Quest Journals
Journal of Electronics and Communication Engineering Research
Volume 2 ~ Issue 7(2014) pp: 01-08
ISSN(Online) : 2321-5941
www.questjournals.org

Research Paper

Content Based Web Page Re-Ranking Using Relevancy Algorithm

Harish Kumar B T[1], Vibha Lakshmikantha[2], Venugopal K R[3]
[1]*Research Scholar, Department of Computer Science & Engineering, BIT, Banglore*
[2]*Professor, Department of Computer Science & Engineering, BNMIT, Banglore*
[3]*Principal, UVCE, Bangalore*

Received 27 September, 2014; **A**ccepted 30 October, 2014 © The author(s) 2014. Published with open access at **www.questjournals.org**

ABSTRACT:- *The World Wide Web is a system of interlinked hypertext documents that are accessed via the internet.* It *plays a leading role for retrieving user requested information from the web resources. In order to retrieve user requested information, search engine plays a major role for crawling web content on different node and organizing them into result pages so that user can easily select the required information by navigating through the result pages link. This strategy worked well in earlier because, number of resources available for user request is limited. It is feasible to identify the relevant information directly by the user from the search engine results. As the Internet era increases, sharing of resource also increases and this leads to develop an automated technique to rank each web content resource. Different search engine uses different techniques to rank search results for the user query. This leads to business motivation of bringing up their web resource into top ranking position. As the competition and web resource increases, the ranking of web content becomes tedious and dynamic with respect to the user query.*

In the proposed work a new approach is introduced to rank the relevant pages based on the content and keywords rather than keyword and page ranking provided by search engines. Based on the user query, search engine results are retrieved. Every result is individually analyzed based on keywords and content. User Query is pre-processed to identify the root words. Root word is considered for Dictionary construction and Dictionary is built with synonyms for the user query. Keywords and content words of each resultant web page is pre-processed and compared against the dictionary. If a match is found, then particular weight is awarded for each word. Finally, the total relevancy of the particular link against user request is computed by summarizing all the weights of the keyword and content words. The results are then re-ranked in descending order of their weights and displayed.

KEYWORDS:- Content Words,Dictionary, Keywords, Rank, Root Words, Search Engine.

I. INTRODUCTION

A web search engine is a software system that is designed to search for information on the World Wide Web. The search results are generally presented in a line of results often referred to as search engine results pages (SERPs).The information may be mix of web pages, images, data from databases 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.

1.1 Working of Search Engine

A search engine operates in the following order:

1. Web crawling
2. Indexing
3. Searching

*Corresponding Author: Harish Kumar B T

[1]*Research Scholar, Department of Computer Science & Engineering, BIT, Banglore*

Web search engines work by storing information about many web pages, which they retrieve from the HTML markup of the pages. These pages are retrieved by a Web crawler(sometimes also known as a spider) — 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). Data about web pages are stored in an index database for use in later queries. A query from a user can be a single word. The index helps to find the information relating to the query as quickly as possible, some search engines, such as Google, store all or part of the source page (referred to as a cache) as well as information about the web pages, while others, like AltaVista, store every word of every page they find.This cached page always holds the actual search text since it is the one that is actually indexed, so it can be very useful when the content of the current page has been updated and the search terms are no longer in it.This problem might be considered a mild form of link rot or link death, and Google's handling of it increases usability by satisfying user expectations that the search terms will be on the returned webpage. This satisfies the principle of least astonishment, since the user normally expects that the search terms will be on the returned pages. Increased search relevance makes these cached pages very useful as they may contain data that may no longer be available elsewhere.

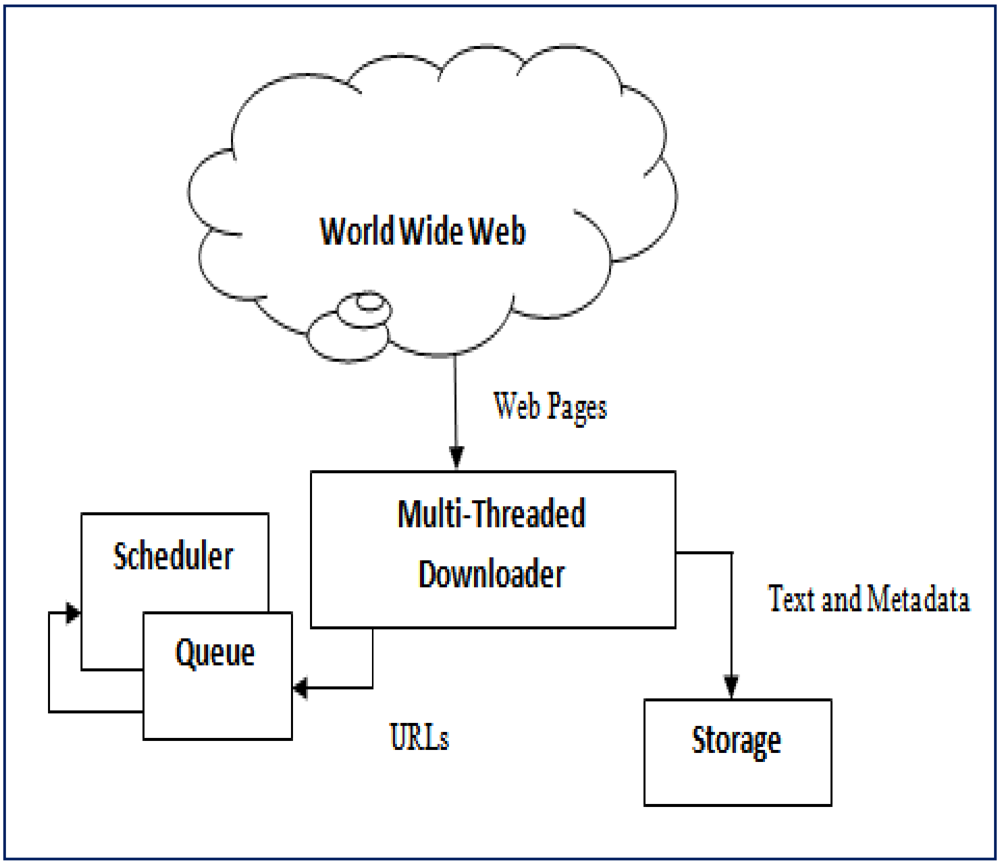


Fig 1: High-level architecture of a standard Web crawler

A user can enter a query into the search engine (typically by using keywords), 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 document's title and sometimes parts of the text. The index is built from the information stored with the data and the method by which the information is indexed. From 2007 the Google.com 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. The engine looks for the words or phrases exactly as entered. Some search engines provide an advanced feature called proximity search, which allows users to define the distance between keywords. There is also concept-based searching where the research involves using statistical analysis on pages containing the words or phrases you search for. Natural language queries allow the user to type a question in the same form as one who would speek to a human being.

Motivation: Modern search engines return millions of pages for a single user query. This amount is prohibitive to preview for human users and some web pages may be irrelevant to the given query. Hence ranking algorithm in web search is required to find a small set of most "authoritative" pages relevant to the user query.

Contribution: In the present work, the web pages are re-ranked by matching the keywords and the content words of the web pages returned by the search engine with the root words and synomys of the user query. This helps in finding the small set of most useful and helpful web pages revelant to the user query and re-ranking more relevant web pages to the users in the higher order.

Remaining sections are organized as follows. Section 2 covers the related work, Architecture modeling is presented in section 3,Section 4 discusses the problem definition and algorithm of the proposed work. Section 5 shows the results of the proposed work, performance evaluation is proposed in section 6 and finally the conclusion and references in section 7.

II. RELATED WORK

A brief survey of related work in the area of content based ranking of web pages is presented here. D. Sridevi et al., [1] have discussed the web mining techniques for applications like e-commerce and e-business. Web mining techniques on these applications enhances the users ability to access information and feel very easy and comfortable to surf.

Gyanendra Kumar and A K Sharma [2] proposed pagerank based on visits of links (VOL). Page ranking mechanism called page ranking based on visits of links is devised for search engine which works on the basic ranking algorithm of google i,e page rank and takes number of visits of inbound links of web pages into account. This concept is very useful to display most valuable pages on the top of the result list on the basis of user browsing behaviours which reduces the search scpace to large scale, but it does not take into account the page content.

Bing Liu and Kevin Chen Chuan Chang [4] have presented the important problems in web content mining and then has introduced the papers published in this regards. G. Poonkuzhaliet al.,[6] proposed a new algorithm using signed approach for improving the results of web content mining by detecting both relevant and irrelevant web documents.

SoumenChakrabarti [7] have explained the various techniques like crawling the web, web search and information retrieval, similarity and clustering, supervised learning, semisupervised learning that are generally applicable to any hypertext data normally called as semistructured or unstructured data because they do not have a compact or precise description of data items.

A focused crawler[8] is a web crawler that collects Web pages that satisfy some specific property, by carefully prioritizing the crawl frontier and managing the hyperlink exploration process. Some predicates may be based on simple, deterministic and surface properties. For example, a crawler's mission may be to crawl pages from only the .jp domain. Other predicates may be softer or comparative, e.g., "crawl pages with large PageRank", or "crawl pages about baseball". An important page property pertains to topics, leading to *topical crawlers*. For example, a topical crawler may be deployed to collect pages about solar power, or swine flu, while minimizing resources spent fetching pages on other topics. Crawl frontier management may not be the only device used by focusing crawlers; they may use a Web directory, a Web text index, backlinks, or any other Web artifact.

A focused crawler must predict the probability that an unvisited page will be relevant before actually downloading the page. A possible predictor is the anchor text of links; this was the approach taken by Pinkerton in a crawler developed in the early days of the Web. Topical crawling was first introduced by Filippo MenczerChakrabarti *et al.* coined the term *focused crawler* and used a text classifier to prioritize the crawl frontier. Andrew McCallum and co-authors also used reinforcement learning to focus crawlers. Diligenti '*et al.* traced the context graph leading up to relevant pages, and their text content, to train classifiers. A form of an online reinforcement learning has been used along with features extracted from the DOM tree and text of linking pages, to continually train classifiers that guide the crawl. In a review of topical crawling algorithms, Menczeret*al.,* show that such simple strategies are very effective for short crawls, while more sophisticated techniques such as reinforcement learning and evolutionary adaptation can give the best performance for crawls.

III. ARCHITECTURE MODELING

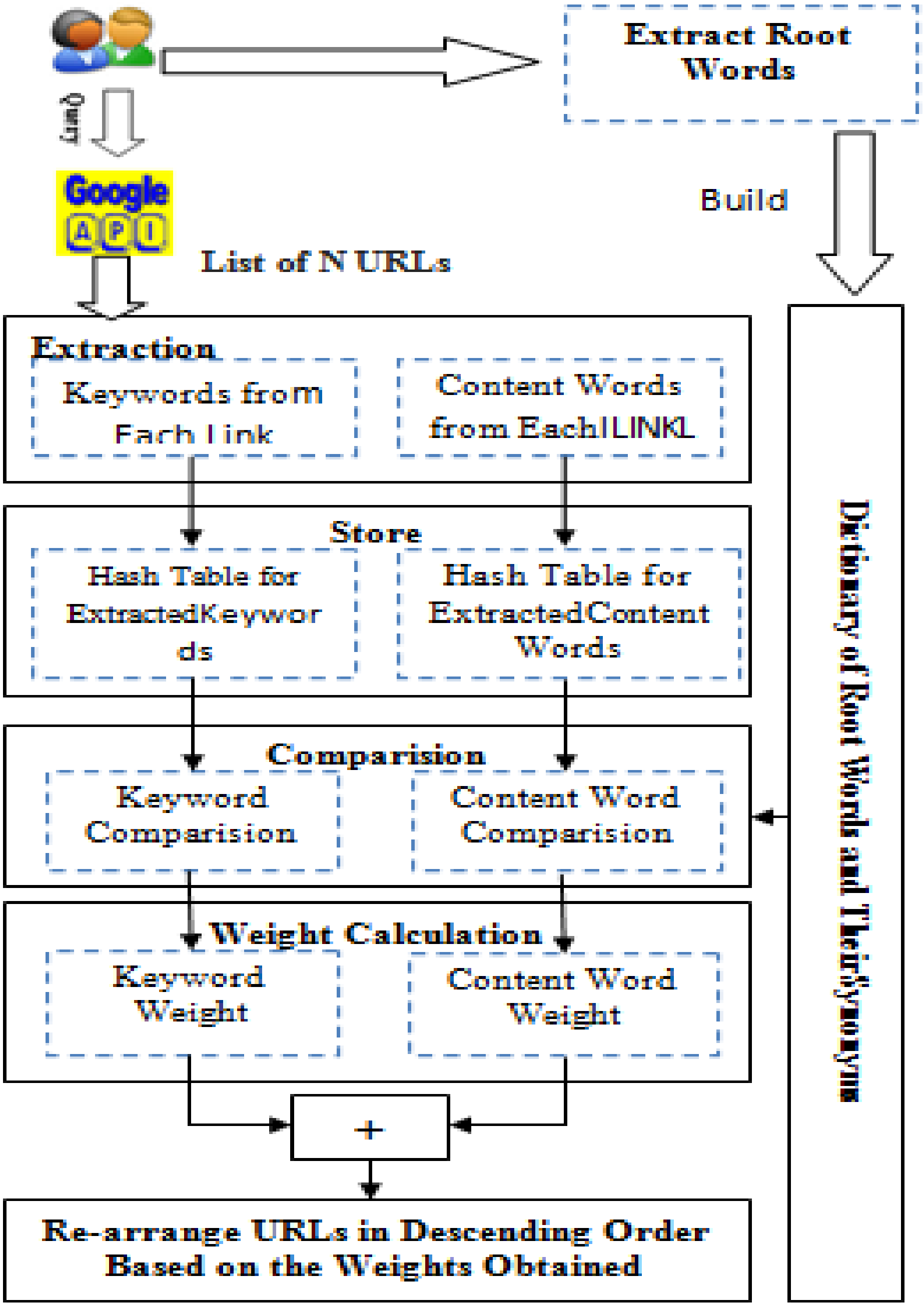

Extract Root Words
Query
Google API
Build
List of N URLs
Extraction
Keywords from Each Link
Content Words from EachLINK
Store
Hash Table for ExtractedKeywords
Hash Table for ExtractedContent Words
Comparision
Keyword Comparision
Content Word Comparision
Weight Calculation
Keyword Weight
Content Word Weight
+
Re-arrange URLs in Descending Order Based on the Weights Obtained
Dictionary of Root Words and TheirSynonyms

Fig 2: Architecture

The general architecture of the proposed work is depicted in Figure 2. The query put forth by users act as input to the Google APIs which are available to build the custom search. Root words are extracted from the user query by removing the stop words and synonyms are identified for each root word and a dictionary is constructed.

Google API produces a list of *'N'* URLs. Keywords and content words are extracted from each URL and stored in the form of a hash table. Keywords are those words that are within the HTML title tag. Content words are those words that are within the HTML body tag. Keywords are compared with the root words and its synonyms in the dictionary, if they match a weight of ten is assigned to the keyword. Similarly content words are compared with the dictionary, if they match a weight of one is assigned to the content word. Finally total relevancy is computed by summing all the keyword weights and the content word weights. This is repeated for all the URLs and based on the total relevancy values URLs are re-ranked by rearranging them in decreasing order.

IV. PROBLEM DEFINATION

World Wide Web is vast and growing in size evey day and has huge number of web pages that contain information on different topics. Users search the desired information by giving the query to the search engine which yields in both relevant and irrelevant web pages. Hence re-ranking of web pages is proposed, whose objectives are:

- Reducing search time
- Satisfying user with the relevant information

4.1Algorithm

Algorithm: Relevancy and Weight based approach
Input: User Query (*Q*)
OutPut: Re-Ranked URLs

Variables List

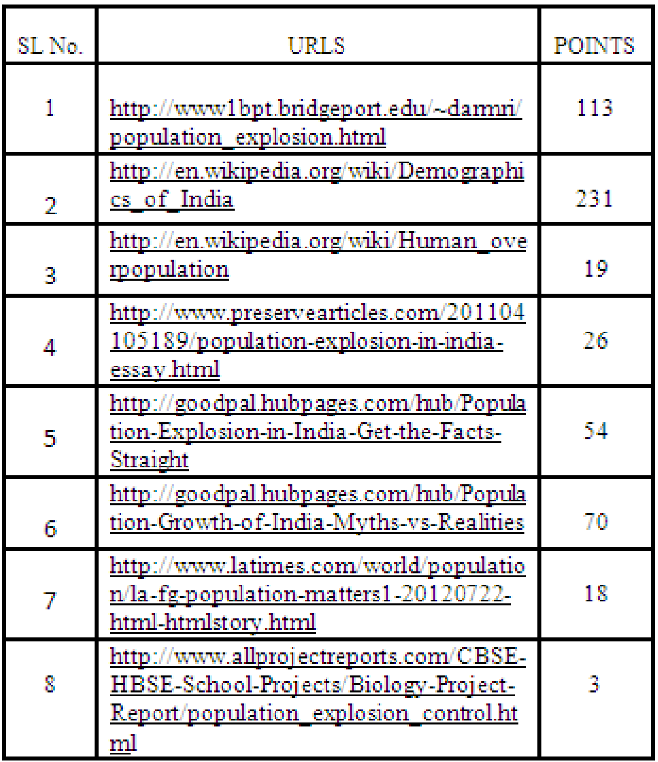
Q	: *Original User Query*
QI	:*Preprocessed User Query*
URL_LIST[N]	: *Array Storing all the URLs Returned by google API*
RW[N]	:*Array for storing important words of query* Q^I
KW[N]	:*Array for storing key words from the* URL_i^I
CW[N]	:*Array for storing Content words from the* URL_i*_BODY*
URL_SCORE[N]	:*Array to store scores copmputed for each* URL_i

Steps:

```
URL_LIST[N]←GOOGLE_API(Q)
$Q^I$<-- Preprocess(Q);
RW[N]←Extract Root Word From Query $Q^I$
For each $RW_i$in RW[N]
    Find the Synonym $S_i$ for $RW_i$
    Add $S_i$ and $RW_i$ to Dictionary D
End for $RW_i$
For each $URL_i$in URL_LIST[N]
    $URL_i^I$←Preprocess($URL_i$);
    $URL_i$.KW[N]←Extract_Keywords_From($URL_i^I$)
    $URL_i$_BODY←Preprocess(Body tag text of $URL_i$)
$URL_i$.CW[N]←Extract_Content_Words_From($URL_i$_BODY)
End for $URL_i$
For each $URL_i$in URL_LIST[N]
    For each $KW_i$ in $URL_i$.KW[N]
        Compare $KW_i$against Dictionary D
        If match then
        $URL_i$_SCORE←$URL_i$_SCORE + 10;
        Else
        $URL_i$_SCORE←$URL_i$+SCORE+0;
    End for $KW_i$
    For each $CW_i$ in $URL_i$.CW[N]
        Compare $CW_i$against Dictionary D
        If match then
        $URL_i$_SCORE←$URL_i$_SCORE + 1;
        Else
        $URL_i$_SCORE←$URL_i$_SCORE+0;
    End for $CW_i$
End For$URL_i$
Sort URL_LIST[N] based on URL_SCORE[N] in descending order.
```

V. RESULTS

Proposed algorithm is tested by giving the query “*Population Explosion in India”.* Table 1 contains the URLs’ and the corresponding points which have been derived by the application of the algorithm. If a keyword successfully matches with the dictionary entry,10 points are awarded else 0 is assigned. In case a content word successfully matches with the dictionary entry, 1 point is awarded else 0 is assigned. These points are added and assigned to the corresponding URL.



SL No.	URLS	POINTS
1	http://www1bpt.bridgeport.edu/~damri/population_explosion.html	113
2	http://en.wikipedia.org/wiki/Demographics_of_India	231
3	http://en.wikipedia.org/wiki/Human_overpopulation	19
4	http://www.preservearticles.com/201104105189/population-explosion-in-india-essay.html	26
5	http://goodpal.hubpages.com/hub/Population-Explosion-in-India-Get-the-Facts-Straight	54
6	http://goodpal.hubpages.com/hub/Population-Growth-of-India-Myths-vs-Realities	70
7	http://www.latimes.com/world/population/la-fg-population-matters1-20120722-html-htmlstory.html	18
8	http://www.allprojectreports.com/CBSE-HBSE-School-Projects/Biology-Project-Report/population_explosion_control.html	3

Table 1: Point calculation of each URL

VI. PERFORMANCE EVALUATION

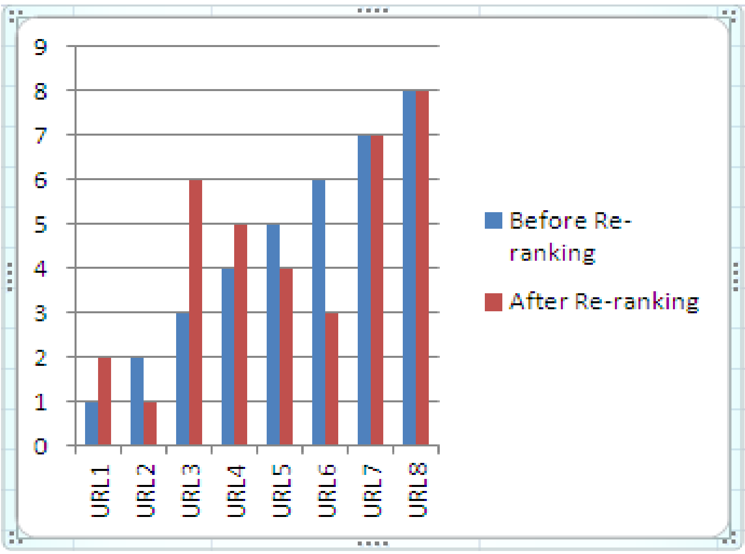


Fig 3: URLs Before and After Re-ranking

Figure 3 shows that the URL1 which was ranked first by the search engine is now ranked second, URL2 which was ranked second now ranks first. Similarly,it is observed that theother ranking variation before and after re-ranking.

The performance of the proposed system is evaluated by computing the precission using the true positive value (TP) and false positive value (FP). Formula to find the precission is as given below.

Precession= TP / (TP + FP) ------>Eq (1)

TP → Correct Identification
FP → Incorrect Identification

The Table 2 show the comparions of google rank, proposed rank and manual ranking by two different users.

URLs	Google Rank	Manual Rank	Proposed Rank
URL1	**1**	**2**	**2**
URL2	**2**	**1**	**1**
URL3	**3**	**6**	**6**
URL4	**4**	**4**	**5**
URL5	**5**	**5**	**4**
URL6	**6**	**3**	**3**
URL7	**7**	**7**	**7**
URL8	**8**	**8**	**8**

Table 2: Rank Comparision

Google has true positive value of 4 for URLs 4, 5, 7, 8 and false positive value of 4 for URLs 1, 2, 3, 6. Proposed system has true positive value of 6 for URLs 1, 2, 3, 6, 7, 8 and false positive value of only 2 for URLs 5, 6. The following graph shows the precission for google rank and the proposed rank.

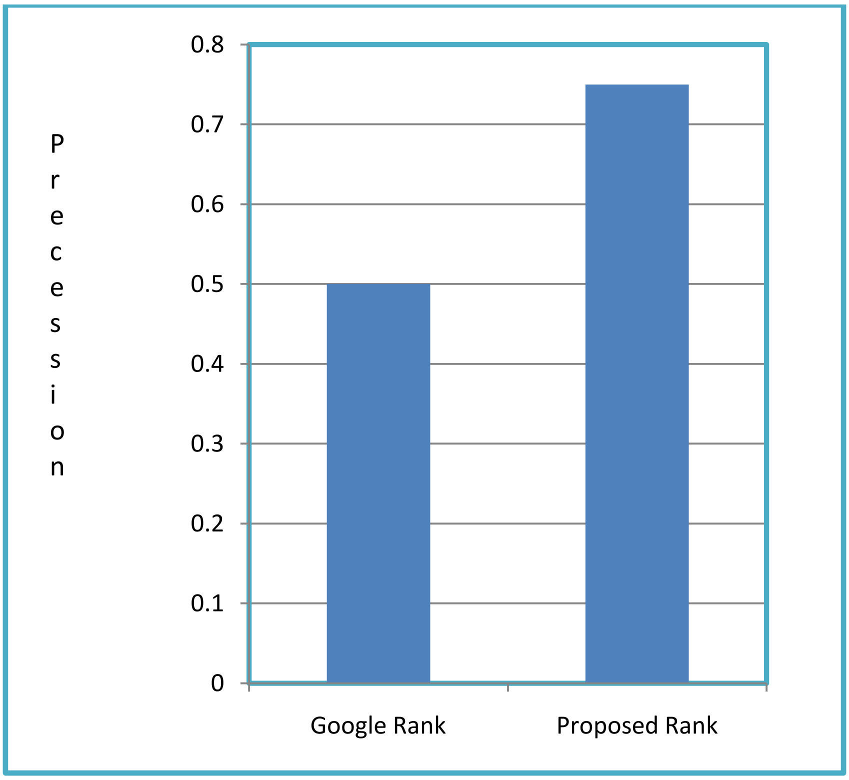


Fig 4: Precession of Google Rank and Proposed Rank

VII. CONCLUSION

The results produced by the search engine are enormous and irrelevant to the user context. The proposed approach makes use of the text mining on the web pages listed by the search engine to rank and increase the relevancy of pages and provide users with quality search results. The experimental results show that the results obtained by this approach have more relevant web pages displayed at the higher positions to the user.

REFERENCES

[1]. D. Sridevi et al., “Survey on Latest Trends in Web Mining”, International Journal of Research in Advent Technologies, Vol. 2, No. 3, E-ISSN:2321-9637, March-2014

[2]. Gyanendra Kumar et al., “Page Ranking Based on Number of Visits of Links of Webpage”, YMCA University of Science & Technology, Faridabad, India(ICCCT-2011)

[3]. Sungrim Kim, “Information Retrieval using Context Information on the Web 2.0 Environment”, Joonhee Kwon (IJCSNS-2009)

[4]. Bing Liu and Kevin Chen Chuan Chang, “Editorial: Special Issue onWeb Content Mining”, SIGKDD Explorations, Volume 6, Issue 2.

[5]. Bin W andLiuZhijing, “Web Mining Research”, 5th InternationalConference on computational Intelligence and MultimediaApplications, 2003

[6]. G. Poonkuzhali et al., “Signed Approach for Mining Web Content Outliers”,International Science Index waset.org/Publication/13636Vol:3, No:8, 2009.

[7]. Chakrabarti S, “Mining the Web: Discovering Knowledge fromHypertext Data”, Morgan-Kauman Publishers,2002.

[8]. Chakrabarti S et al., “Focused Crawling: A NewApproach to Topic-Specific Web Resource Discovery”,ComputerNetworks, Amsterdam, Netherlands, 1999.

[9]. Cheng Wang et al., “A Utility basedWeb Content Sensitivity Mining Approach”, InternationalConference on Web Intelligent and Intelligent Agent Technology(WIIAT), IEEE/WIC/ACM 2008.

[10]. R. Cooley et al., “Web mining:Information and Pattern Discovery on the World Wide Web”, InProceedings of the 9th IEEE International Conference on Toolswith Artificial Intelligence (ICTAI’97), 1997.

[11]. GeorgiesLappas, “An Overview of Web Mining in Societal BenefitAreas”, The 9th IEEE International Conference on E-CommerceTechnology, IEEE 2007.

[12]. Gibson, J et al., "Adaptive Web-Page ContentIdentification", In WIDM ’07:Proceedings of the 9th annual ACMinternational workshop on Web information and data management.New York, USA,2007.

[13]. Han, J. and Kamber, M. “Data Mining: Concepts and Techniques”,Morgan Kaufmann Publishers,2001.

[14]. Hongqi li et al., “Research on the Techniquesfor Effectively Searching and Retrieving Information fromInternet”, International Symposium on Electronic Commerce andSecurity, IEEE 2008.

[15]. Jaroslav Pokorny and JozefSmizansky, “Page Content Rank: Anapproach to the Web Content Mining”.

[16]. Junghoo Cho et al.,“Finding replicated Web collections”, MOD 2000, Dallas, TX USA.

[17]. Raymond Kosala and Hendrik Blockeel, “Web Mining Research: ASurvey”, ACM SIGKDD, July 2000, Vol-2, pp 1-15.

[18]. Kshitija Pol et al., “A Surveyon Web Content Mining and Extraction of Structured andSemistructured data”,First International Conference on Emergingtrends in Engineering and Technology, 2008.