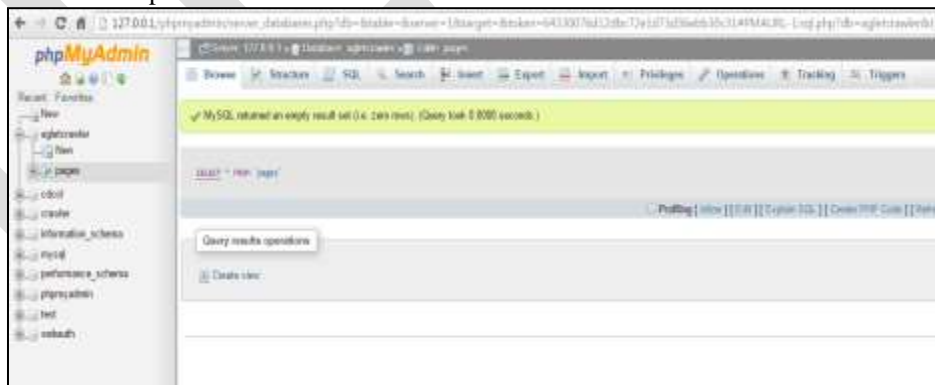


Figure 3: Local server is setup

- Then we run an aglet application known as Tahiti.

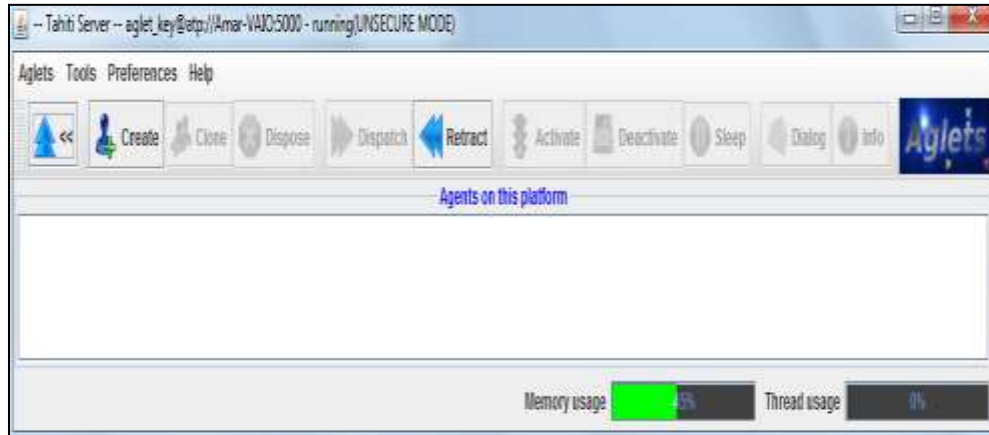


Figure 4: Aglet application Tahiti running at port 5000.

- We can run multiple servers (Tahiti) on a single computer by assigning them different ports. Another server running at port 5001.



Figure 5: Aglet application Tahiti running at port 5001.

- Create an aglet- Aglet creation is done by using the tab button named create.

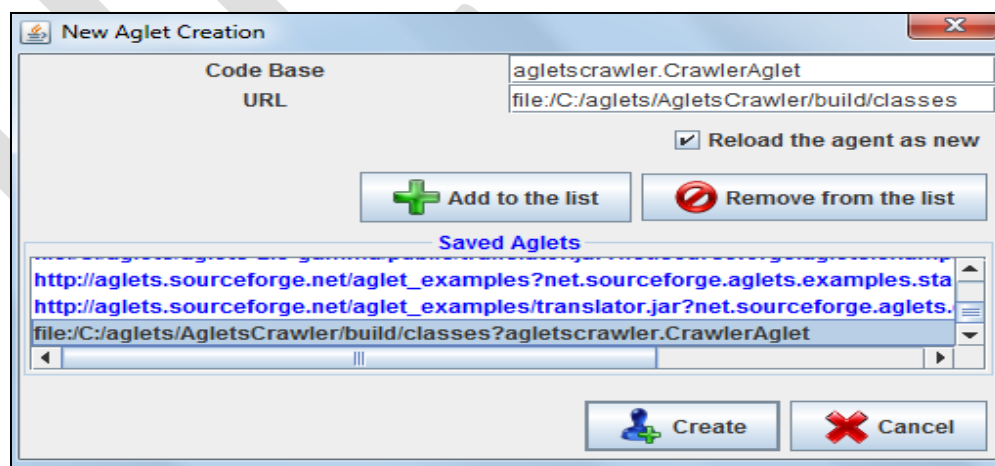


Figure 6: Aglet Creation

- Enter the seed URL and dispatch it- “Seed URLs” is the initial entry point for any crawler from where it starts crawling. An Academic site of java tutorials has been designed in which there is a presence of duplicate documents over different links for seed urls this academic site’s starting point has given.

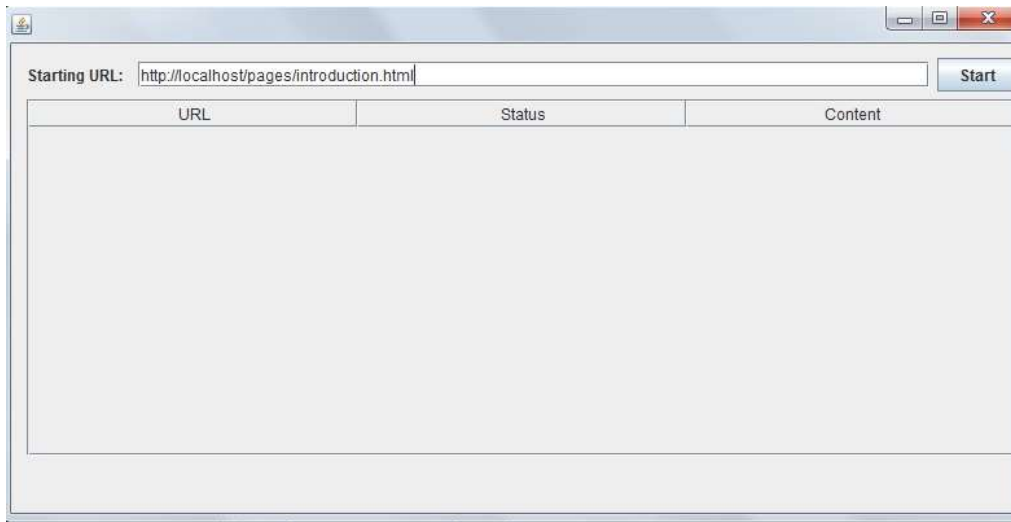


Figure 7: User Interface to enter seed url.

- Programmer instructs the crawler to migrate from web server at 5000 port to a web server at 5001 port.
- Dispatch- In this aglet migrate from one platform to the another platform reducing network load and saves network bandwidth.



Figure 8: Aglet dispatched.

- Here we can see that a list of fetched web pages is created which is of an HTML site containing total of 26 links and due to detection of duplicate web pages of this web crawler it is crawling only 7 links out of 26 which are unique in terms of content and in terms of domain names as well.

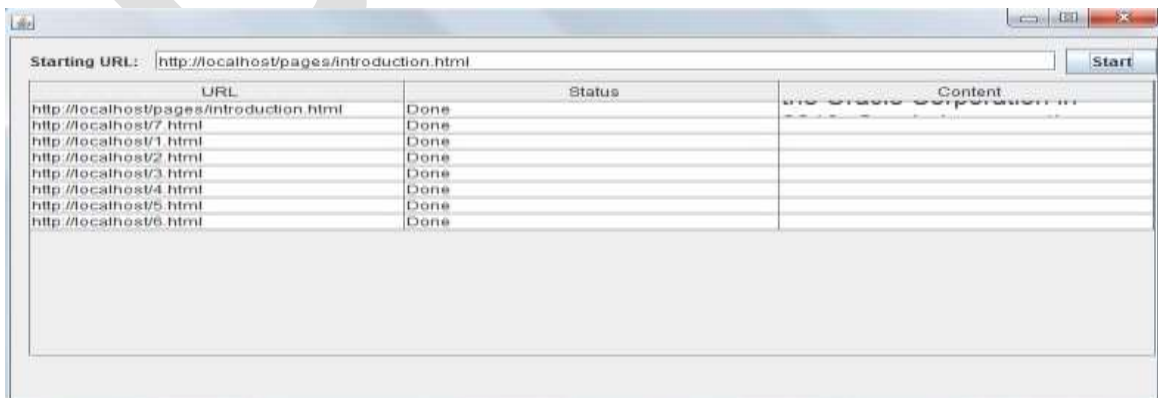
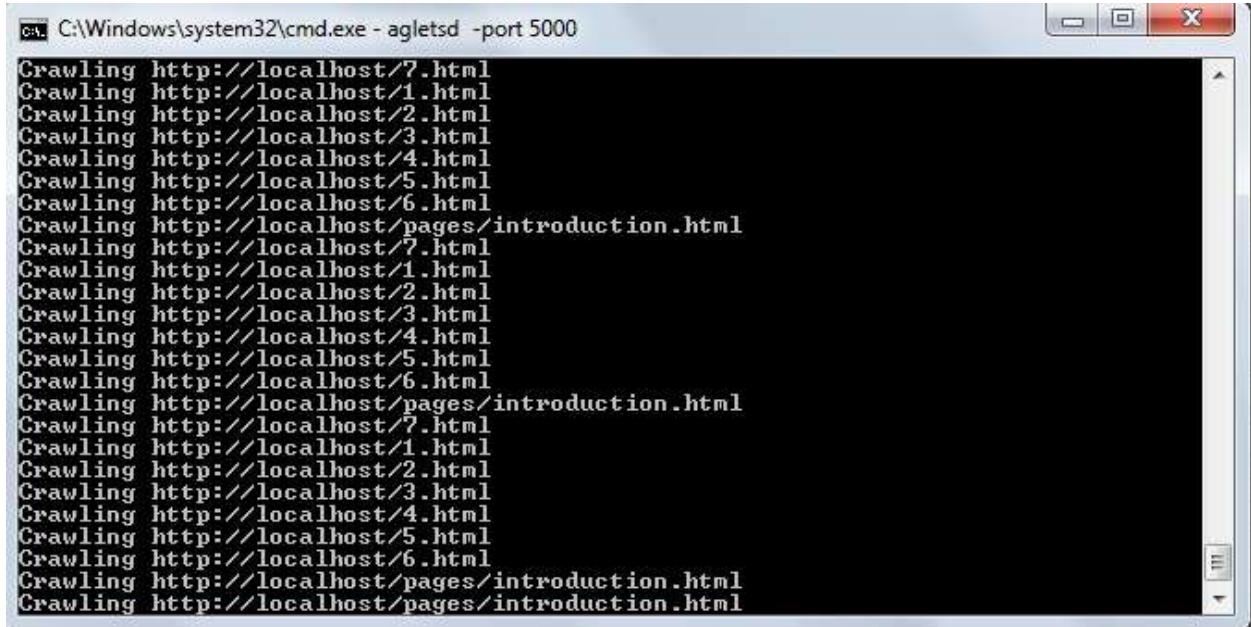


Figure 9: list of fetched web pages URLs along with the content

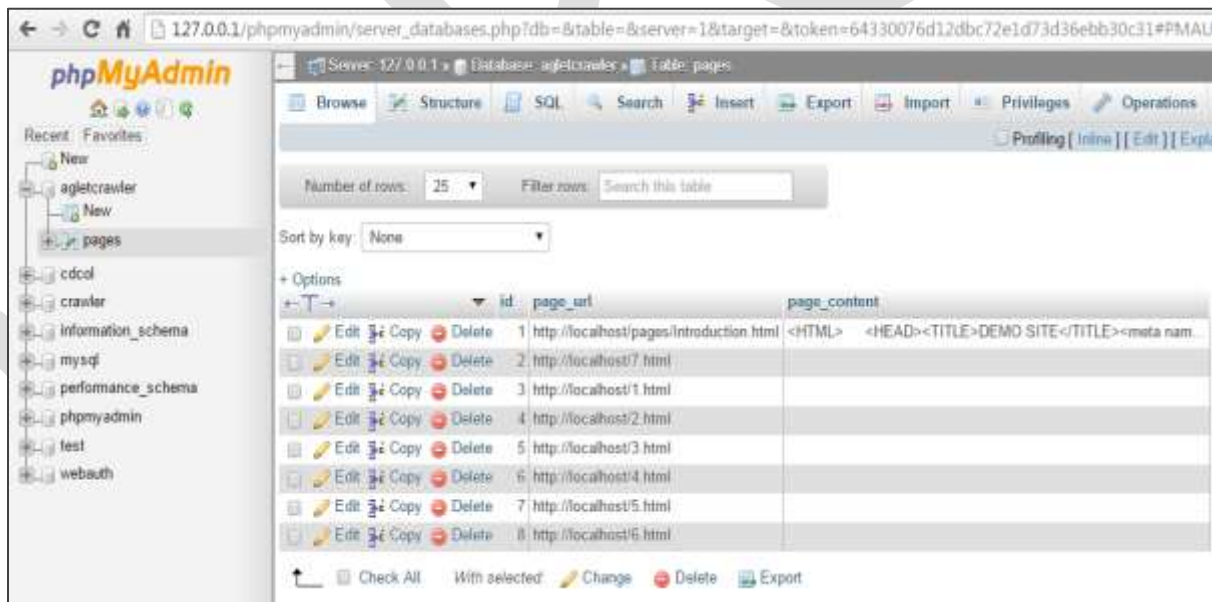
- Crawled links brought back to original host by mobile Agent.



```
C:\Windows\system32\cmd.exe - agletsd -port 5000
Crawling http://localhost/7.html
Crawling http://localhost/1.html
Crawling http://localhost/2.html
Crawling http://localhost/3.html
Crawling http://localhost/4.html
Crawling http://localhost/5.html
Crawling http://localhost/6.html
Crawling http://localhost/pages/introduction.html
Crawling http://localhost/7.html
Crawling http://localhost/1.html
Crawling http://localhost/2.html
Crawling http://localhost/3.html
Crawling http://localhost/4.html
Crawling http://localhost/5.html
Crawling http://localhost/6.html
Crawling http://localhost/pages/introduction.html
Crawling http://localhost/7.html
Crawling http://localhost/1.html
Crawling http://localhost/2.html
Crawling http://localhost/3.html
Crawling http://localhost/4.html
Crawling http://localhost/5.html
Crawling http://localhost/6.html
Crawling http://localhost/pages/introduction.html
Crawling http://localhost/pages/introduction.html
```

Figure 10: Crawled links brought back to the original host

- The relevant data crawled by the mobile crawler is stored in the local server



id	page_url	page_content
1	http://localhost/pages/introduction.html	<HTML> <HEAD><TITLE>DEMO SITE</TITLE><meta nam...
2	http://localhost/7.html	
3	http://localhost/1.html	
4	http://localhost/2.html	
5	http://localhost/3.html	
6	http://localhost/4.html	
7	http://localhost/5.html	
8	http://localhost/6.html	

Figure 11: List of fetch URLs stored in local server.

- An android mobile application working as a client fetch data from local server

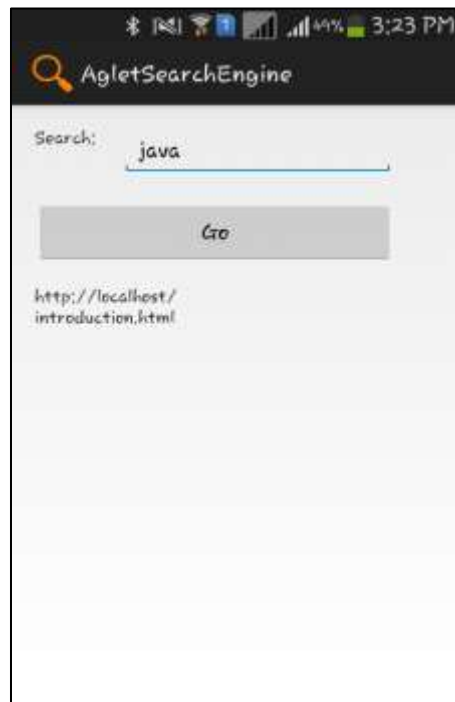


Figure 12: Screenshot of fetching data from local server on Real Device.

## CONCLUSION

An eruptive explosion of World Wide Web containing duplicate web pages posed a significant challenge for web crawling. Presence of enormous amount of duplicate content has become a major challenging task for information retrieval with respect to user's query. The proposed architecture will play a notable role for addressing this problem. Major objective of the proposed system is to develop a mobile crawler which provide Detection of similar web documents, similar sentence in any two web documents which sums up with collection of web pages containing relevant results in order to maintain efficient qualities of web indexes which allows to decrease load on remote host that serve such web pages. Proposed method solves the difficulties of information retrieval from the mobile crawler for a respective domain. The approach has detected the duplicate web pages efficiently based on mining rules from URIs strings without fetching the content explicitly. The detection of duplicate web pages will results in increasing the search queries qualities.

## REFERENCES:

- [1] <http://searchengineland.com/googles-matt-cutts-25-30-of-the-webs-content-is-duplicate-content-thats-okay-180063>
- [2] <http://moz.com/learn/seo/duplicate-content>
- [3] R.G Tambe and M.B Vaidya, "Implementing Mobile Crawler Using JINI Technology to Reduce Network Traffic and Bandwidth Consumption", International journal of engineering and advanced technology(IJEAT), Feb 2013.
- [4] Fiedler J. and Hammer J, "Using Mobile Crawlers to Search the Web Efficiently", International Journal of computer and information science, 2000.
- [5] A.K Sharma and Neelam Duhan, "Optimisation of Search Results with Duplicate Page Elimination using Usage Data", ACEEE ,April 2011.
- [6] Y. Syed Musadhir, J. Deepika, S. Sindhikumar and G.S Mahalakshmi, "Near Duplicates Detection and Elimination Based on Web Provanance for Effective Web Search", International Journal on Internet and Distributed Computing System, 2011
- [7] C. Olston and M. Najork, "Web Crawling Foundations and Trends in Information Retrieval", 2010.



- [8] T. Lei, R. Cai, J.-M. Yang, Y. Ke, X. Fan and L. Zhang. "A pattern tree-based approach to learning URL normalization rules", WWW, pages 611-620, 2010.
- [9] <http://download.java.net/jdk7/archive/b123/docs/api/java/net/URI.html>
- [10] Hema Swetha kappula, Krishna P. Leela, Amit Agarwal, "Learning URL Patterns for web page De-Duplications", ACM, Feb 2010.
- [11] Amit Agarwal, Hema Swetha kappula, Krishna P. Leela, "URL Normalisation for De-Duplication of web pages", ACM 2009
- [12] Gurmeet Singh Manku, Arvind Jain and Anish Das Sarma, "Detecting Near Duplicates for Web Crawling", ACM 2007
- [13] Dennis Fetterly, Mark Manasse and Marc Najork, "Detecting Phrase-level Duplication on the World Wide Web", ACM 2005.
- [14] Shine N Das, Midhun Mathew and Pramod K Vijayraghvan "An Efficient Approach for Finding Near-Duplicates Web Pages Using Minimum Weight Overlapping Method", IEEE 2012
- [15] <http://searchsecurity.techtarget.com/definition/tokenization>
- [16] <http://www.protegrity.com/2011/11/the-difference-between-basic-and-modern-tokenization/>
- [17] [https://danielmiessler.com/study/url\\_vs\\_uri/](https://danielmiessler.com/study/url_vs_uri/)
- [18] <https://www.informatica.us.es/~ramon/tesis/agentes/Aglets1.0.3/doc/tahiti/tahiti.html>
- [19] A. Dasgupta, R. Kumar, and A. Sasturkar. De-duping urls via rewrite rules. In KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2012.
- [20] D. Fetterly, M. Manasse, and M. Najork. On the evolution of clusters of near-duplicate web pages. In LA-WEB '03: Proceedings of the First Conference on Latin American Web Congress, November 2012