A Novel Search Approach in Combination of Search Engine and Meta Search Engine by using Page Rank and Query Optimization

Naresh Kumar
Asst. Prof., CSE deptt.
MSIT, Delhi, India

Dr. Rajender Nath
Professor, DCSA,
K. U., Kurukshetra, Haryana, India

***Abstract*— this paper proposes a new technique which helps the user to search the information on the World Wide Web. The main feature of this system is to record those queries which remain un- solvable with the current existing system. If, any information corresponding to these queries is found late on then this information is mailed to the recorded e-mail id. At the same time proposed system also update its own database so that it can be made available to the users next time. This information retrieval system provide the best possible results with the help of meta search engine and proves to be more promising and efficient.**

Keywords—World Wide Web, Information Retreival, Query Log, Ranking Algorithm, Search Engine, Meta Search Engine.

I. INTRODUCTION

The size of World Wide Web (WWW) and its popularity is increasing much more than its development days. Due to which searching the information on the WWW becomes a difficult task. Search engine (SE) searches the information based on the keywords provided to them [1]. But, searching using keywords is not an easy task [2] because for a single word lot of meaning is there on the web. This may results in retrieval of documents which are not required by the user [3]. All this happens due to small size of the query. According to [4], most of the queries are limited to 2 terms per query. As all the users are not expert in searching the information on the web, so, the major factor responsible for this is user's awareness in framing the queries [5]. They are not well skilled in organizing and formulating their queries. In current scenario, getting a set of web pages corresponding to the given query is not a big issue rather issue becomes at the user end, when user has to go through the resultant list of URLs to find the required contents [6]. In order to overcome such type of problems, some search engines suggests an alternative queries to users. These queries suggestions are comes from the list of words collected through various information sources like Wordnet [7].

The main aim of the work proposed in this paper is to optimize the results of a search engine by returning the more relevant web pages to end user. Web log is an excellent way to know about "what user wants" [8][9]. It returns a list of suggestive key words with the returned URLs. These suggestions are taken from the previous history of query. There is also a provision for providing the search results later on. This happens only when, if there is no searched results found instantly.

II. RELATED WORK

A method based on query clustering was proposed in [10]. The similar queries were clustered in a group and information related to them was stored in query log files. The clustering procedure uses historical preferences of users registered in the query log of SE. This method suggests the related queries and approves their relevancy also. Authors of this paper find out the similarity between two queries by using the term weight vector. They also measure the attraction of user towards the results returned by the SE. for their experiment they took log file of Google containing records of 15 days. This log file contained 6042 queries, 22190 hits and 18597 different URLs. They used K-means clustering algorithm using CLUTO software package and compute the clusters successively. At the end they showed created clusters and their corresponding queries. Based on these achieved results the precision was calculated. For first three queries they claimed to achieve 80% precision.

In [11], proposed an automatic suggesting method to find the similar queries. This system makes the use of collected knowledge from different users of SE to recommend new ways of expressing the same information need. The authors considered different query similarity measures i.e. Naïve query-based method, Naïve simplified URL based method, Naïve URL-based method, Query-Title-based method, Query-Content-based method, Common query title method and Common query text method. The results corresponding to the above mention measures were stored. They implement their proposed work on Linux using NetBSD and MySQL database. The processing of queries was implemented in Python scripting language, whereas, the similarity engine was Implemented in C++. They got two things from their experimental results. First was richer the query memory then better will the recommendations of a system. Second observation was difficulty in selection of best method. According to them some combination of different methods with assigned weights produced better results than every single similarity taken alone.

A system based on learning from query logs by predicting user information needs was proposed in [12]. To achieve this they mine the query log using similarity function. They choose only seven query session to conduct the experiment and five functions were tested. These functions were keyword similarity ($sim_{keyword}$), similarity using documents clicks (sim_{click}), similarity using both keyword and document clicks ($sim_{combined}$), query clustering and updater of rank. For similarity and clustering calculations they consider similarity based on query keywords, similarity based on clicked URLs and Bipartite graph of query log. At the end they proposed combined similarity measure and clustering algorithm to cluster the queries.

III. MOTIVATING EXAMPLE

Generally, user searches his required information through any search engine. He becomes satisfied, if the desired information is achieved through the search system. But, what happens if he got a massage like “your search did not match any document”. After this massage, user left the system and assumes that web has no information related to his query. But, crawl the web by a crawler is a regular process. So, it may become possible that the previous query that has no information on the web, but now become available to the SE. Therefore, this information must be propagated to the user who had left the system. This is possible only when if SE interface take some information from the user like email-id, number of days and number of hours etc. The number of days or hours provides the time bound that if SE got this information in provided time limit then it is directly forwarded to the user. If the desired information is found beyond the time limit then this is updated only in SE database for future response.

IV. PROPOSED WORK

The proposed architecture is shown in Figure 1 and works only on below given three conditions:

Case 1: When the user fires the query and the corresponding data is not found in local database of a SE(s), then proposed architecture enable the user to search the result through other available search engine(s) via same interface.

Case 2: When case 1 fail then the role of Meta Search Engine (MSE) comes in light. To search the query user can select all the listed (say n) SEs concurrently. The top 10 results are considered as most important, so, top most ten common links from n-1 SE are considered as most important and displayed to the user.

Case 3: This case arises when user do not get any URL link, even from the MSE. When this happens proposed system will ask the user to provide his/her email id so that system can provide the required information in specified time period (i.e. given by user in hours). System finds the result and mails these results to user on his/her email id. User will get the results in his mail box only; if the results are found in specified time period. The result obtained from the MSE is also updated in the existing system’s database so that if any other user search for the same query it can be served without the use of Meta search engine.

The whole processes of giving query to getting results are organized in following modules:

Search Interface: Through this module user can give query and receive the results from the system.

Query Processor (QP): QP matches the query terms with the indexed data base of the SE and returns a list of matched documents.

Similarity Analyzer (SA): SA analyze the browsing behavior and clicked URLs stored in log files [13].

Rank Updater: This module is used to update the rank of clicked url’s corresponding to the queries in the cluster. Rank is updated using PR formula [14] with two assumptions. First, PR(v) denotes the sum of all rank score instead of rank score of page v.

$$PR(u) = (1-\mathrm{d}) \sum_{v \in B(u)} \frac{PR(v)}{N_V}$$

Similarly; N_v is number of occurrence of page v instead of number of outgoing links of page v.

Favored Query Finder and Recommender (FQFR): If a query contains important portion of the query then it is said to a favored query. The algorithm used for FQFR is same as used in [13] but with a difference in weight calculation i.e.

$$Wt = \frac{Occurrance.of.query.in.database}{Totalo.ccurance \cdot of.query.in.cluster}$$

Here, Occurrence of a query in database is used in place of no. of IP address which fired the query. Further, total occurrence of all the query in cluster is used instead of total number of IP address in that cluster.

Feedback form: This module comes in action when user neither gets the required information from SE nor from MSE. This module contains the information like query, mail id of user and number of days (hours in this paper), so that user required information can be mailed on his/her mail box. Whenever, the result of query is mailed to the user then the status of delivery time is set to 00:00:00 automatically.

V. EXPERIMENTAL SETUP AND DISCUSSION

To perform the experiment authors implement a SE and a MSE interface by including four SEs i.e. Google, Bing, Ask and AltaVista. As AltaVista is overtaken by yahoo so the results comes from yahoo SE instead of AltaVista. The proposed work was implemented in C# programming language. SQL server 2005 was used for database. The database used a query log was taken from the web site having URL: http://jeffhuang.com/search_query_logs.html. This database is of 2006 taken from query log of AOL SE. It consists of 386620 entries but authors take only 11599 entries for processing because of System configuration limitation.

The front end of the proposed work is shown in Figure 2. Here user can give his query and system will return the corresponding links (if available).

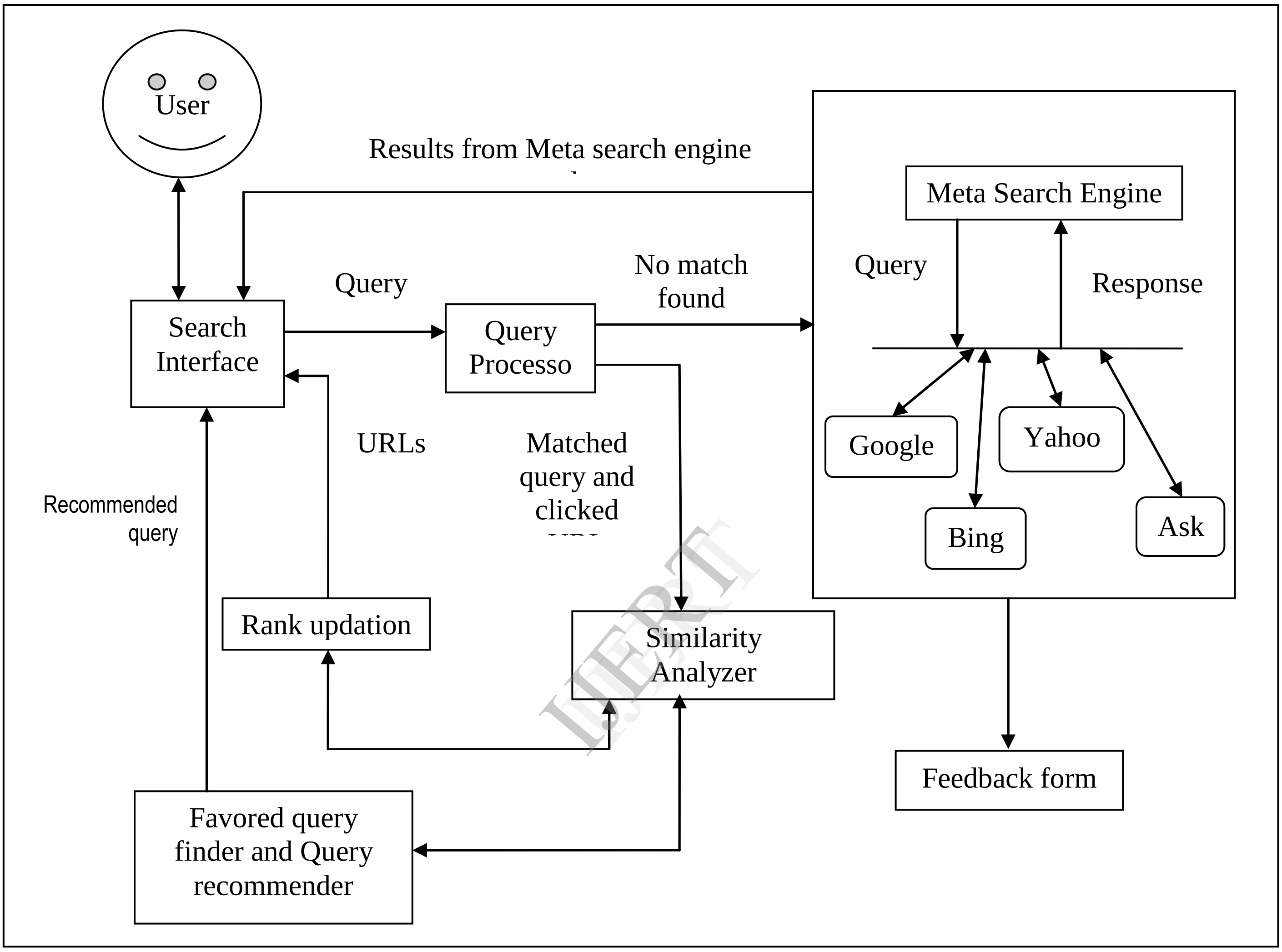
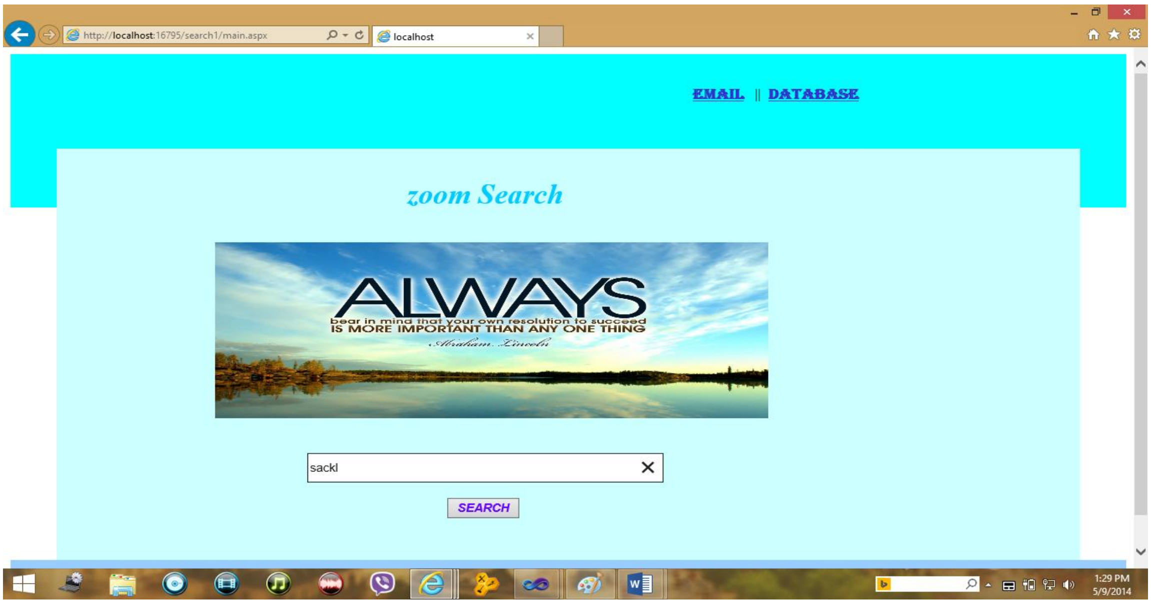


Figure 1: Proposed Architecture



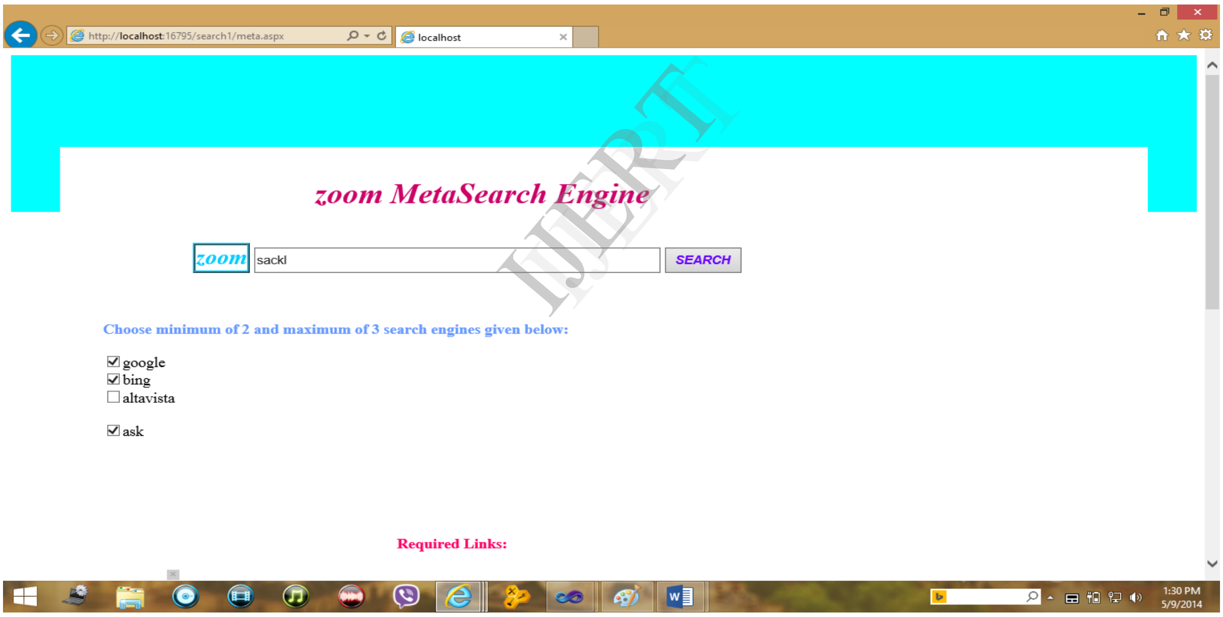
http://localhost:16795/search1/main.aspx
localhost
EMAIL || DATABASE

zoom Search

ALWAYS
bear in mind that your own resolution to succeed
IS MORE IMPORTANT THAN ANY ONE THING
Abraham Lincoln

sackl

Figure 2: Interface of Meta Search Engine



zoom MetaSearch Engine

sackl

Choose minimum of 2 and maximum of 3 search engines given below:

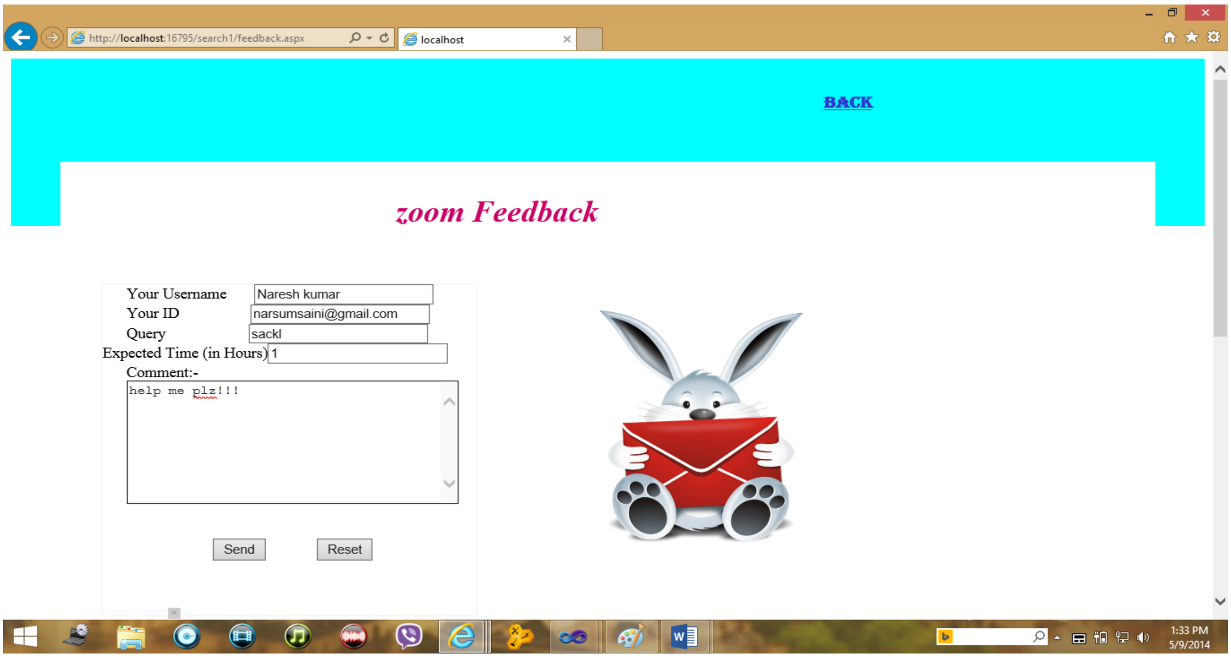
☑ google
☑ bing
□ altavista

☑ ask

Required Links:

Figure 3: Interface when query not found by search system

If the search interface doesnot get any link from the indexed data base then user is automatically switched to MSE interface. Figure 3 shows the interface of MSE where user have to select atleast three SE. Now this interface receive and merge the resultant links coming from the selected SEs. Then this merged list is presented to the user. If, in any how this interface doesnot get any URL from the selcetd SEs then user have to fillup the feedback form as shown in Figure 4

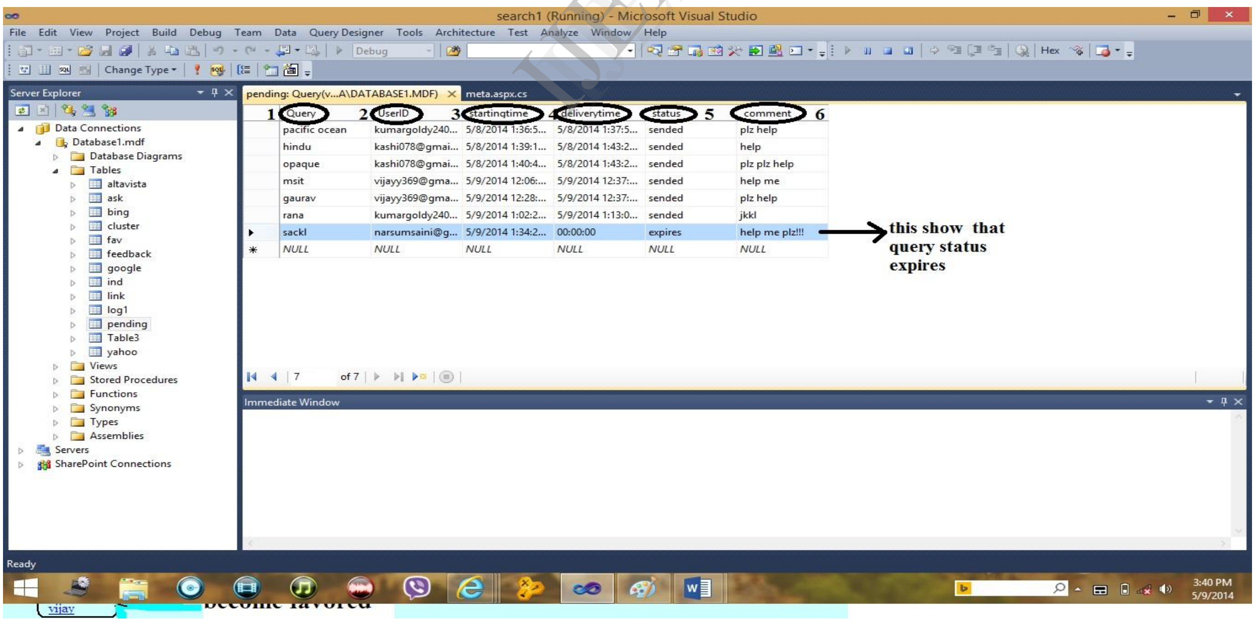


zoom Feedback

Your Username	Naresh kumar
Your ID	narsumsaini@gmail.com
Query	sackl
Expected Time (in Hours)	1
Comment:-	help me plz!!!

Figure 4: Feedback form

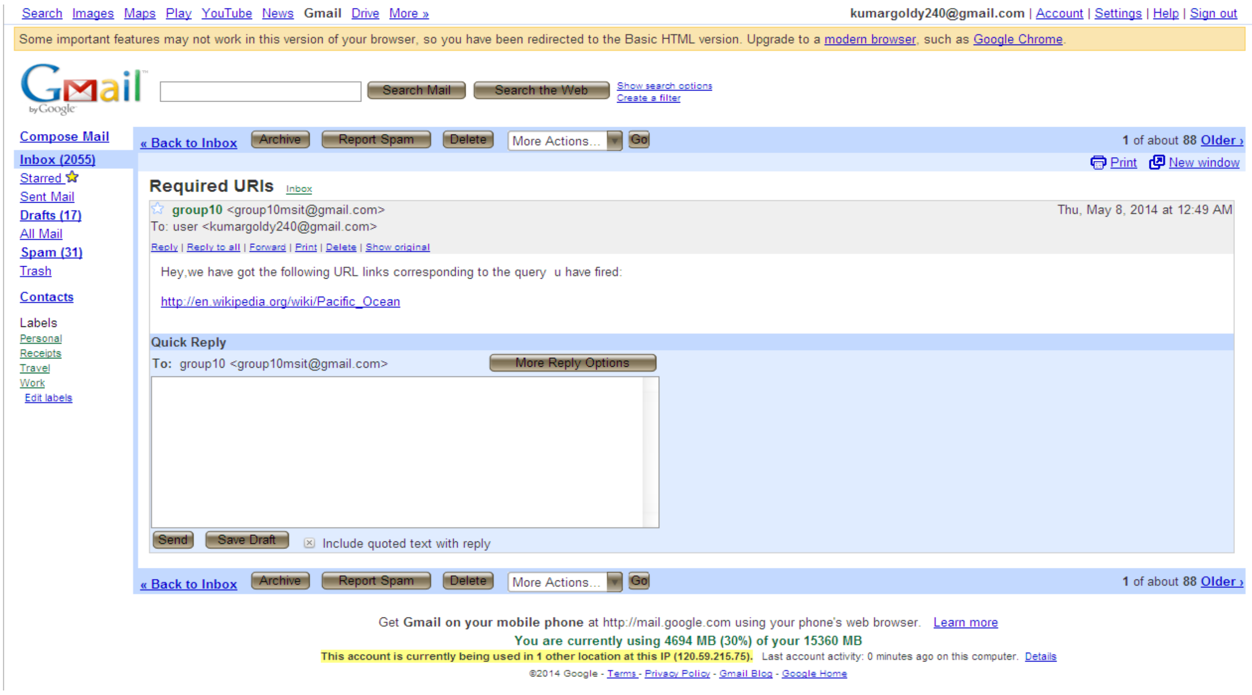
By clicking on the send button the data is stored in a separate database, and processed by the system in back end to find the necessary result. The Figure 5 shows the back end snap shot which shows the query to be search, user e-mail id, starting time, delivery time, status of query and comments.



	1 Query	2 UserID	3 startingtime	4 deliverytime	status 5	comment 6
	pacific ocean	kumargoldy240...	5/8/2014 1:36:5...	5/8/2014 1:37:5...	sended	plz help
	hindu	kashi078@gmai...	5/8/2014 1:39:1...	5/8/2014 1:43:2...	sended	help
	opaque	kashi078@gmai...	5/8/2014 1:40:4...	5/8/2014 1:43:2...	sended	plz plz help
	msit	vijayy369@gma...	5/9/2014 12:06:...	5/9/2014 12:37:...	sended	help me
	gaurav	vijayy369@gma...	5/9/2014 12:28:...	5/9/2014 12:37:...	sended	plz help
	rana	kumargoldy240...	5/9/2014 1:02:2...	5/9/2014 1:13:0...	sended	jkkl
▶	sackl	narsumsaini@g...	5/9/2014 1:34:2...	00:00:00	expires	help me plz!!!
✳	NULL	NULL	NULL	NULL	NULL	NULL

this show that query status expires

Figure 5: Status of queries



Some important features may not work in this version of your browser, so you have been redirected to the Basic HTML version. Upgrade to a modern browser, such as Google Chrome.

Required URIs Inbox

group10 <group10msit@gmail.com>
To: user <kumargoldy240@gmail.com>
Thu, May 8, 2014 at 12:49 AM

Hey,we have got the following URL links corresponding to the query u have fired:

http://en.wikipedia.org/wiki/Pacific_Ocean

To: group10 <group10msit@gmail.com>

Get **Gmail on your mobile phone** at http://mail.google.com using your phone's web browser.

You are currently using 4694 MB (30%) of your 15360 MB

This account is currently being used in 1 other location at this IP (120.59.215.75). Last account activity: 0 minutes ago on this computer.

Figure 6: Mail received by the user

Whenever any information is found in given time limit then it is forwarded to the user via his mail id. Figure 6 shows the snap shot of the mail received by the user.

VI. CONCLUSION

This paper has proposed a novel idea of searching the results on the Internet. The proposed approach has been implemented in C# programming language and has been tested. The experimental results have shown that the proposed approach provide better results than the existing search approaches of SE and MSEs. This system never frustrates the user even when user currently does not get any information on the spot. In this system user becomes sure that whenever search system get any information then it will be directly propagated to him/her via a simple mail. So, user has not been required to search his information in different time slots. The performance of proposed system in terms of time and space has been found a bit lesser as compared to existing search systems which is attributed to extra calculations done for switching to MSE interface and filling a feedback form. But the results achieved by the proposed system compensate the time limitations.

REFERENCES

[1] Rekha et. al., “Design of Query Suggestion using Rank Updater”, in International Journal of Computer Trends and Technology (IJCTT) , ISSN: 2231-5381, volume 11, number 5, pp. 220-227, May 2014.

[2] Hamada M.Zahera et. al.,” Query Recommendation for Improving” Search Engine Results in the proceedings of the World Congress on Engineering and Computer Science, ISSN: 2078-0966, 2010 Vol I, San Francisco, USA, 2010.

[3] A. Arasuat. Al., “Searching the Web,” ACM Transactions on Internet Technology, Vol. 1, No. 1, pp. 97-101, 2001

[4] M. Jansen et. al.,” Real life information retrieval: a study of user queries on the web”, ACM SIGIR Forum, Volume 32, Issue 1, pp. 5-17, 1998.

[5] Nikita Taneja et. al.,” Query Recommendation for Optimizing the Search Engine Results”, in International Journal of Computer Applications,ISSN: 0975 –8887) Volume 50, No.13, pp. 20-27, July 2012.

[6] Kajal Y.Vyas,” Improved Web Search Result Rank Optimization Using Search Engine Query Log”, in Journal Of Information, Knowledge And Research In Computer Engineering, ISSN: 0975 – 6760 volume – 02, ISSUE – 02 Pp. 433-436.

[7] L. Li et. al.,“Query Recommendation Using Large-Scale Web Access Logs and Web Page Archive,” in DEXA '08 Proceedings of the 19th international conference on Database and Expert Systems Application, ISBN: 978-3-540-85653-5, pp. 134–141, 2008.

[8] Thorsten Joachirns,"Optimizing search enginesusing clickthrough data," Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining, ISBN:1-58113-567-X, pp. 133-142, 2002.

[9] Neelam Duhan et. al.,"Rank Optimization and Query Recommendation in Search Engines using Web Log Mining Techniques," journal of computing, ISSN: 0975 – 8887, vol. 2,pp. 1-9, December 2010.

[10] Neha singh et. al.,” Query Recommendation employing Query Logs in Search Optimization”, in Int. J. Advanced Networking and Applications, ISSN : 0975-0290, Volume: 05, Issue: 03, pp:1917-1921, 2013.

[11] Osmar R. Zaïane at. al.,” Finding Similar Queries to Satisfy Searches based on Query Traces”, in proceeding of OOIS '02 Proceedings of the Workshops on Advances in Object-Oriented Information Systems, ISBN:3-540-44088-7, pp. 207-216, 2002.

[12] Geetanjli Gaur et. al.,” Query Recommendation Approach For Searching Database Using Search Engine”, in International Journal of Research in Engineering & Applied Sciences, ISSN: 2249-3905, Volume 3, Issue 3pp. 146-154, 2013.

[13] Neelam Duhan et. al.,”Rank Optimization and Query Recommendation in Search Engines using Web Log Mining Techniques”, in Journal Of Computing, Volume 2, Issue 12, ISSN 2151-9617,pp. 97-104, 2010, December 2010.

[14] Rekha Jain et. al.,” Page Ranking Algorithms for Web Mining”, in International Journal of Computer Applications, ISSN: 0975 – 8887, Volume 13– No.5, pp. 22-25, 2011.