A Novel Architecture for Search Engine using Domain Based Web Log Data

Prem Sharma
Computer Science and Engineering, Veer Madho Singh Bhandari Uttarakhand Technical University, India
premsagar1987@rediffmail.com

Divakar Yadav
School of Computer and Information Sciences, Indira Gandhi National Open University, India
divakaryadav@ignou.ac.in

Abstract: *Search engines, an information retrieval tool are the main source of information for users' information need now a day. For every query, the search engine explores its repository and/or indexer to find the relevant documents/URLs for that query. Page ranking algorithms rank the Uniform Resource Locator in abstract section (URLs) according to its relevancy with respect to users' query. It is analyzed that many of the queries fired by users on search engines are duplicate. There is a scope to improve the performance of search engine to reduce its efforts for duplicate queries. In this paper a proxy server is created that keep store the search results of user queries in web log. The proposed proxy server uses this web log to find results faster for duplicate queries fired next time. The proposed scheme has been tested and found prominent. The proposed architecture tested for ten duplicate user queries. it return all relevant web pages for duplicate user query (if query is found in web log at proxy server) from a particular domain instead of entire database. It reduces the perceived latency for duplicate query and also improves the value of precession and accuracy up to 81.8% and 99% respectively for all duplicate user queries.*

Keywords: *Search engine, information retrieval, web usage mining, content mining*.

Received February 14, 2021; accepted March 16, 2022
https://doi.org/10.34028/iajit/20/1/10

1. Introduction

The size of World Wide Web (WWW) is increasing day by day. Users of the WWW are highly dependent on search engines [4, 18, 38] to search information for their need. The time frame between clicking the search button after typing the queries and getting the desired results consists of a lot of different complicated small and heavy processes. Search engines usually have their own databases in which they save data that they acquire from the web crawlers [42] not only that, but this data is also then reformatted again and again and modified to be fast enough to be indexed as well as they remove any noise or redundancy from the data. Some search engines these days are also processing queries beforehand and rank the queries themselves on the basis of their relevance and increase the efficiency even before actually going to fetch the data for the user. Many search engines consider web as directional graphs and decide the relevance of a particular page depending upon the number of incoming and outgoing edges. Some ranking algorithms also take into consideration a page or topic's popularity into their consideration as well. Because of large number of queries, there is a challenge for search engines to respond to all the queries quick in time [31]. To solve this problem with search engines, many researchers has worked and provided different solutions. Web personalization, clustering of user queries, mining of query logs is the candidate solutions to handle this problem. Web personalization is a technique [12, 15, 25, 35] to personalize the World Wide Web according to user profiles. The different techniques of web personalization store user search patterns in history. Most of the time, there are similarities in search patterns of a particular user [20]. The web personalization algorithms do some processing on the queries fired by users and retrieve relevant results in quick time. In web personalization, web pages are considered as web objects and users are considered as subjects. The web personalization algorithms map the web objects with subjects. Better is the mapping between these two, better are the results of web personalization algorithm [17, 24, 31]. In this work Shou *et al*. [37] address a major issue in personalized search. The search engine [40] which uses Personalized Web Search (PWS) make user profiles. For creating user profiles, the search engines [10] fetch the information about the user. So, there is an issue of privacy protection in PWS systems. Ramitha and Jayasudha [26] proposed and discussed greedy approach that deal with the problem of privacy protection in PWS systems. There is a challenge to identify the users for web personalization algorithms. The users do not want to disclose their private information. So, the unwillingness of the user to disclose his or her private information influence the performance of PWS based search engines.

Search engines also store query logs and use this information to make the system faster. The results for similar quires may be retrieved from the query logs. Many researchers are currently working on query logs

to optimize the performance of search engines [9, 34] and information retrieval [3, 41, 43] systems for user queries.

2. Related Work

With the exponential growth of WWW and its users, it becomes very difficult to retrieve the information that is looked into by particular groups of users. For example, employees of an organization may need similar type of information. Therefore, a need for a mechanism is strongly felt that can personalize the contents of WWW according to the groups of users. The aim of various research papers in this literature is to identify similar URLs or web pages related to a specific domain [27, 29, 36]. The work in each research paper has different objectives and tries to apply on specific applications. In the previous work, most of the researcher used structural feature to identify the similar web pages [16, 23]. Uniform Resource Locator (URLs) or web pages, considered main parameters in structural features to classify the Website. The main objective of past research work is to grouping web pages based on the text they have. This method of clustering needs an impressive text processing methods and absolute retrieval of the page. In [6, 13, 30], an algorithm based on computation of similarity and limited features have been developed in ordered to express the resemblance between common text file. Al-Badarneh *et al*. [2] presents method to create sub clusters of the space and built Multi Small Index MSI i.e., the index for each sub cluster. It provides the quick response for user query.

Elmacioglu *et al*. [7] used clustering methods to distinguish between people's name on the web page grouping for particular application. In this method, a feature set received from the page content and URL is used as an input. This technique breaks the URL into various parts (e.g., domain name, path, parameters) and removes the properties of the page that have to be combined again and again and makes whole process very complicated.

In the context of clustering websites, Blanco *et al*. [5] emphasis to URLs instead of page content. A minimum description length based algorithm proposed Grünwald [8] that is used on URLs. The rules mining algorithm [1, 14] that can be applied only on URLs.

An algorithm for clustering of user queries suggested in paper [19]. The concept-based clustering was introduced that cluster the user queries. The algorithm solved the problem of short and ambiguous queries. The system generates personalized query clusters for the user queries.

In this paper Rodrigues *et al*. [28] has developed an efficient rule based model to identify the duplicate URLs that has similar text, in order to takes the benefit of a multi-sequence alignment strategy. In DUSTER model normalized rules have used to convert the distinct URLs in very precise manner that can detect duplicate URLs very easily. For training and testing the duplicate URLs were extracted from the TREC GOV2, and experimental results claimed the reduction of duplicate URLs up 80%. Mahafzah *et al*. [22] presents a new sampling technique (parameterized sampling) for Association Rule Mining (ARM), to improve the speed of ARM apply the mining technique on sample dataset instead of whole database.

This paper [12] presented an individual web revisitation method based on human's searched content history and keywords, known as “WebPagePrev”. In this method two probabilistic terms, context trees and term lists have been used to organized the context instances and page context respectively, which effectively developed by downfall and reinforcement with relevance feedback.

In this paper Hu *et al*. [11] have proposed query-URL click graph based location prediction method for Web pages. In this paper, the author has used the location vector to capture the location information of all terms (location and non-location), and proposed as automatic method to learn the captured rules of term location vector in order to improve the location prediction accuracy.

In this paper Liao *et al*. [21] make the use of task trail to optimize the performance of search engine. Task trail are the user activities such as query reformulation, URL clicked by the user. The task trail activities are very useful to map user queries with URLs which are relevant for the query. The worked on fetching the task trails from the web usage log and utilized it for optimizing the performance of search engines [21].

The Query Auto Completion (QAC) to make better queries by the user [39]. The QAC have an important role in making and storing query logs. Different users may write different queries when they are looking for the same data. QAC system resolve this issue. But there is a need to update the queries in the QAC on the basis of recent trends of the users.

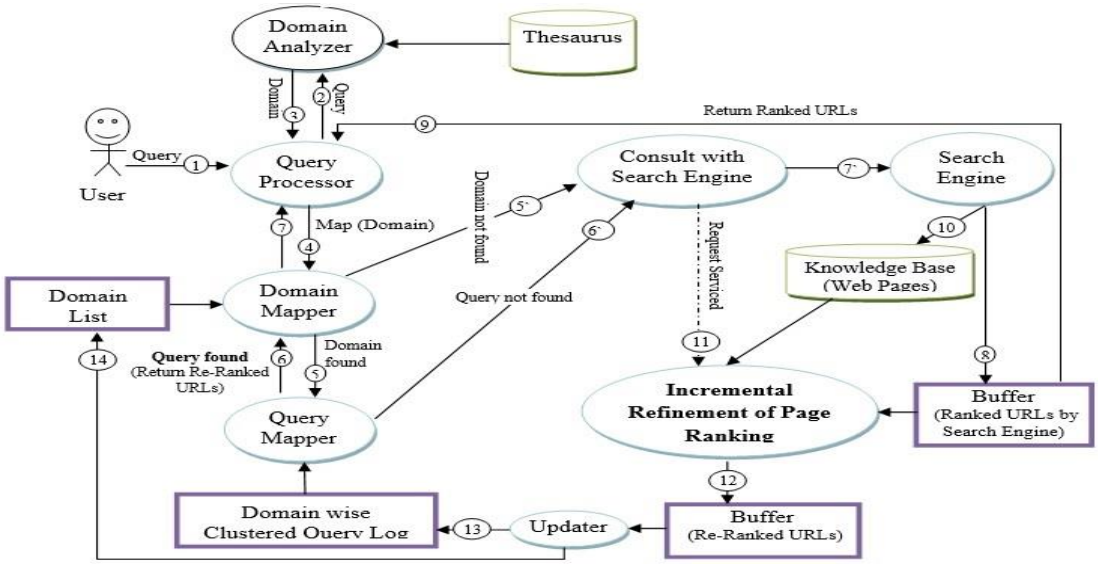

Figure 1. Architecture for search engine using domain based web log data.

3. Proposed Architecture

We proposed a mechanism to search repeated user queries in a specific domain instead of searching them in the entire repository and return relevant list of URLs. The proposed architecture given in Figure 1 for search engine using domain based web log data reduces the perceived latency for duplicate user query. The proposed architecture has following components as discussed below.

3.1. Query Interface

The query interface is used to make the interaction between the user and the system. User enters the query for his/her information need and press the search button. The query is then submitted to the search engine to make the information available. The system searches the information for the user query and return the results. The ranked URLs which are having the information related to user query are return back to the user at the search interface.

3.2. Search Engine

The search engine searches the information related to the user query and return the results in the form of list of URLs. The URLs are ranked on the basis of the relevancy of the information with user query. This system work like a normal search engine such as Google, Bing etc.

3.3. Domain Analyzer

When a query is fired then this query is passed to domain analyzer to find its domain. The domain analyzer makes use of Thesaurus to find domain for a query. It makes use of sense of query words and their synonyms to decide the domain for a user query. The working of this component if not in the scope of this paper but this component only assigns a domain name for the user query Working process of domain analyzer is given in Algorithm (1).

Algorithm(1): domainAnalyzer (String query)

```
{
Domain d1 = new Domain();
String [] words = query.split(" ");
int query_words_count = words.length;
for(int i=0;i<query_words_count;i++){
      System.out.print ("\n Query word " + (i+1) + "=" +
       words[i]);}
// find a domain in which maximum words of query found
String domain_selected = "No domain selected";
int maximum_matched_words_count = 0;
int i,j,n=this.list_of_domains.size();
for(i=0;i<n;i++)  {
    int mathced_word_count_for_this_domain =0;
Domain d = (Domain)this.list_of_domains.get(i);
// match words from query
int related_words_count = d.list_of_keywords.size();
for(j=0;j<related_words_count;j++)
{
  String related_word = (String)d.list_of_keywords.get(j);
   for(int k=0;k<query_words_count;k++)
      {
        String query_word = words[k];
         if(related_word.equalsIgnoreCase(query_word))
           {
              mathced_word_count_for_this_domain++;
           }
        }
    }
if(mathced_word_count_for_this_domain >=
                          maximum_matched_words_count)
{
 maximum_matched_words_count =
                    mathced_word_count_for_this_domain;
  d1 = d;  }
}
return d1;}
```

3.4. Domain List

The proposed system keep stores the list of all the domains created by the domain analyzer for the user

queries. If a new domain is assigned by the domain analyzer for a user query, then domain mapper updates the domain list by adding the new domain to the domain list (method is given Algorithm 2).

Algorithm (2): Create domain List

```
class Domain
{  String domain_id = new String();
   String domain_name = new String();
   ArrayList list_of_keywords = new ArrayList();
   Domain (String did,String dname, ArrayList keywords )
   {  this.domain_id = did;
      this.domain_name = dname;
      this.list_of_keywords = keywords;
   }
 }
```

3.5. Domain Wise Clustered Query Log

It is a repository which stores the results of the queries which are fired on the search interface in past. It has entry for every query. Every entry consists of a domain to which query belong, text of the query and list of URLs returned by the search engine when that queries was entered first time. This repository is used by the proxy server when the same query is fired again. If same query is fired again then proxy server returns the list of URLs as query result from this query log. In this way the time is saved because this time search engine is not used to search the information. Structure of query log is defined in Table 1.

Table 1. Structure of query log.

Domain	Query	List of URLs

```
class QueryLog()
{
   String query_string = new String();
   ArrayList list_of_urls = new ArrayList();
   Domain domain_of_the_query_object = new Domain();
}
```

3.6. Domain Mapper

This component searches the domain of the query in the domain list. If the domain of the query already exists in the domain list, then domain mapper passes the query with its domain to the Query Mapper. Otherwise it passes the query to the search engine for the results.

3.7. Query Mapper

Query mapper search the domain wise clustered query log for the input query. It searches the query only in the cluster storing results for queries for the given domain. By searching only one cluster make the working of the system fast. If same query is entered in past, then the list of URLs stored for that query become the result of the input query. These URLs are passed as final result for the input query at the search interface.

3.8. Incremental Web Page Ranking System

This system used to improve the quality of the ranking system [32, 33] on the basis of content of the web pages. It uses the knowledge base for improving the ranks and update the cluster query log with re-ranked list of URLs for the query. When a query is fired again then these URLs will be returned to the user by the Query Mapper.

3.9. Knowledge Base

This component stores the downloaded web pages for the URL list returned by the search engine for the user query. This repository is used by the incremental web page ranking system to re-rank the URL list by making use of content mining as specified in [32].

Flowchart for search engine using domain based web log data is given in Figure 2, and working process of different components are define in Algorithm (3) as given below:

Algorithm (3): queryProcessor (query)

```
{
  Query=getQuery(user);
  Domain=domainAnalyzer(thesaurus);
if(mapDomain(Domain, Domain List))
 {  If(mapQuery(Domain, Query, Query Log))
       Return (Re-ranked URLs as a result)
    else
       Consult(Query, Search Engine);
       //Defined in algorithm-4 }
  else
      Consult(Query, Search Engine)
  }
```

Algorithm (4): Consult (Query, Search Engine)

```
{
return (ranked URLs as a result)
knowledgebase=downloadWebpage(URLs)
Wait (request serviced)
 re-rankedURLs_List=IRPRanking(knowledgebase,URLs)
 signal(update domain_List)
 update_query_Log();     // Defined in algorithm-5
 signal(update queryLog)
 update_domain_List();     // Defined in algorithm-6
 }
```

Algorithm (5): Update_query_Log( )

```
{
while (1){
Wait (update_query_log)
Updater(URL_Bubber, Query_Log, domain, Query);
                        // Set re-ranked URLs from URL buffer
with user query in specified domain.     }
}
```

Algorithm (6): Update_domain _List( )

```
{ while (1)
        { Wait (update_domain_list)
          Updater(domain,domain_List);
// Set domain in domain List.}
}
```

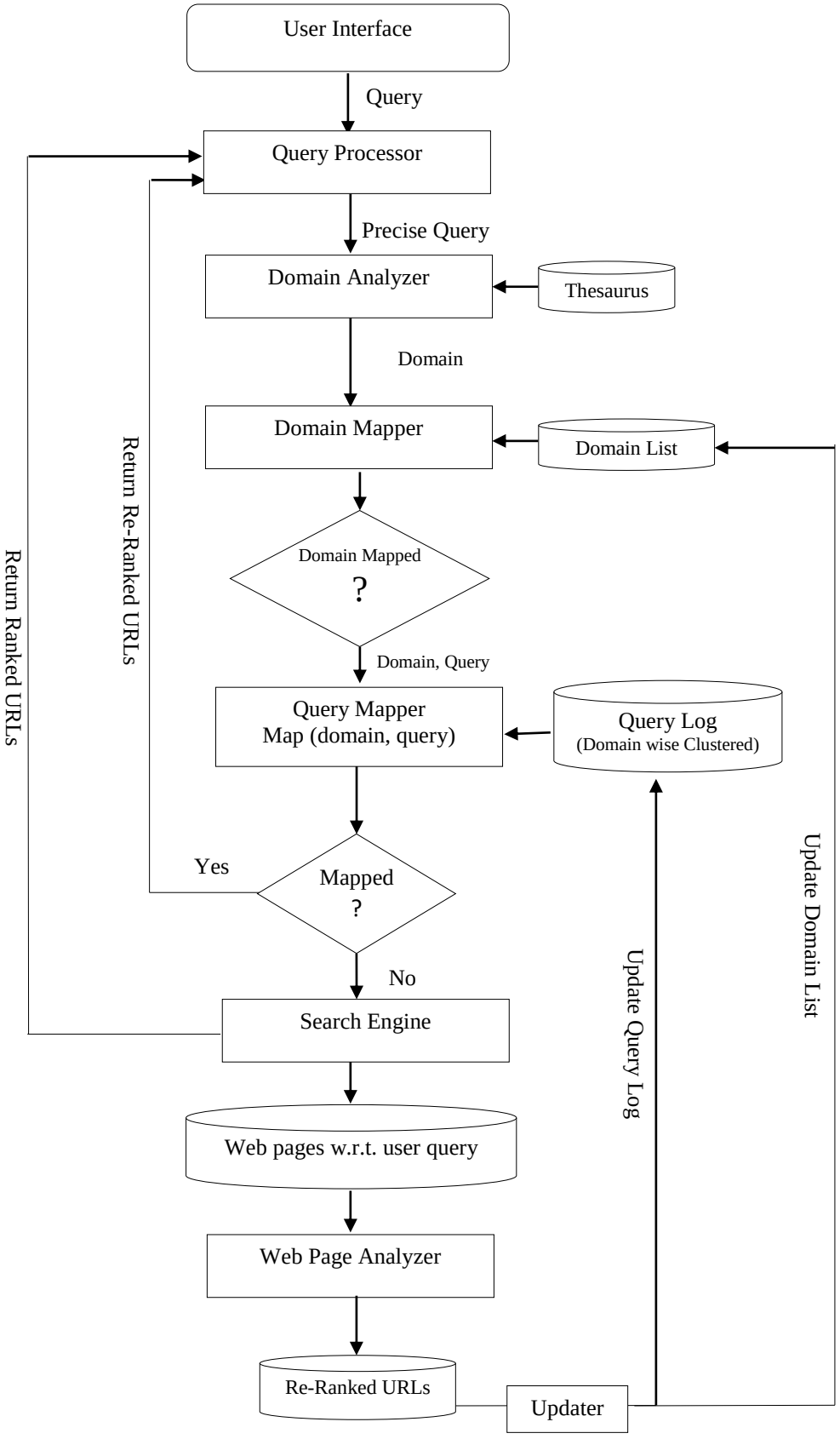

Figure 2. Flowchart for search engine using domain based web log data.

4. Experimental Results

We fired hundreds of queries regarding different fields. Domain analyzer identify domain name for corresponding user queries. So all these queries have been stored domain wise in query log to improve the searching time (reduced the perceived latency for duplicate queries).

Table 2. List question from different domain.

QID	Query Statement
1	Who among the following was the founder of Sanskrit College at Banaras?
2	Why India's education system is not changing?
3	Would privatization of education not commercialize education?
4	E learning in higher education in India
5	higher education system in india
6	higher education in india challenges and prospects
7	primary education in india
8	challenges of primary education in india
9	top 50 engineering colleges in india for computer science
10	Engineering Colleges for B. Tech. in India
11	Will a person with Type 2 diabetes under control end up with the need for insulin
12	What medications are best for the treatment of asthma?
13	list out the symptoms of diabetes
14	What causes Hashimoto's thyroiditis
15	What is Ebola virus disease?
16	Which player has made highest numbers of runs in IPL history?
17	Name the first Indian batsman to score 1,000 runs, in which most runs are in a calendar year and most fifties in the format
18	When did Virat Kohli became India's Test captain
19	Why do Co-operative Housing Societies collect a Sinking Fund?
20	Are there any benefits in converting a leasehold property to a freehold one?

Domain analyzer, analyzes each query as given in input Table 2, finds their domain name on the basis of query keywords and mapped with domain list. If it is mapped then add the <query ID, domain> in to domain list otherwise updated domain list and then add the <query ID, domain> in to domain list. The results of domain analyzer are shows in Table 3:

Table 3. Results generated by domain analyzer.

QID	Domain	QID	Domain
1	Education	**11**	Medical
2	Education	**12**	Medical
3	Education	**13**	Medical
4	Education	**14**	Medical
5	Education	**15**	Medical
6	Education	**16**	Sports
7	Education	**17**	Sports
8	Education	**18**	Real estate
9	Education	**19**	Real estate
10	Education	**20**	Real estate

Table 4. The value of the parameters for various queries in different domains.

QID	Domain	NWPQ	NWPD	TR
1	Education	3420000	13790000000	3420000
2	Education	308000000	13790000000	308000000
3	Education	657000	13790000000	657000
4	Education	185000000	13790000000	185000000
5	Education	318000000	13790000000	318000000
6	Education	147000000	13790000000	147000000
7	Education	310000000	13790000000	310000000
8	Education	109000000	13790000000	109000000
9	Education	22400000	13790000000	22400000
10	Education	245000000	13790000000	245000000
11	Medical	43300000	8440000000	43300000
12	Medical	108000000	8440000000	108000000
13	Medical	193000000	8440000000	193000000
14	Medical	519000	8440000000	519000
15	Medical	19400000	23710000000	19400000
16	Sports	1920000	23710000000	1920000
17	Sports	314000	23710000000	314000
18	Realestate	6700000	668000000	6700000
19	Realestate	9080000	668000000	9080000
20	Realestate	7840000	668000000	7840000

The value of the following parameters like Number of relative web pages corresponding to the user Query (NWPQ), Total number of web pages in web repository corresponding to the specific domain (NWPD), Number of Web pages return by Google Search Engine for a user Query (NWPRQ) and Total Relevant Web pages for user query in web repository (TR) are given in Table 4.

$$TR = NWPQ$$

- TP: True Positive

$$TP = NWPRQ$$

- FN: False Negative

$$FN = TR\text{-}TP$$

- FP: False Positive-Wrong prediction i.e., predicted-positive, actual-negative
- TN: True Negative

$$TN = NWPD\text{-}(TR+FP)$$

Derived the values of TP, FN, FP, and TN (shown in Table 5) using the values Table 4.

Table 5. Derived the values of TP, FN, FP, and TN.

QID	TP	FN	FP	TN
1	250	3419750	50	13786579950
2	250	307999750	50	13481999950
3	220	656780	50	13789342950
4	220	184999780	50	13604999950
5	220	317999780	50	13471999950
6	220	146999780	50	13642999950
7	220	309999780	50	13479999950
8	220	108999780	50	13680999950
9	220	22399780	50	13767599950
10	220	244999780	50	13544999950
11	220	43299780	50	8396699950
12	220	107999780	50	8331999950
13	220	192999780	50	8246999950
14	220	518780	50	8439480950
15	220	19399780	50	23690599950
16	220	1919780	50	23708079950
17	220	313780	50	23709685950
18	220	6699780	70	661299930
19	220	9079780	80	658919920
20	210	7839790	50	660159950

Precession = TP/(TP+FP)
Recall=TP/TP+FN)
Accuracy=TP+TN/TP+FP+FN+TN

Result of domain analyzer for each query <query ID, domain> (shown in Table 3) fired on proposed modal. It computes the relevancy of results for each query on the basis of different parameters like precision, recall and accuracy (use Table 5) as shown in Table 6.

Table 6. Precision, recall and accuracy of proposed method for different queries from different domain.

QID	Domain	Precision	Recall	Accuracy
1	Education	0.833333333	0.0000731	0.999752009
2	Education	0.833333333	0.0000008	0.977664989
3	Education	0.814814815	0.0003349	0.999952369
4	Education	0.814814815	0.0000012	0.986584494
5	Education	0.814814815	0.0000007	0.976939824
6	Education	0.814814815	0.0000015	0.989340114
7	Education	0.814814815	0.0000007	0.977519954
8	Education	0.814814815	0.0000020	0.992095734
9	Education	0.814814815	0.0000098	0.998375647
10	Education	0.814814815	0.0000009	0.982233515
11	Medical	0.814814815	0.0000051	0.994869688
12	Medical	0.814814815	0.0000020	0.987203812
13	Medical	0.814814815	0.0000011	0.977132722
14	Medical	0.814814815	0.0004239	0.999938527
15	Medical	0.814814815	0.0000113	0.999181787
16	Sports	0.814814815	0.0001146	0.999919029
17	Sports	0.814814815	0.0007006	0.999986764
18	Realestate	0.75862069	0.0000328	0.989970284
19	Realestate	0.733333333	0.0000242	0.986407395
20	Realestate	0.807692308	0.0000268	0.988263713

Evaluate the average value of precision, recall, F-measure and accuracy of proposed method (shown in Table 7 for different domain using Table 6.

Table 7. The average value of precision, recall, F-measure and accuracy of proposed method for different domain.

Domain	Precision	Recall	Accuracy
Education	0.818518519	0.000042559	0.9880
Medical	0.814814815	0.000088698	0.9917
Sports	0.814814815	0.000407610	0.9950
Realestate	0.766548777	0.000027950	0.9882

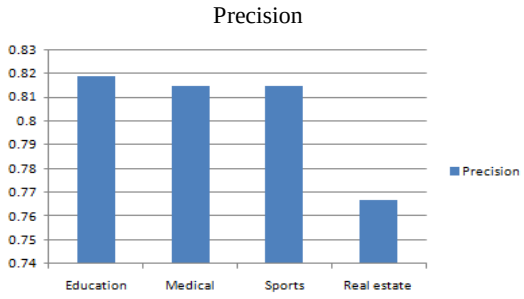

Figure 3. Precision value for specific query domains.

Figure 3, Shows the precision value for specific query domains based on Table 7. The precision value for different query domain is calculated by using Equation (1) as given below.

$$Precision = \frac{TP}{(TP+FP)} \quad (1)$$

- Where:
  - TP (True Positive): relevant web pages return by search engine for specific queries in a domain.
  - FP (False Positive): non-relevant web pages return by search engine for specific queries in a domain.

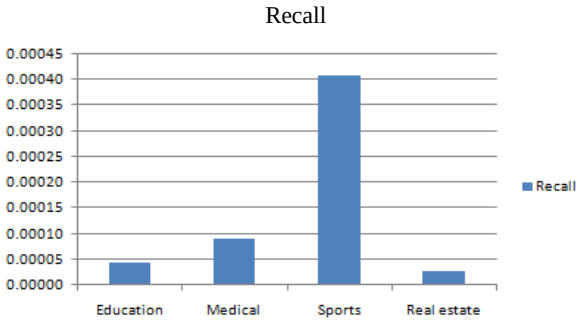

Figure 4. Recall values for specific query domains.

Figure 4, Shows the recall value for specific query domains based on Table 7. The precision value for different query domain is calculated by using Equation (2) as given below.

$$Recall = \frac{TP}{(TP+FN)} \quad (2)$$

Where:

- TP (True Positive): relevant web pages return by search engine for specific queries in a domain.
- FN (False Positive): all relevant web pages which are not return by search engine for specific queries in a domain.

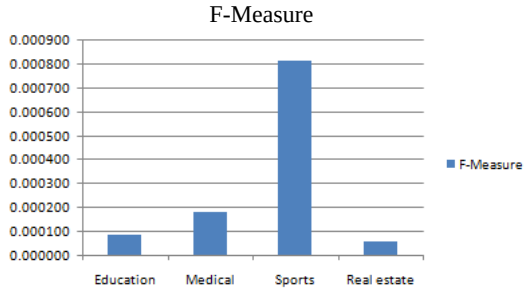

Figure 5. F-measure values for specific query domains.

Figure 5, Shows the F-measure value for specific query domains based on Table 7. The precision value for different query domain is calculated by using Equation (3) as given below.

$$F - measure = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall}\right) \quad (3)$$

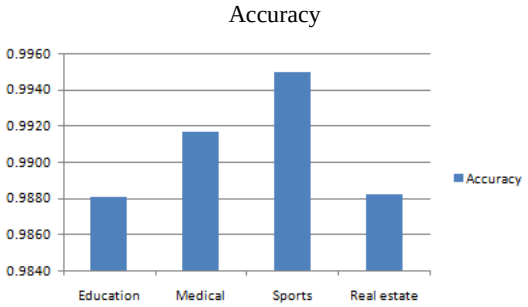

Figure 6. Accuracy for duplicate queries in specific domain.

Figure 6, Shows the Accuracy for specific query domains based on Table 7. The Accuracy for different

domains is calculated by using Equation (4) as given below.

$$Accuracy = \left(\frac{TP+TN}{\text{TP+FP+FN+TN}}\right) \quad (4)$$

5. Conclusions and Future Scope

In this work a novel architecture for improving the performance of search engine is proposed. The proposed system creates a web log of all the queries with their results and use this web log to retrieve the results for repeated queries. It divides the web log into a number domains and group the queries according the domain to which the query belongs. When a new query is entered then domain analyzer is used to find domain of this query and query log is searched only for those queries which have the same domain i.e. domain of the input query. This improvement reduces the search time of the repeated query in web log. It is concluded that if the queries fired by users are repeated then the results for these queries are found in web log and returned in less time as compared to those queries which are fired first time. It is also improve the value of precession up to 81.8% and accuracy up to 99% for all duplicate user queries. However, there are some future scopes of this work which are listed below:

1. Work can be done in future to implement this architecture and test it on real data.
2. The working and performance of the domain analyzer can be explored in more detail and its performance can be tested.
3. The performance of the proposed architecture can be calculated and compared with other existing techniques.

References

[1] Agarwal A., Koppula H., Leela K., Chitrapura K., Garg S., and GM P., "URL Normalization for De-Duplication of Web Pages," *in Proceedings of the 18th ACM Conference on Information and Knowledge Managemen*, Hong Kong, pp. 1987-1990, 2009.

[2] Al-Badarneh A., Al-Alaj A., and Mahafzah B., "Multi Small Index (MSI): A Spatial Indexing Structure," *Journal of Information Science*, vol. 39, no. 5, pp. 643-660, 2013.

[3] Aqla H., Ahmed S., and Danti A., "Death Prediction and Analysis Using Web Mining Techniques," *in Proceedings of 4th International Conference on Advanced Computing and Communication* Systems, Coimbatore, pp. 1-5, 2017.

[4] Bidoki A. and Yazdani N., "Distancerank: an Intelligent Ranking Algorithm for Web Pages," *Information Processing and Management*, vol. 44, no. 2, pp. 877-892, 2008.

[5] Blanco L., Dalvi N., and Machanavajjhala A., "Highly Efficient Algorithms for Structural Clustering of Large Websites," *in Proceedings of the 20th International Conference on World Wide Web*, New York, pp. 437-446, 2011.

[6] Broder A., Glassman S., Manasse M., and Zweig G., "Syntactic Clustering of the Web," *Computer Networks and ISDN Systems*, vol. 29, no. 8, pp. 1157-1166, 1997.

[7] Elmacioglu E., Tan Y., Yan S., Kan M., and Lee D., "PSNUS: Web People Name Disambiguation By Simple Clustering With Rich Features," *in Proceedings of the 4th International Workshop on Semantic Evaluations*, Prague, pp. 268-271, 2007.

[8] Grünwald P., *The Minimum Description Length Principle*, MIT Press, 2007.

[9] Gupta P., Singh S., Yadav D., and Sharma A., "An Improved Approach to Rank Web Document," *Journal of Information Processing Systems*, vol. 9, no. 2, pp. 217-236, 2013.

[10] Gupta P., Sharma A., and Yadav D., "A Novel Technique for Back-link Extraction and Relevance Evaluation," *International Journal of Computer Science and Information Technology*, vol. 3, no. 3, pp. 227-238, 2011.

[11] Hu Y., Kang C., Tang J., Yin D., and Chang Y., "Large-Scale Location Prediction for Web Pages," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 9, pp. 1902-1915, 2017.

[12] Jin L., Feng L., Liu G., and Wang C., "Personal Web Revisitation by Context and Content Keywords with Relevance Feedback," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 7, pp. 1508-1521, 2017.

[13] Kakol M., Nielek R., and Wierzbicki A., "Understanding and Predicting Web Content Credibility Using the Content Credibility Corpus," *Information Processing and Management*, vol. 53, no. 5, pp. 1043-1061, 2017.

[14] Kheir N., Blanc G., Debar H., Garcia-Alfaro J., and Yang D., "Automated Classification of C&C Connections through Malware URL Clustering Nizar," *IFIP Advances in Information and Communication Technology*, vol. 455, pp. 252-266, 2015.

[15] Khribi M., Jemni M., and Nasraoui O., "Automatic Recommendations for E-Learning Personalization Based on Web," *Educational Technology and Society*, vol. 12, no. 4, pp. 30-42, 2009.

[16] Kim S. and Kang J., "Analyzing The

Discriminative Attributes of Products Using Text Mining Focused on Cosmetic Reviews," *Information Processing and Management*, vol. 54, no. 6, pp. 938-957, 2018.

[17] Kleinberg J., "Authoritative Sources in A Hyperlinked Environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604-632, 1999.

[18] Lee L., Jiang J., Wu C., and Lee S., "A Query-Dependent Ranking Approach for Search Engines," *in Processing of 2nd International Workshop on Computer Science and Engineering*, Qingdao, pp. 259-263, 2009.

[19] Leung K., Ng W., and Lee D., "Personalized Concept-Based Clustering of Search Engine Queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 11, pp. 1505-1518, 2008.

[20] Li L., Xu G., Zhang Y., and Kitsuregawa M., "Random Walk Based Rank Aggregation to Improving Web Search," *Knowledge-Based Systems*, vol. 24, no. 7, pp. 943-951, 2011.

[21] Liao Z., Song Y., Huang Y., He L., and He Q., "Task Trail: An Effective Segmentation of User Search Behavior," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 12, pp. 3090-3102, 2014.

[22] Mahafzah B., Al-Badarneh A., and Zakaria M., "A New Sampling Technique for Association Rule Mining," *Journal of Information Science*, vol. 35, no. 3, pp. 358-376, 2009.

[23] Moreno M., Segrera S.,López V., Muñoz M., and Sánchez A., "Web Mining Based Framework for Solving Usual Problems in Recommender Systems: A Case Study for Movies Recommendation," *Neurocomputing*, vol. 176, pp. 72-80, 2016.

[24] Nguyen T., Lu H., and Lu J., "Web-Page Recommendation Based on Web Usage and Domain Knowledge," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, pp. 2574-2587, 2014.

[25] Patil S. and Sarkar S., "Personalized Web Page Recommendation Using Ontology," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3, no. 7, pp. 4431-4436, 2015.

[26] Ramitha A. and Jayasudha J., "Personalization and Privacy in Profile-Based Web Search," *in Processing of International Conference on Research Advances in Integrated Navigation Systems*, Bangalore, pp. 1-4, 2016.

[27] Rizvi N. and Keole R., "Web Page Recommendation in Information Retrieval using Domain Knowledge and Web Usage Mining," *International Journal of Science, Engineering and Technology Research*, vol. 4, no. 5, pp. 1531-1535 2015.

[28] Rodrigues K., Cristo M., Moura E., and Silva A., "Removing DUST Using Multiple Alignment of Sequences," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 8, pp. 2261-2274, 2015.

[29] Roobam R. and Vallimayli V., "Survey on Ontology based Semantic Web Usage Mining for Enhanced Recommendation Model," *International Journal of Scientific and Engineering Research*, vol. 5, no. 12, pp. 1164-1170, 2014.

[30] Sharma D. and Ganeshiya D., "HierarchicalRank: Webpage Rank Improvement Using HTML TagLevel Similarity," *The International Arab Journal of Information Technology*, vol. 15, no. 3, pp. 485- 492, 2018.

[31] Sharma D. and Sharma A., "A Comparative Analysis of Web Page Ranking Algorithms," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 2, no. 8, pp. 2670-2676, 2010.

[32] Sharma P. and Yadav D., "Incremental Refinement of Page Ranking of Web Pages," *International Journal of Information Retrieval Research*, vol. 11, no. 2, pp. 57-73, 2020.

[33] Sharma P., Sharma A., and Garg P., "Design of a Framework for Knowledge Based Web Page Ranking," *International Journal of Engineering and Technology*, vol. 9, no. 3, pp. 2236-2244, 2017.

[34] Sharma P., Yadav D., and Garg P., "A Systematic Review on Page Ranking Algorithms," *International Journal of Information Technology*, vol. 12, pp. 329-337 2020.

[35] Sharma S. and Lodhi S., "Development of Decision Tree Algorithm for Mining Web Data Stream," *International Journal of Computer Applications*, vol. 138, no. 2, pp. 34-43, 2016.

[36] Shirgave S., Kulkarni P., and Borges J., "Semantically Enriched Web Usage Mining for Personalization," *International Journal of Computer and Information Engineering*, vol. 8, no. 1, pp. 249- 257, 2014.

[37] Shou L., Bai H., Chen K., Chen G., "Supporting Privacy Protection in Personalized Web Search," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 453-467, 2014.

[38] Vojnovic M., Cruise J., Gunawardena D., and Marbach P., "Ranking and Suggesting Popular Items," *IEEE Transactions on Knowledge and*

Data Engineering, vol. 21, no. 8, pp. 1133-1146, 2009.

[39] Wang Y., Ouyang H., Deng H., and Chang Y., "Learning Online Trends for Interactive Query Auto-Completion," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 11, pp. 2442-2454, 2017.

[40] Yadav A. and Yadav D., "Wavelet Tree based Dual Indexing Technique for Geographical Search," *The International Arab Journal of Information Technology*, vol. 16, no. 6, pp. 624-632, 2019.

[41] Yadav A., Yadav D., and Prasad R., "Efficient Textual Web Retrieval using Wavelet Tree *International Journal of Information Retrieval Research*, vol. 6, no. 4, pp. 16-29, 2016.

[42] Yadav D., Sharma A., Sanchez-Cuadrado S., and Morato J., "An Approach to Design Incremental Parallel WebCrawler," *Journal of Theoretical and Applied Information Technology*, vol. 43, no. 1, pp. 8-29, 2012.

[43] Yadav D., Sharma A., and Gupta J., "Topical Web Crawling Using Weighted Anchor Text and Web Page Change Detection Techniques," *WSEAS Transactions on Information Science and Applications*, vol. 6, no. 2, pp. 263-275, 2009.

Prem Sharma working as Assistant Professor, G. L. Bajaj Institute of Technology and Management Greater Noida (UP), India. He is pursuing Ph.D. (CSE) from Veer Madho Singh Bhandari Uttarakhand Technical University, Dehradun, India. He did his B. Sc. (PCM) from Agra University in 2007, Master of Computer Applications (MCA) from IIMT Engineering College Meerut (UP), Affiliated to UPTU Lucknow in 2010. Further, he did his Master of Technology in Computer Science and Engineering from Shobhit University Meerut in 2012. He supervised 05 M.Tech. dissertations and published more than 20 papers in international journals and conference proceedings. His area of research includes Information Retrieval, Crawler, Machine Learning, Web mining.

Divakar Yadav is working as Professor in School of Computer and Information Sciences (SOCIS), Indira Gandhi National Open University (IGNOU), New Delhi, India. He did his PDF from UC3M (Spain), PhD in CSE (2010), M.Tech. from IIIT Allahabad (2005) and B.Tech. from IET Lucknow (1999). He supervised 07 PhD, 31 M.Tech. dissertations and published more than 125 papers in international journals and conference proceedings. His area of research includes Information Retrieval, Machine Learning, Soft-computing and E-learning.