Advances in Networks
2017; 5(1): 1-13
http://www.sciencepublishinggroup.com/j/net
doi: 10.11648/j.net.20170501.11
ISSN: 2326-9766 (Print); ISSN: 2326-9782 (Online)

Extracting Textual Information from Google Using Wrapper Class

A. Muthusamy[1], A. Subramani[2]

[1]Department of Computer Technology, N. G. P Arts and Science College, Coimbatore, India
[2]Department of Computer Science, Govt. Arts College, Dharmapuri, India

Email address:
muthusamy.arumugam@gmail.com (A. Muthusamy), subramani.appavu@gmail.com (A. Subramani)

To cite this article:
A. Muthusamy, A. Subramani. Extracting Textual Information from Google Using Wrapper Class. *Advances in Networks*. Vol. 5, No. 1, 2017, pp. 1-13. doi: 10.11648/j.net.20170501.11

Received: April 22, 2017; **Accepted**: May 11, 2017; **Published**: July 5, 2017

Abstract: In general, the web text documents are often structured, un-structured, or semi-structured format that is promptly growing everyday with massive amounts of data. The users provided with many tools for searching relevant information. Some of the searches include, Keyword searching, topic and subject browsing can help users to find relevant information quickly. In addition, Index search mechanisms allow the user to retrieve a set of relevant documents. Occasionally these search mechanisms are not sufficient. With the rapid development of Internet, amount of data available on the web regularly increased, which makes it difficult for humans to distinguish relevant information. A wrapper class is proposed to extract the relevant text information and focus on finding useful facts of knowledge from unstructured web documents using Google. Techniques from information retrieval (IR), information extraction (IE), and pattern recognition are explored.

Keywords: Information Extraction, Retrieval, Semantic Web, Web Search Engine

1. Introduction

Web mining is the use of data mining techniques to automatically discover and extract information from Web documents and services. Web content mining is the process of mining useful information from the contents of Web pages and Web documents, which are mostly text, images and audio / video files. Techniques used in this discipline have been heavily drawn from natural language processing (NLP) and information retrieval. Web structure mining is the process of analyzing the nodes and connection structure of a website through the use of graph theory. There are two things that can be obtained from this: the structure of a website in terms of how it is connected to other sites and the document structure of the website itself, as to how each page is connected. Web usage mining is the process of extracting patterns and information from server logs to gain insight on user activity including where the users are from, how many clicked what item on the site and the types of activities being done on the site.

At present, search engines are the primary gateways of information access on the Web. Today search engines are becoming necessity of most of the people in day to day life for navigation on internet or for finding anything. Search engine answer millions of queries every day. Whatever comes in our mind just enter the keyword or combination of keywords to trigger the search and get relevant result in seconds without knowing the technology behind it. The search keyword is search engine it returns 36 million results. In addition with this, the engine returned some sponsored results across the side of the page, as well as some spelling suggestion in 0.36 seconds. And for popular queries the engine is even faster. To engineer a search engine is a challenging task. Web crawler is an essential part of search engine. A web crawler is a program that, given one or more seed URLs, downloads the web pages associated with these URLs, extracts any hyperlinks contained in them, and recursively continues to download the web pages identified by these hyperlinks. Web crawlers are an important component of web search engines, where they are used to collect the corpus of web pages indexed by the search engine. Moreover, they are used in many other applications that process large numbers of web pages, such as web data mining, comparison shopping engines, and so on.

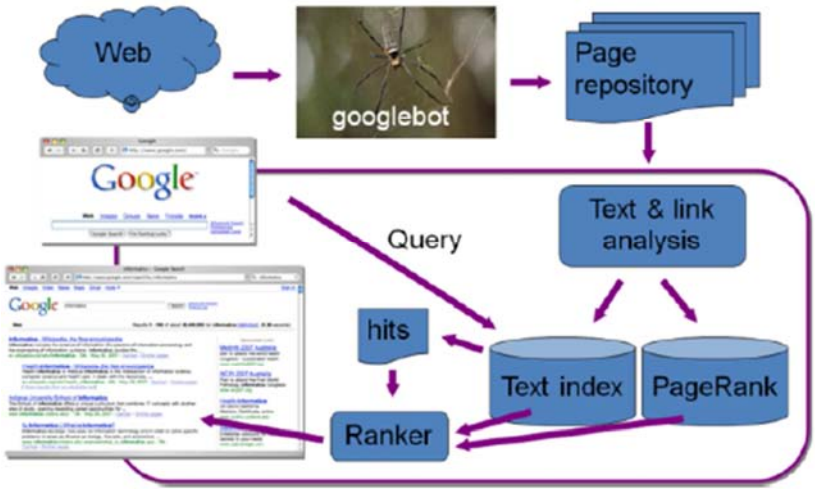

Figure 1. *Functioning method of Search Engine.*

Search engine is a tool that allows people to find information over WWW. It is a website use to look up web pages, like yellow pages for the Internet. A web search engine is a software system designed to search for information on the WWW. Search engines are constantly building and updating their index to the WWW. They do this by using "spiders" that crawled the web and fetch web pages. Then the words used in these webpages added to the index along with where the words came from [13]. A search engine operates in the following order as,

a. Web crawling
b. Indexing
c. Searching

Web search engines work by storing information about many web pages. These pages are retrieved by a Web crawler an automated Web crawler which follows every link on the site. The search engine then analyzes the contents of each page to determine how it should be indexed for example, words can be extracted from the titles, page content, headings, or special fields called meta tags as depicted in Fig. 1.

1.1. Google

Google launched officially on September 21, 1999 with Alpha and Beta test versions released earlier. Since then it has pushed through with its relevance linking based on pages link analysis, cached pages and a rapid growth. In June 2000 it announced a database of over 560 million pages and they moved up their claim up to 3 billion by November 2002. As of April 12, 2005 the number is 8,058 044 651 Web pages [16]. Google is implemented in C and C++ and some parts of it are written in Python. It is designed to avoid disk seeks whenever possible because a disk seek takes about 10 ms on average. To satisfy its storage needs it takes advantage of virtual files spanning across multiple file systems. Documents in the repository are compressed using zlib [19]. All in all, the authors of Google insist that a Web search engine is a very rich environment for research ideas [4]. Googlebot is Google's Web crawling robot written in C++ programming language. It collects documents from the web to build a searchable index for the Google search engine.

When a user enters a query into a search engine, the engine examines its index and provides a listing of best matching web pages according to its criteria, usually with a short summary containing the documents title and sometimes parts of the text. The index is built from the information stored with the data. From 2007 [17] search engine has allowed one to search by date by clicking "Show search tools" in the leftmost column of the initial search results page, and then selecting the desired date range. Most search engines support the use of the Boolean operators AND, OR and NOT to further specify the search query. Boolean operators are for literal searches that allow the user to refine and extend the terms of the search. As well, natural language queries allow the user to type a question in the same form one would ask it to a human. A web site like this would be ask.com. The usefulness of a search engine depends on the relevance of the result set it gives back. While there may be millions of web pages that include a particular word or phrase, some pages may be more relevant, popular, or authoritative than others. Most search engines employ methods to rank the results to provide the "best" results first. Search engines that do not accept money for their search results make money by running search related ads alongside the regular search engine results. The search engines make money every time someone clicks on one of these ads [14].

1.2. Google Database

a. Indexed Web pages are Web pages whose words have been indexed, i.e. some records have been made about what terms and how many times they occur on a specific page. Typically, the terms are sorted descending as in an inverted index.
b. Daily re-indexed Web pages are the same, except that Google re-indexes them "every day". These pages display the date they were last refreshed after the URL and size in Google's results.
c. Un-indexed URLs represent URLs for Web pages or

documents that Google's spider (Googlebot) has not actually visited and has not indexed.

d. Other file types are Web-accessible documents that are not HTMLlike Web pages, such as Adobe Acrobat PDF (.pdf), PostScript (.ps), Microsoft Word (.doc), Excel (.xls), PowerPoint (.ppt), Rich Text Format (.rtf) and others.

2. Overview

2.1. Web Crawling

Web crawling or spidering is the process of collecting Web pages and other Web documents by recursively following the out links from a set of starting pages. Its primary goal is to create a corpus of Web documents that could subsequently be indexed by a Web search engine in order to respond to user's requests. Every search engine relies on its indexed corpus and so the way of its creation is essential. The role currently played by Web search engines in the world is incontestable, and, therefore, it is somewhat surprising that crawling is still under-represented in the Web mining research. The experiment described in Section VII could not have been conducted without Web crawling techniques, so the researchers find useful to incorporate a section on this topic in this dissertation. Unless the researchers indicate another source of information, the facts presented here come from our own experience, the most comprehensive overview of Web crawling strategies ever by [22] or from the Web mining book [23, 24].

2.2. Web Crawler Architecture

In Fig. 2. it depicts the typical architecture of a large-scale Web crawler. By a large-scale crawler, itmeans a system capable of gathering billions of documents from the current World Wide Web. It is clear that with such a huge amount of data more sophisticated techniques must be apply than simply parsing HTML files and downloading documents from the URLs extracted from there. As the researchers observe at the Fig. 2., much attention is paid to the problems of avoiding Web pages (URLs) already visited before, parallelizing crawling (fetching threads) and balancing the load of Web servers from which documents are obtained (server queues), and speeding up the access to Web servers (via DNS caching).

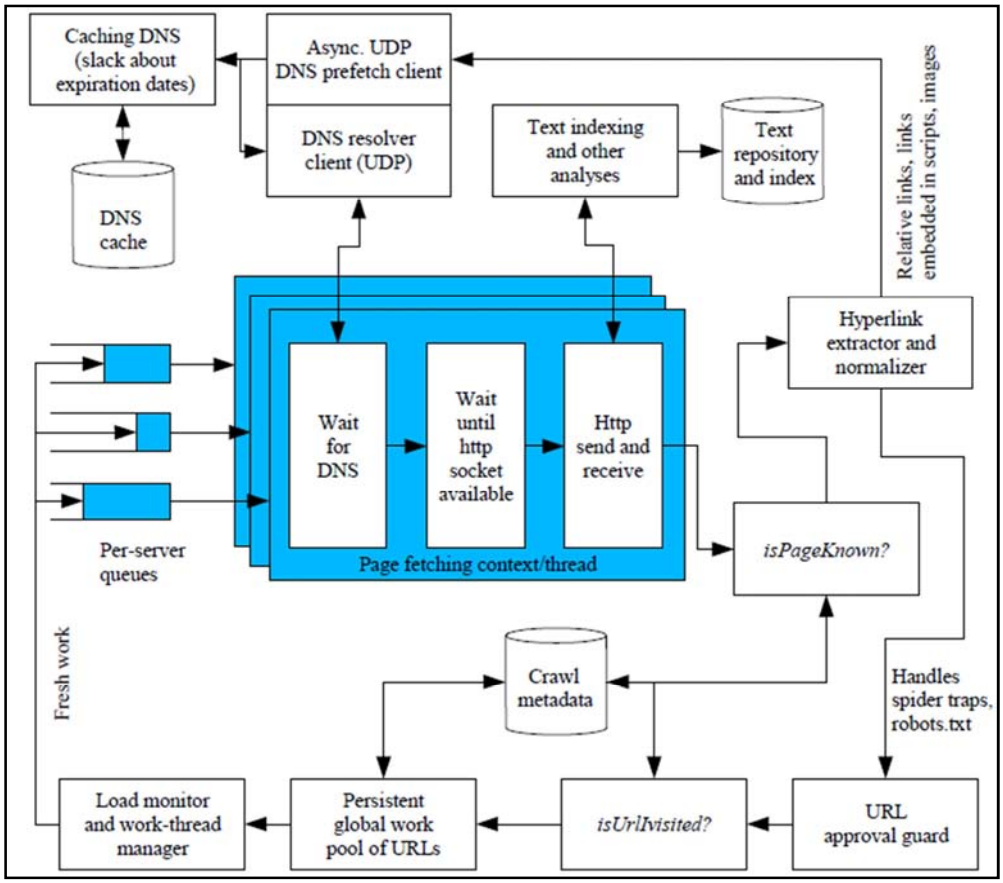

Figure 2. *Web crawler Architecture.*

2.3. Role of Web Crawler

In general, a Web crawler takes a URL from the queue of pending URLs, it downloads a new page from the URL, it stores the document to a repository and it parses its text to find hyperlinks to URLs, which it then en-queues in the queue of pending URLs in case they have not yet been downloaded ("fetched"). Ideally, crawling stopped when the queue of pending URLs is empty. In practice, however, this will never happen, as the universe of a large-scale Web

crawler is almost infinite. The Web steadily changing and will never be crawled as a whole. A reasonable terminating condition must be set up for the crawler to stop. For example, a certain number of documents have been fetched, a specific number of terabytes of data has been downloaded, a particular time period has elapsed, or the crawler simply runs out of resources (main memory, storage capacities, etc..,).

2.4. Internals

More specifically, a Web spider would like to do many activities in parallel in order to speed up the process of crawling. In fact, the DNS name resolving, i.e. getting IP address of an Internet host by contacting specific servers with name-to-IP mappings, and opening an HTTP connection to a Web server may take up to a second which is often more than receiving the response from a Web server (i.e. downloading a small or middle-sized document with a sufficiently fast connection). Therefore, the natural idea is to fetch many documents at a time.

The Current commercial large-scale Web robots fetch up to several thousands of documents in parallel and crawl the "whole" Web (billions of documents) within a couple of weeks. Interestingly, parallelization objects offered by operating systems such as processes and threads do not seem advantageous for multiple fetching of thousands of documents due to thread (process) synchronization overheads. Instead, a non-blocking fetching via asynchronous sockets is preferred. Indeed, present commercial search engines work with such huge amounts of data that they have to use technologies that are often beyond capabilities of traditional operating systems. Google, for example, has a file system of its own [28].

Implement of large-scale Web crawlers try to reduce the host name resolution time by means of DNS caching. The DNS server mapping host names to their IP addresses is customized and extended with a DNS cache and a pre-fetching client. The cache is preferably placed in the main memory for a very fast lookup in the table of names and IPs. In this way, server names that have already been put in the cache before can be found almost immediately. New names, though, have still to be searched for on distant DNS servers. Therefore, the pre-fetching client sends requests to the DNS server right after URL extraction from a downloaded page and does not wait until the resolution terminates (non-blocking UDP data grams are sent). Thus, the cache is filled up with corresponding IPs long before they are actually needed. (DNS requests are kept completely away from a common Web surfer. It is the Web browser that gets all the work done.)

2.5. Web Crawling Strategies

In [21, 22] it defines three groups of crawling strategies:

2.5.1. No Extra Information

When deciding which page to crawl next, the spider has any additional information available except knowing the structure of the Web crawled so far in the current crawl.

a. Breadth-first is reported to collect high quality (important) pages quite soon [25]. On the other hand, depth-first strategies were not much used in real Web crawling, also because the maximum crawling depth is worse controllable in them.
b. Back-link-count [26]. Pages in the frontier with a higher number of in-links from pages already downloaded have a higher priority of crawl.
c. Batch-PageRank [26]. This technique calculates PageRank values for the pages in the frontier after downloading every k pages these PageRank based on the graph constituted of the pages downloaded so far, and they are only estimates of the real PageRank derived from the whole Web graph. After each re-calculation, the frontier prioritized according to the estimated PageRank and the top k pages will be downloaded next.
d. Partial-PageRank It is like Batch-PageRank but with temporary PageRank assigned to new pages until a new re-calculation is done. These temporary PageRank are computed non-iteratively unlike normal PageRank as the sum of PageRank of in-linking pages divided by the number of out-links of those pages (the so-called out-link normalization).
e. OPIC [27]. This technique may considered as a weighted back link count strategy.
f. Larger-sites-first This method tries to manage best with the rule that Web sites must not be overloaded and choose preferentially pages from Web sites having a large number of pending pages. The goal is not to have at the end of the crawl a small number of large sites, because that would slower down crawling due to the delay required between two accesses to the same site.

2.5.2. Historical Information

The crawler additionally knows the Web graph obtained in a recent "complete" crawl. Although the Web changes very fast (about 25% new links are created every week [29]), the historical data were too costly to acquire so that it could be entirely neglected. Thus, the selection of a next page to crawl will be based on the historical information. Again, the researchers would like to order the pages in the frontier by their PageRank and crawl the more important ones first. For the pages encountered in the current crawl that existed when the last crawl was run, the researchers use their historical PageRank even though we are aware that the current PageRank may have changed. The pages that did not exist then have to be assigned some estimates. There are several methods how to deal with these new pages:

a. HistoricalPageRank-Omniscient Again, it is a theoretical variant which knows the complete graph and assigns "true" PageRanks to the new pages.
b. HistoricalPageRank-Random It assigns to the new pages random PageRanks chosen from those computed for the previous crawl.
c. Historical PageRank-Zero New pages are all assigned a zero PageRank and are thus crawled after "old" pages.

d. Historical PageRank-Parent Each new page is assigned an out-link-normalized PageRank of its parent pages linking to it. If a parent page is new the researchers obviously proceed to the grand parent and so forth.

2.5.3. Include All Information

This is a theoretical strategy; the researchers will call it the omniscient method, which perfectly knows the whole Web graph that should be crawled including the values of importance of individual pages. This method always chooses the page with the highest importance from the frontier.

2.6. Merits& Features of Web Crawler

Crawlers can also be used for automating maintenance tasks on a Web site, such as checking links or validating HTML code. Crawlers can be used to gather specific types of information from Web pages, such as harvesting e-mail addresses. Search engines frequently use web crawlers to collect information that is available on public web pages. They collect data so that when Internet surfers enter a search term on their site, they can quickly provide the surfer with relevant web sites. Linguists use web crawlers to perform a textual analysis. They comb the Internet to determine what words are commonly used today. Some of the features are,

a. Robustness
b. Politeness
c. Distributed
d. Scalable
e. Performance and efficiency
f. Quality
g. Freshness
h. Extensible

To remove these difficulties the web crawler is having the following policies [6].

a. A Selection Policy that states which page to download.
b. A Re-Visit Policy that states when to check for changes in pages.
c. A Politeness Policy that states how to avoid overloading web sites.
d. A Parallelization Policy that states how to coordinate distributed Web Crawlers.

2.7. Problem Identification

Finding relevant information in unstructured web document retrieve inconsistent data, it involve time-consumption and difficult task. The data is unknown in terms of structure and values. The characteristics of web make crawling complicated due to its huge Volume of data, and fast data rate change. The biggest task of a crawler is to avoid redundancy by eliminating duplicate pages and links from the crawl. Unfortunately, many problems arises Different forms of URLs, Too many URLs, Duplicate pages with different URLs, the biggest trouble is with dynamic pages such as CGI, PHP, or Java scripts. In addition, the text extraction includes noisy information like advertising, pictures, videos etc..,

3. Literature Review

In this section the researchers describe the different techniques with different authors which are related to the Information extraction and retrieval from the web. Patrick Mair and Scott Chamberlain [3] presented the Comprehensive R Archive Network (CRAN) Task View on Web Technologies. It describes the most important aspects of Web Technologies and Web Scraping and lists some of the packages that are currently available on CRAN. Finally, it uses to plot the network of Web Technology related package dependencies based on a scraping job where the researchers harvested all corresponding package dependencies (and imports) from CRAN. The Web Technologies Task View will be update on a regular basis, and therefore, the network plot will change accordingly.

Krishan Kant Lavania, Sapna Jain, Madhur Kumar Gupta, and Nicy Sharma [2] has reviewed today search engines are becoming necessity of most of the people in day-to-day life for navigation on internet or for finding anything. Search engine answer millions of queries every day. Whatever comes in our mind just enter the keyword or combination of keywords to trigger the search and get relevant result in seconds without knowing the technology behind it. To engineer a search engine is a challenging task. In this part Google is the most popular scalable search engine, and in-depth description of methods and techniques that the Google uses in searching. It employs a number of techniques or methods to improve search quality including page rank calculation, anchor text, and many other features.

Rama Subbu Lakshmi B, Jayabhaduri R [10] proposed a method would order the aliases based on their associations with the name using the definition of anchor texts-based co-occurrences between name and aliases in order to help the search engine tag the aliases according to the order of associations. The association orders would automatically discovered by creating an anchor texts-based co-occurrence graph between name and aliases. Ranking Support Vector Machine (SVM) used to create connections between name and aliases in the graph by performing ranking on anchor texts-based co-occurrence measures. The hop distances between nodes in the graph will lead to have the associations between name and aliases. The hop distances will found by mining the graph. The limitation of the proposed method is applicable only to a single document.

Singh, B. and Singh, H. K. [11] has Problems faced by Web Content mining such as extracting information from heterogeneous environment, the redundancy, the linked nature of the web, the dynamic and noisy nature of the web were highlighted. Solutions for the above stated problems would discuss. Web usage mining result can improved by analyzing web content. The system integrates web page clustering and cluster labels would use as web page content indicators. Then the web page clustering has done using K-means algorithm. The clusters obtained from the web log file and integrated data file has manually summarized.

Christopher D. Manning, Prabhakar Raghavan, Hinrich

Schütze[6] has described how to build the crawler for a full-scale commercial web search engine. It is the process in which it gathers pages from the Web, in order to index them and support a search engine. The objective of crawling is to quickly and efficiently gather as many useful web pages as possible, together with the link structure that interconnects them. The resulting difficulty for the Web is focus on the web crawler sometimes referred to as a spider.

S. Sekine and J. Artiles [8] grouping the web pages referring to the same person and extracting the attributes for each of the persons sharing the same name. Some of the web people services are zoominfo.com / spock.com / 123people.com. The extracted attributes contains description of the attribute class as DOB, birth of place, other name, occupation, affiliation, award, school, major, degree, mentor, nationality, relatives, phone, fax, e-mail and websites. These are the attributes list retrieved from each cluster of the document (display any one attribute). The attribute information retrieved from the cluster is incomplete.

Y. Matsuo, J. Mori, M. Hamasaki, K. Ishida, T. Nishimura, H. Takeda, K. Hasida, and M. Ishizuka [9] proposed Social networks or Referral web employs several advanced techniques to extract relations of persons, to detect groups of persons, and to obtain keywords for a person. Search engines, especially Google, are used to measure co-occurrence of information and obtain Web documents. Social networks are obtained using analyses of Web pages, e-mail messages, and publications and self-created profiles. POLYPHONET uses co-occurrence information to search the entire web for a person's name. The extract keyword algorithm will collect documents retrieved by a person name and obtain a set of words, phrases as candidates for keywords. If two names co-occur in the same line, they are classified as co-authors.

Fiala D. [5] proposed Cite Seeker, a tool for automated citations retrieval on the Web using fuzzy search techniques based on the.NET platform and is almost entirely written in C sharp. However, it uses a number of external utilities that help handle non-textual documents such as archives, PostScript and PDF files, etc.., Inputs for CiteSeeker and its outputs are text files, but CiteSeeker also provides a comfortable graphical user interface, which allows the user to set many search parameters or even submit queries to Google.

P. Srinivasan, J. Mitchell, O. Bodenreider, G. Pant, andF. Menczer [1] presented the typical use of crawlers has been for creating and maintaining indexes for general purpose search engines, diverse usage of topical crawlers is emerging both for client and server-based applications. Topical crawlers are becoming important tools to support applications such as specialized Web portals, online searching, and competitive intelligence. A crawler used in biomedical applications is proposed to find relevant literature on a gene. On a different note, there are some controversial applications of crawlers such as extracting e-mail addresses from Web sites for spamming.

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka [7] proposed Lexical pattern-based approach to extract aliases of a given name using snippets returned by a web search engine. The lexical patterns generated automatically using a set of real world name alias data. To select the best aliases among the extracted candidates, proposed numerous ranking scores based upon three approaches: lexical pattern frequency, word co-occurrences in an anchor text graph, and page counts on the web. Moreover, using real-world name alias data, train a ranking support vector machine to learn the optimal combination of individual ranking scores to construct a robust alias extraction method. Moreover evaluate the aliases extracted by the proposed method in an information retrieval task and a relation extraction task. The extracted pattern contains words, symbols and punctuation markers. In web snippets, candidates extracted by lexical pattern might include some invalid aliases.

Muthusamy, and Subramani [29] has presented the survey article explains about the extraction and retrieval of personal name alias using various techniques from the web with the help of web crawls. The existing methods help to improve the depth of knowledge relevant to alias extraction and retrieval process. The various studies [17], [18] have shown that Google outperforms other search engines in terms of the size of its databases, frequency of Web crawls, rapidity of responses to user queries, richness of its databases, and so on. Exact numbers may found in [5]. The researchers do not present them here because they usually change very quickly. But the researchers do present a summary of Google's properties in Table 1. Google has recently introduced many services related to Web searching such as Google Scholar, Google Local, Froogle, and particularly Google Desktop, which brings the power of Google's indexing capabilities to users' personal computers.

Table 1. *Google's Strengths.*

Strength	Description
size	It has the largest database including many file types.
relevance	Pages linked from others with a greater weight given to authoritative sites are ranked higher (PageRank analysis).
cached archive	It is the only search engine providing access to pages at the time they were indexed.
Freshness	The average time of page re-indexing is one month.
Specialquery terms	It offers a number of special query terms enabling very specific searches.

4. Related Works

A web people search engine is designed as a software system to find information on the World Wide Web (WWW) [12]. The people search results are presented in a line of results often referred to as search engine results pages (SERPs) is the listing of results returned by a search engine in response to a keyword query as depict in Fig. 3. The information may be a mix of web pages, images, and other types of files. Some search engines also mine data available in databases or open directories. Unlike web directories, which are maintained only by human editors, search engines also maintain real-time information by running an algorithm on a web crawler. A single name is shared by many persons arise an ambiguity has recently become an active research topic and, simultaneously,

a relevant application domain for Web search services. Pipl, Comfibook, Ex.plode.us, InfoSpace, PeekYou, Spokeo, Worldwide Helpers, Zabasearch.com, Zoominfo.com, Spock.com and 123people.com are examples of web sites which perform web people search, al- though with limited disambiguation capabilities.

Figure 3. *Information retrieval of individual person name from Pipl.*

Pipl was launched in 2007 and has been very popular since the user looking for people search. Mostly 80-90% people using Pipl is a people search engine that search the Invisible Web for information; which compiles and produce data from 70 plus social media networking services, search engines and other databases based on the user's search. Pipl search results include data from every popular social networking service like Facebook, Twitter, LinkedIn or MySpace. Pipl search results are even better and reliable than many search engine result. Pipl is more effective than other search engines as it works on the principle of searching through deep web. It is different from most of present day search engines, as unlike other popular search engines it is used exclusively for searching people around the world. You can get your search results simply by entering name or location or email id of the person. The Search result returned relevant images and link to social-media profiles.

4.1. Web Scrapping

Web scraping also termed Screen Scraping, Web Data Extraction, Web Harvesting etc.., is to harvest or extract unstructured data [3], often texts, from the Web. The Internet provides massive amounts of text data from sources like blogs, online newspapers, social network platforms, etc.., especially in the areas like Social Sciences and Linguistics, this type of data provides a valuable resource for research. Companies such as Google, Face book, Twitter, or Amazon provide API allows analysts to retrieve data. Web scraping is closely related to web indexing [20], which indexes information on the web using web crawler and is a universal technique adopted by most search engines. In contrast, web scraping focuses more on the transformation of unstructured data on the web, typically in HTML format, into structured data that can be stored and analyzed in a central local database or spreadsheet. Web scraping is also related to web automation, which simulates human browsing using computer software. Uses of web scraping include online price comparison, contact scraping, weather data monitoring, website change detection, research, web mash up and web data integration [20]. The techniques for web scraping vary widely in effort and complexity. Some of the main web harvesting techniques is as follows:

a. Copy and Paste It means literally going to a website and copying the needed information and then pasting it into the document.
b. Text grepping and regular expression matching Text grepping is a command-line utility that allows you to search plain text on websites that match a regular expression. Originally developed for UNIX, but has evolved to include other operating systems (OS).
c. HTTP programming Static and dynamic web pages can retrieved by posting HTTP requests to the remote web server using socket programming.
d. DOM parsing DOM parsing is the practice of retrieving dynamic content generated by client side scripts that execute in a web browser such as Internet Explorer, Mozilla Firefox, or Google Chrome. Client side scripts are usually embedded within an HTML or XHTML document. The dynamic content is typically formatted in XML which enables it to be transferred from the website into your specified format.
e. Web-scraping software there is many software tools available that can be used to customize web-scraping

solutions. SEO software is much easier such as ScrapeBox, ScreamingFrog or URLProfiler. There are other more recent web-scraping software's such as Mozenda, Kimono Labs, or Import.io which allow you to easily select web page elements you would like to extract. These elements are dumped into structured columns and rows in an automated fashion and exported into an excel file or even custom API.

4.2. Limitations

The demerits of Pipl, however, sometimes the search results it returns on any given name may be bulky. There are generally many people with same name and in some countries the users will able to find many people with same name and surnames also, here your search can be little hazy for you. A person finding someone in the whole world can take out time to reach the right person by going through all the profiles.

5. Method

To download, and optimize multiple text document from the web, the following procedure can be followed systematically.The contributions of the work can be summarizedas follows,

a. By creating a wrapper class to download entire text documents from WWW utilize web search engine as Google. A text extraction algorithm have proposed for extracting text content of a person based on their alias, nick, or real names as a seed query.
b. Large numbers of new web pages show more content of the web page focused, tends to be concentrated under a handful of nodes, and the remaining nodes mainly are noise nodes such as advertising, pictures, audios, videos etc., are detached.
c. The URL is limited and the text information has optimized based on HTML tags. It is applicable only to semi-structured or un-structured secured web documents.
d. Finally, the downloaded text documents are loaded into the corpus for further text analytics.

The proposed method framed as a graphical representation consisting of set of vertices and edges in the form of G (V, E). Where, V identifies to a set links or paths in a graph, E indicates a set of nodes which can be transmitted in the form of lexical pattern. In Fig. 4.depicts the web graph structure by means of document structure and hyperlinks.

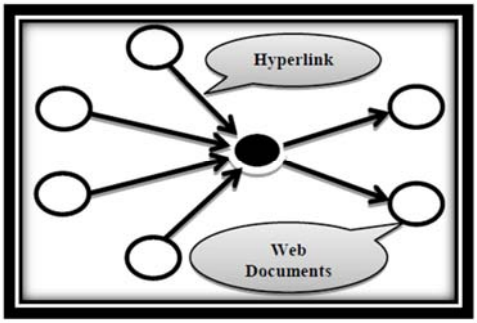

Figure 4. *Web as a Graph Structure.*

Text Extraction Algorithm for Person Name Aliases

STEP1The attributes or entities of a person search have done by providing input values as a parameter with the help of patterns generated by extraction algorithm. Where a search engine such as Google / bingo / yahoo returns a set of Web page URL within a snippet, in ranked order, that are deemed to be relevant to the search keyword entered (the alias name of a person in this case).

STEP2Next step is to extracts and parses HTML and Web related text dataon each webpage, such as hyperlinks. By using a Wrapper class, it helps to read and write the optimized text information in a unified way. Whereas, it is consisting of set of attributes or entities like Fullname, Date of Birth, Nick name / Alias name, Organization, Location etc., as a text document.

STEP3 The text document downloaded from the web is loaded into an R environment using Corpus () function from textmining (tm) package in R. Corpus handle multiple lists of text document.

6. Implementation Considerations

6.1. Lexical Pattern Extraction Method

The proposed method is based on alias / real name / nickname detection to extract the lexical pattern of a person [30], initially from the web retrieved from the web search engine. For lexical pattern extraction initially construct training data set which consists of alias name / nickname / real name, profession and location of person names. These data sets are framed with the help of social media network like Wikipedia, Face book, Twitter, Linked In etc.., are available on the web. Pattern Generator Comprises Pattern Extraction algorithm [30] to automatically generate lexical patterns with the help of trained data set. Then the confidence of extracted lexical patterns is evaluated and it retains the patterns that can accurately discover aliases for various personal names.

Table 2. *Lexical Pattern Based Approach.*

Pattern based approach
Alias name or Real name or Nick name
Alias name* profession
Alias name * location
Alias name * profession * location

Lexical pattern template
<name> commonly known as <name>
<name> also known as <name>
<alias name>worked as <profession>
<alias name>working as <profession>
<alias name>doing <profession>
<alias name>lives in <location>
<alias name>was born in <date of birth>

a. If the personal name under consideration and a candidate alias name occur frequently, then it can be considered as a good alias for the personal name.
b. Consequently, ranking is performed in the descending

order in which the term appear with a name as depict in Table 2.

6.2. Wrapper Class

In Wrapper Generation, it provides information on the capability of sources. Web pages are already ranked by traditional search engines. According to the query web pages are retrieved by using the value of page rank. The sources are what query they will answer and the output types. The wrappers will also provide a variety of Meta information. E.g. Domains, statistics, index look up about the sources. It uses internal HTML mark-up language to increase the effectiveness of text mining in semi-structured or unstructured web documents. An internet resource contains relational data. It uses formatting mark-up clearly present the information they contain to users. However, it is quite difficult to extract data from such resources in an automatic way. The standard HTML tags designed to overcome these problems;

The Wrapper is written in C sharp code with HTML tags from which the text information was generated; it retrieves relevant information from the web with the help of Google and save it as text document.It is possible to infer such wrappers by induction. It comprises a set of web pages with attributes or entities representing the information derived from each web page. This can be done by iterating over all choices of delimiters, stopping when a consistent wrapper is encountered. One advantage of automatic wrapper induction is that recognition then depends on a minimal set of indications, providing various justifications against extraneous text.

To train and evaluate the proposed method, a web crawler is created to scrap the URL from Google and placed it in a snippet. A Web search for a person, queried in the form of lexical pattern as “dhoni * cricket” and it will return web pages relevant to any person with the name as Mahendra Singh Dhoni as shown in Fig. 5.

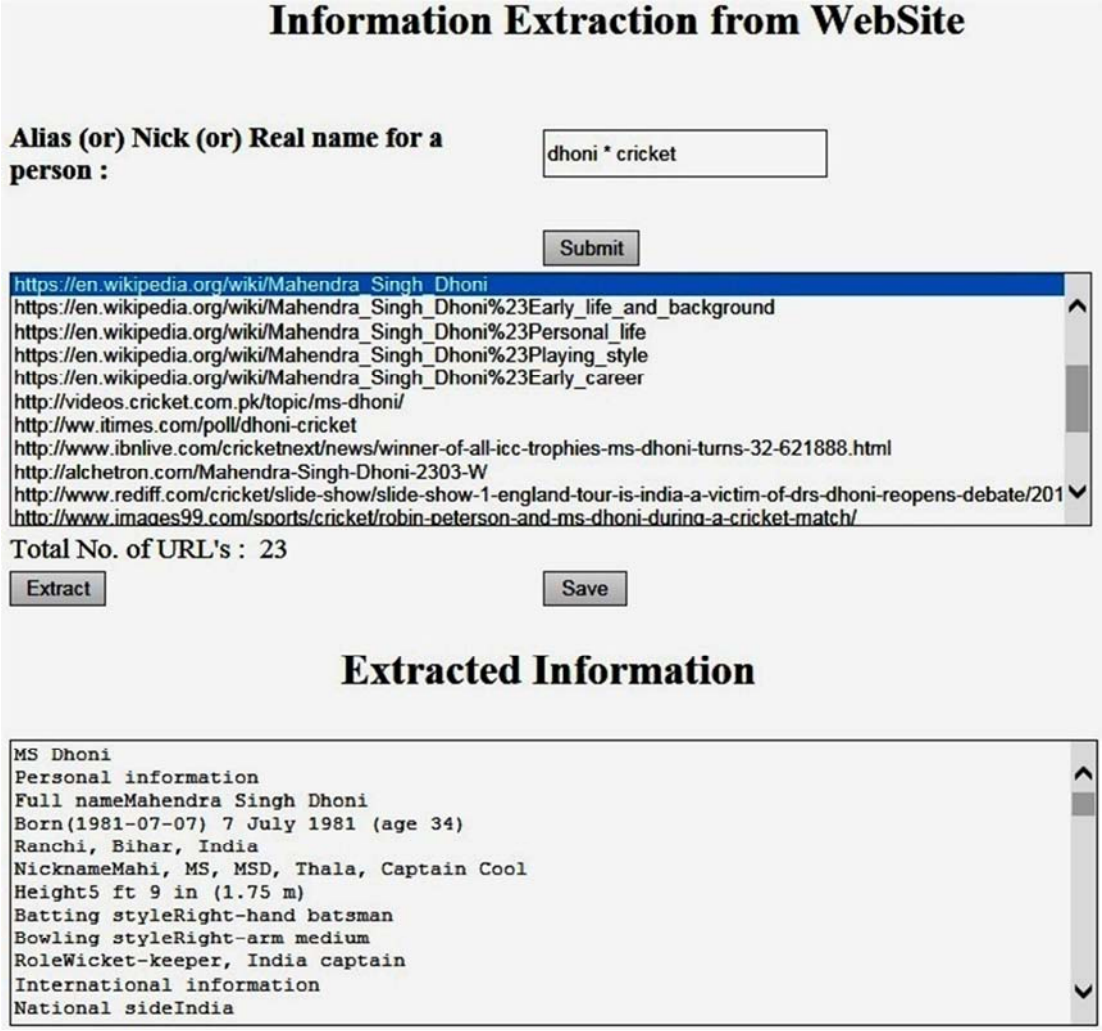

Figure 5. *Web Crawler (Google).*

By Choosing URL, then click extract function for text information extracted from the web as exposed in a container. Then by clicking save function accordingly, the text information directly downloaded to the directory as TextMining folder. Next, a corpusin R is utilized to handle multiple documents used for text analytics.

7. Experiment Results and Discussion

7.1. Pattern Selection

By using all lexical patterns shown in Table 3 are equally informative about aliases of a real name. Consequently, the patterns are ranked according to their F-scores to identify the patterns that accurately convey information about aliases. F-score of a pattern s is compute as the harmonic mean between the precision and recall of the pattern [32]. First, for a pattern, the precision and recall are computed as follows:

$$\text{Precision(s)} = \frac{\text{No. of correct aliases retrieved by s}}{\text{No. of total aliases retrieved by s}}$$

$$\text{Recall(s)} = \frac{\text{No. of correct aliases retrieved by s}}{\text{No. of total aliases in the dataset}}$$

s identify to Snippet.

Then, its F-score can computed as,

$$\text{F(s)} = 2 * \left[\frac{\text{Precision(s)} \times \text{Recall(s)}}{\text{Precision(s)} + \text{Recall(s)}}\right]$$

It is noteworthy that most aliases do not share any words with the name nor acronyms thus would not be correctly extract from approximate string matching methods.

Table 3. *Lexical Pattern with Google Results.*

Pattern	Precision P	Recall R	F-score F
<name> commonly known as <name>	0.512	0.73	0.602
<name> also known as <name>	0.497	0.739	0.594
<alias name>worked as <profession>	0.489	0.746	0.591
<alias name>working as <profession>	0.334	0.712	0.455
<alias name>doing <profession>	0.312	0.687	0.429
<alias name>lives in <location>	0.302	0.591	0.4
<alias name>was born in <date of birth>	0.295	0.582	0.392

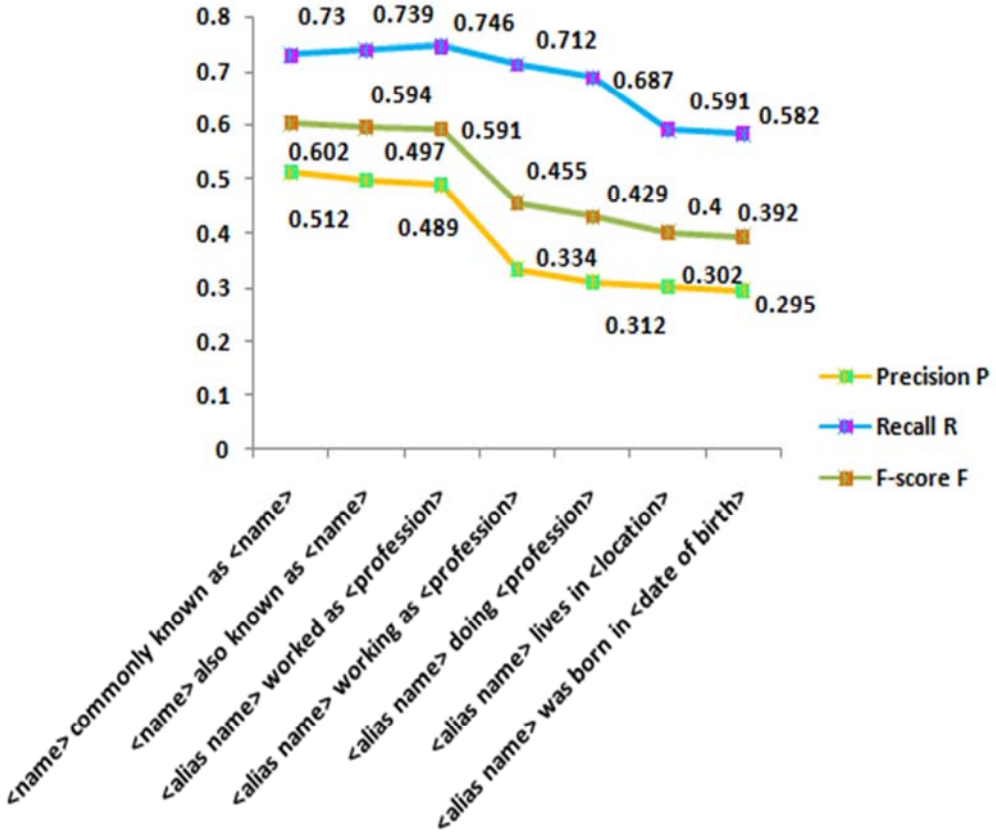

Figure 6. *Lexical Pattern Analysis.*

In Fig. 6, the overall recall of using a set of patterns is computed as the ratio of the number of aliases extracted using all the patterns in the set to the total number of correct aliases in the dataset. The experimental results are shown in Fig.6. It clearly stated that greater number of patterns rapidly enhances the overall recall. However, low-precision patterns do not increase recall to a great degree.

7.2. Comparison of Major Search Engine

Today, there are many search engines available to web searchers. What makes one search engine different from another? Following are some important measure [2].

a. The contents of that database are a crucial factor determining whether or not you will succeed in finding the information needed. Because when the peoples are doing searching, they are not actually searching the Web directly. Rather, they are searching the cache of the web or database that contains information about all the Web sites visited by that search engine's spider or crawler.
b. Size is also one important measure. How many Web

pages has the spider visited, scanned, and stored in the database. Some of the larger Search Engines have databases that are covering over three billion Web pages, while the databases of smaller Search Engines cover half a billion or less.

c. Another important measure is how up to date the database is. As the researchers know, the Web is continuously changing and growing. New Websites appear, old sites vanish, and existing sites modify their content. So the information stored in the database will become out of date unless Search engine's spider keep up with these changes.

d. In addition with these, the ranking algorithm used by the Search Engine determines whether the most relevant search results appear or not towards the top of results list.

Table 4. *Types of Web Search Engine in 2015.*

Search Engine	Market Share in September 2015
Google	67.49%
Yahoo!	10.77%
Bing	10.66%
Baidu	8.13%
AOL	1.08%
Ask	0.21%
Lycos	0.01%

Google has been in the search game a long time, it has the highest share market of Search Engine as shown in Fig. 7. (67.49%) [15]Web Crawler-based service provides both comprehensive coverage of the Web along with great relevancy. In Table 4 depicts that Google is much better than the other engines at determining whether a link is an artificial link or true editorial link. Google gives much importance to Sites which add new content on a regular basis. This is why Google likes blogs, especially popular ones. Google prefer informational pages to commercial sites. A page on a site or sub domain of a site with significant age or link can rank much better than it should, even with no external citations. It has aggressive duplicate content filters that filter out many pages with similar content. Crawl depth determined not only by link quantity, but also link quality. Excessive low quality links may make your site less likely to be crawled deep or even included in the index. In addition the researchers can search for twelve different file formats, cached pages, images, news and Usenet group postings.

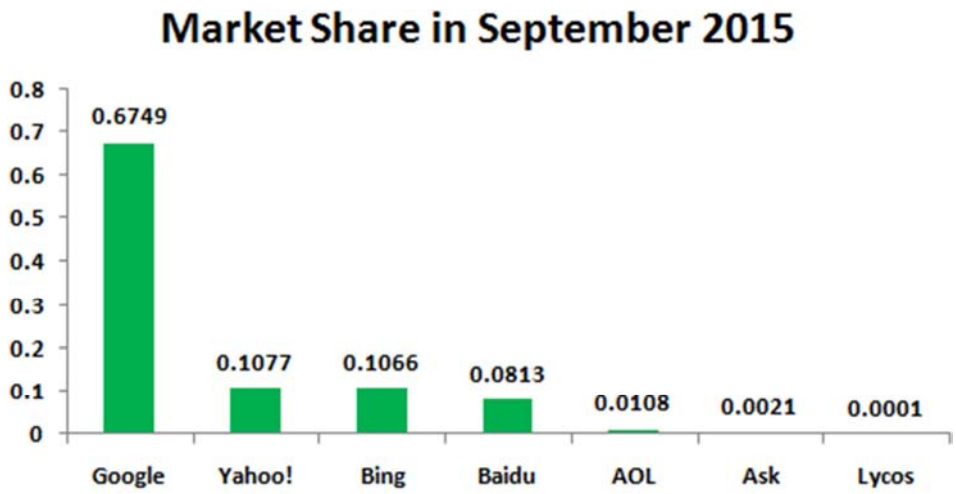

Figure 7. *Web Crawler (Google).*

8. Conclusion and Discussion

Wrapper Class is the essential cause of information retrieval in which the given query traverses to the Google search engine to download web documents that suit the user's need. Initially, Lexical template Pattern is used as Input parameter in query form to retrieve the relevant text document from WWW and saved it in a TextMining folder. In Future, Corpus() in R is handled to pre-process, transform the text with text mining functions (tm) to eliminate the noisy data and produce the relevant attributes or entities of a personal name aliases. The lexical template pattern are created automatically, in which significantly improves the harmonic mean between the precision and recall of the pattern at a rate of 60.2 %. Google seems to obey the motto "high precision is important even at the expense of recall". Web page ranking method called PageRank to present the most relevant results upon a user query is used based on the link structure of the Web. Apart from PageRank, Google employs a number of techniques to improve search quality with innovative features like Anchor Text, Proximity Search, Word Presentation, and Pages Repository. Google yields 67.49% of relevant information when compared to other search engine. Finally, it can be improved and verified with the help of pattern recognition technique.

References

[1] P. Srinivasan, J. Mitchell, O. Bodenreider, G. Pant, and F. Menczer. Web crawling agents for retrieving biomedical nformation. In NETTAB: Agents in Bioinformatics, Bologna, Italy, 2002.

[2] Krishan Kant Lavania, Sapna Jain, Madhur Kumar Gupta, and Nicy Sharma,"Google: A Case Study (Web Searching and Crawling)", International Journal of Computer Theory and Engineering, Vol. 5, No. 2, April 2013.

[3] Patrick Mair and Scott Chamberlain," Web Technologies Task View", The R Journal Vol. 6/1, June 2014, ISSN 2073-4859, pp.178-181.

[4] Brin S., Page L.: "The Anatomy of a Large-Scale Hypertextual Web Search Engine", Proceedings of the 7th World Wide Web Conference, pp.107 -117, 1998.

[5] Fiala D. "A System for Citations Retrieval on the Web", MSc. thesis, University of West Bohemia in Pilsen, 2003.

[6] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, "An introduction to Information Retrieval", online Edition 2009 Cambridge University Press Cambridge, England pp.443-459.

[7] D. Bollegala, Y. Matsuo and M. Ishizuka, "Automatic Discovery of Personal Name Aliases from the Web," IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 6, (2011) June.

[8] S. Sekine and J. Artiles, "Weps 2 Evaluation Campaign: Overview of the Web People Search Attribute Extraction Task," Proc. Second Web People Search Evaluation Workshop (WePS '09) at 18th Int'l World Wide Web Conf., (2009).

[9] Y. Matsuo, J. Mori, M. Hamasaki, K. Ishida, T. Nishimura, H. Takeda, K. Hasida and M. Ishizuka, "Polyphonet: An Advanced Social Network Extraction System," Proc. WWW '06, (2006).

[10] Rama Subbu Lakshmi B, Jayabhaduri R, "Automatic Discovery of Association Orders between Name and Aliases from the Web using Anchor Texts-based Co-occurrences", International Journal of Computer Applications (0975 – 8887) Volume 41– No.19, March 2012.

[11] Singh, B. and Singh, H. K. 2010. Web Data Mining Research: A Survey. Computational Intelligence and Computing Research (ICCIC).IEEE International Conference, pp. 1-10.

[12] Mr. A. Muthusamy and Dr. A. Subramani "Lexical Pattern Extraction from Data Set Make Use of Personal Name Aliases", in International Journal of Advancements in Computing Technology ISSN: 2005-8039(print), 2233-9337(online) Vol 7, No.3 May 2015,pp. 102-108.

[13] Basic Search Handout URL: WWW.digitallearn.org.

[14] Web Search Engine URL: www.wikipedia.org.

[15] Web Search Engine market Share URL: https:// en.wikipedia.org/wiki/Web_search_engine#Market share.

[16] Google URL: http://www.google.com.

[17] Freshness Showdown URL: http://www.searchengineshowdown.com/stats/freshness.shtml.

[18] Search Engines Showdown: Size Comparison Methodology: URL:

[19] http://www.searchengineshowdown.com/stats/methodology.shtml.

[20] RFC 1950 URL: http://www.faqs.org/rfcs/rfc1950.html.

[21] Web Scrapping URL: https://en.wikipedia.org/wiki/Web_scraping.

[22] Baeza-Yates R., Castillo C. Crawling the infinite Web: five levels are enough. Proceedings of the third Workshop on Web Graphs (WAW), Rome, Italy, Lecture Notes in Computer Science, Springer, vol. 3243, pp. 156-167, 2004.

[23] Baeza-Yates R., Castillo C., Marín M., Rodríguez A. Crawling a country: better strategies than breadth-first for web page ordering. Proceedings of the 14th international conference on World Wide Web (WWW 2005), Chiba, Japan, pp. 864-872, 2005.

[24] Chakrabarti S. Mining the Web: Analysis of Hypertext and Semi Structured Data. Morgan Kaufmann Publishers, San Francisco, California, USA, 2002.

[25] Chakrabarti D., Faloutsos C. Graph mining: Laws, generators, and algorithms. ACM Computing Surveys, vol. 38, no. 1, 2006.

[26] Najork M., Wiener J. L. Breadth-first crawling yields high-quality pages. Proceedings of the 10th international conference on the World Wide Web (WWW10), Hong Kong, pp. 114-118, 2001.

[27] Cho J., Garcia-Molina H., Page L. Efficient Crawling Through URL Ordering. Proceedings of the 7th international conference on the World Wide Web (WWW7), Brisbane, Australia, pp. 161-172, 1998.

[28] Abiteboul S., Preda M., Cobena G. Adaptive on-line page importance computation. Proceedings of the 12th international conference on World Wide Web (WWW'03), Budapest, Hungary, pp. 280-290, 2003.

[29] Ghemawat S., Gobioff H., Leung S.-T. The Google file system. Proceedings of the 19th ACM symposium on Operating systems principles, Bolton Landing, NY, USA, pp. 29-43, 2003.

[30] Ntoulas A., Cho J., Olston C. What's new on the web?: the evolution of the web from a search engine perspective. Proceedings of the 13th international conference on the World Wide Web (WWW '04), New York, NY, USA, pp. 1-12, 2004.

[31] Mr. A. Muthusamy and Dr. A. Subramani "Automatic Discovery of Lexical Patterns using Pattern Extraction Algorithm to Identify Personal Name Aliases with Entities", in International Journal of Software Engineering and Its Applications ISSN: 1738-9984 Vol 9, No.12(2015),pp. 165-176.

[32] Mr. A. Muthusamy and Dr. A. Subramani "A Survey of Automatic Extraction of Personal Name Alias from the Web", in International Journal of Signal Processing, Image Processing and Pattern Recognition ISSN: 2005-4254 Vol. 7, No. 6 (2014), pp. 75-84.

[33] Mr. A. Muthusamy, Dr. A. Subramani, "Framework for pattern generation from discriminating datasets", International Journal of Collaborative Intelligence Vol.1, No.2 (2015), pp. 115-123.

Biography

Dr. A. Muthusamy is currently working as an Associate Professor, Department of Computer Technology, Dr.N.G.P Arts and Science College, Coimbatore. He received his Ph.D. Degree in Computer Science from Bharathiar University, Coimbatore and received his MCA degree from Anna University, Chennai. He is a Reviewer of 2 National / International Journals. He is an Associate Editor of Journal of Computer Applications. He has published 5 technical papers in National / International Conference, 7 National / International Journal. His area of research includes Data and Web Mining.

Dr. A. Subramani is currently working as an Assistant Professor, Department of Computer Science, Government Arts College, Dharmapuri and as a Research Guide in various Universities. He received his Ph.D. Degree in Computer Applications from Anna University, Chennai. He is a Reviewer of 10 National / International Journals. He is in the editorial board of 6 International / National Journals. He is an Associate Editor of Journal of Computer Applications. He has published more than 75 technical papers at various International, National Journals, and Conference proceedings. His areas of research includes High Speed Networks, Routing Algorithm, Soft computing, Wireless Communications, Mobile Ad-hoc Networks.