ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (985-992), Month: April - June 2015, Available at: www.researchpublish.com

QUERY RECOMMENDATION WITH SIMILARITY IMPROVEMENT IN QUERY LOG

[1]Ritu Maheshwari Bansal, [2]Meenakshi Bhdadana

[1]Assistant Professor (CSE), Faculty of Engineering & Technology Engineering, MRIU, Faridabad, Haryana, India
[2]Research Scholar, M.Tech (CSE), MRIU, Faridabad, Haryana, India

Abstract: **When a user input a query, intelligent search engine can suggest a list of related queries. Query recommendation is the method to improve the search results on web. This paper presents the method of mining the search engine query log to get the fast query Recommendation from large scale. In this, a formula is applied to fast recommend the most related queries for the user with useful information. For this, technology used for allowing query recommendation is query log which contains the attributes like query name, document which contains term occurred in the document , normalization factor, inverse document frequency and it also helps in minimize the retrieval time of user submitted query. As a result it shows the user most relevant queries to make user to find the query easily and satisfy their needs.**

Keywords: **Query log, Search engine, Clustering, Query similarity, Information retrieval (IR), Document frequency (df), Term frequency (tf), Inverse document frequency (idf), Normalization.**

I. INTRODUCTION

Nowadays the popularity of internet increases day by day so it is difficult to extract the relevant information from the large volume of data. The user face the difficulty in searching the desired information from the search engine like google, yahoo, etc. The is a lot of difference between the search engine and the query recommendation such as the user search the query from the search engine if it knows exactly what to fired to satisfy the user needs. The user prefers to use the recommendation system if they do not know from where exactly they can get their query solution and the appropriate wording to fire the query[5].

Thus improving user satisfaction is a key challenge for web search engines. For this purpose search engine provide way to the users to specify their information need more clearly in the form of queries simply as list of keywords or phrases. Many search engine companies apply significant efforts to develop means that correctly guess by which hidden intent the user has submitted the given query. In the recent years, web search engines have started to provide users with query recommendation to help them reformulate queries and quickly satisfy their needs[4]. In this paper, we follow the approach of query recommendation with similarity improvement in query log. When the visitor search query in search engine then the search engine result list get displayed on the screen. Results are according to improved similarity, that search engine is following. The information retrieval technique is the most popular technique used for the most relevant information retrieval. It is very much crucial to get most appropriate queries when the user enter the query. The work proposed in this paper aims to optimized the results of a search engine by returning the more relevant and user most wanted pages on the top of search result list. To achieve the required task, the approach pre-mines the query logs to retrieve the potential clusters of queries followed by finding the most popular queries in all clusters. The outputs of both mining processes are utilized to return relevant pages to the user while recommending him with popular historical queries.

The organization of this paper is as follows: In Section II, we provide a related work. In section III, we propose the concept of similarity. In section IV, we discuss the overview of the query log .Then, we discuss the proposed formula and the query clustering and the in section V.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (985-992), Month: April - June 2015, Available at: www.researchpublish.com

II. RELATED WORK

A number of researchers have discussed the problem of finding relevant search results from the search engines. Relevant query recommendation research is mainly based on previous query log of the search engine, which contains the history of submitted query and the similarity of keywords[4]. and in this we used a information retrieval method by using term frequency, document frequency ,inverse document frequency and the normalization factor for fast retrieval of result that means time reduction of user query retrieval reduces by using this method the formula which used in this paper helps in improving the efficiency, helps in mining of query log for giving a relevant result and reduces the query retrieval time. This formula includes many important factors on the basis of similarity score will be calculated which will be used in the query log. The factor included in this formula are similarity on the basis of keywords, weight factor which in turn includes the tf term frequency that is the number of time the term occurred in the document, df is the document frequency that is the number of documents in which that frequency is related, idf is the inverse of document frequency and the normalization factor is also included in this formula. By using this method or concept of information relevancy we are able to get the relevant information in minimum time. The resulting query log helps the user to find the relevant query easily and quickly. The history of queries stored in the query log helps the user. This method searches the related query based on the input query while the user searches so he can build a proper search query with the knowledge domain terminology which is important for search engine to get the related results[4].

III. PROPOSED WORK

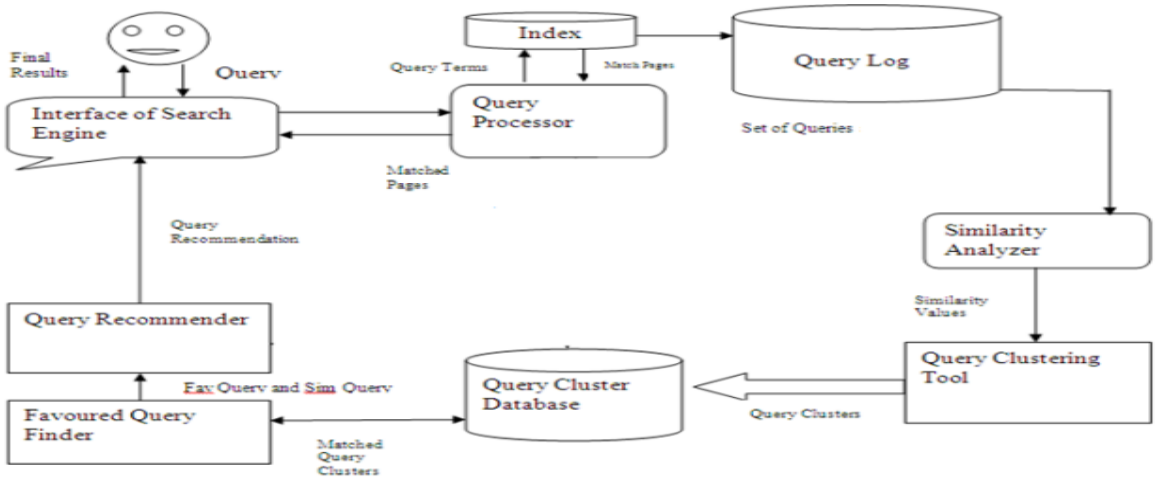
The proposed optimization system lying on learning from historical query logs is proposed to calculate user's information requirements in a better way. The proposed system works as follow. The prime feature of the system is to perform query clustering by finding the similarities between the two queries, which is based on designed formula and it includes user query keywords, Normalization factor and the idf which is inverse document frequency. It includes the term frequency which means the number of times the term occurring in the document, the document frequency are the documents in which the number of terms are related.idf is the inverse document is the rare terms of the documents. The tf ,idf and df are the terms used in the information relevance method. It is basically the vector space model of the information retrieval. It is basically used for finding the similarity by using the various formaula like the cosine similarity metrics. By this method, the time user spends looking for the required information from search result list can be reduced and the more relevant Web pages can be obtained facilitate .The proposed architecture of optimization system is shown in Fig.1 which consists of following functional components:

1. Query Log
2. Query Similarity

2.1 Based on the keywords

2.2 Based on the normalization factor and The tf,idf.

3. Query Clustering Tool
4. Favored Query finder
5. Query recommender.



ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (985-992), Month: April - June 2015, Available at: www.researchpublish.com

IV. QUERY LOG

In this section we discuss the query log which contain the historical data of the search engine. It is a popular data source of query recommendation .the query log is generally a repository of search engine data. It has a complete record of what user search in and what time frame. Depending on the specifics of how the data is collected, typically logs of search engines include the following entries When user submits a query on the interface of search engine, the query processor component matches the query terms with the index repository of the search engine and taking a list of matched documents in reply. On the reverse order, result optimization system performs its task of gathering user intentions from the query logs. The user browsing behavior as well as the submitted queries and clicked URLs get stored in the logs and are analyzed continuously by the Similarity Analyzer module, the output of which is forwarded to the Query Clustering Tool to create potential groups of queries based on their similarities[4]Then the clusters are stored in query cluster database. Then the favored query finder find out the relevant query from the database. The query recommender recommends the similar query.

1. User IDs,

2. Query q issued by the user,

3. Document selected by the user

5. Time at which the query has been submitted for query

User id	query	Clicked url	time
admin	Data mining	www.dming.com	12:00
admin	Data warehouse	www.dwarehouse.com	12:01
admin	Search engine	www.google.com	12:02

V. QUERY SIMILARITY

The next step in proposed system is computing the query similarity. It is an important crisis and has a wide range of applications in Information Retrieval in query recommendation. Traditional approaches make the use of keywords extracted from documents. If two documents share some keywords, then they are thought to be similar to some extent. The more they share common keywords, and the more these common keywords are important, the higher their similarity is. This similar approach may also apply to query clustering, when a query may also be represented as a set of keywords in the same way as a document. On the other hand, it is well known that clustering using keywords has some drawbacks, due to the fact that keywords and meanings do not strictly correspond[4]. The approach taken by this module is based on two criteria: one is on the queries keywords, the number of times the term occuuring in the document tf, the documents occurred with that term df. The idf shows how rare the term occurred in the document.If two user queries contain the same or similar terms, they denote the same or similar information needs. The following formula is used to measure the content similarity between two queries:

$$\frac{KW(p,q)}{KW(p) \cup KW(q)}$$

Where kw (p) and kw (q) are the sets of keywords in the queries p and q respectively, KW (p, q) is the set of common keywords in two queries [9]

Similarity Based on Document terms and frequency.

In finding the similarity between the query and the document another method is information retrieval method . An IR method governs how a document and

a query are represented and how the relevance of a document to a user query is defined. Main models:

1. Boolean model

2. Vector space model

3. Statistical language model

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (985-992), Month: April - June 2015, Available at: www.researchpublish.com

Explanation of various models:

1. Boolean model:

Each document or query is treated as a **"bag" of words** or **terms**. Word sequence is not considered.

Given a collection of documents *D*, let $V = \{t_1, t_2, ..., t_{|V|}\}$ be the set of distinctive words/terms in the collection. *V* is called the vocabulary. A weight $w_{ij} > 0$ is associated with each term t_i of a document $\mathbf{d}_j \in D$. For a term that does not appear in document $\mathbf{d}_j$, $w_{ij} = 0$.

$$\mathbf{d}_j = (w_{1j}, w_{2j}, ..., w_{|V|j}),$$

Strengths of Boolean model are:

a) Precise, if you know the right strategies

b) Precise, if you have an idea of what you're looking for Implementations are fast and efficient

c) Weaknesses Users must learn Boolean logic

d) Boolean logic insufficient to capture the richness of language .No control over size of result set: either too many hits or none. e) When do you stop reading? All documents in the result set are considered "equally good" What about partial matches? Documents that "don't quite match" the query may be useful also

Boolean Models – Problems

Very rigid: AND means all; OR means any.

Difficult to express complex user requests**.**

Difficult to control the number of documents retrieved. *All* matched documents will be returned.

Difficult to rank output. *All* matched documents logically satisfy the query.

Difficult to perform relevance feedback. If a document is identified by the user as relevant or irrelevant, how should the query be modified?

2. Vector Model:

Non-binary (numeric) term weights used to compute *degree of similarity* between a query and each of the documents. Enables partial m*atches* to deal with incompleteness . Answer set ranking to deal with information overload.

Vector Space Retrieval Model:

Advantages

Simplicity: Easy to implement

Effectiveness: It works very well

Ability to incorporate any kind of term weights

Can measure similarities between almost anything:

- documents and queries, documents and documents, queries and queries, sentences and sentences, etc.

Used in a variety of IR tasks:

- Retrieval, classification, summarization, SDI, visualization, …

The vector space model is the most popular retrieval model (today).

As queries submitted by the user is in the form of term so to convert it into the numbers we use the term frequency.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 2, pp: (985-992), Month: April - June 2015, Available at: www.researchpublish.com

TERM FREQUENCY (TF)

Number of times a term occurred in a sentence is called as the 'Term frequency'. It is represented as **"**tf".

$$= \sum_{t \in q \cap d} (1 + \log \mathrm{tf}_{t,d})$$

DOCUMENT FREQUENCY (DF)

Number of times a term occurred in the whole document is called as 'Document frequency'. It is represented as **"**df**"**.

INVERSEDOCUMENT FREQUENCY (IDF)

It is the logarithm of inverse of the document frequency. Inverse document frequency is represented as **"**idf**"**[10]where n – Number of sentences df – document frequency

$$\mathrm{idf}_t = \log_{10} (N/\mathrm{df}_t)$$

Query-document matching scores

- We need a way of assigning a score to a query/document pair
- Let's start with a one-term query
- If the query term does not occur in the document: score should be 0
- The more frequent the query term in the document, the higher the score (should be)
- We will look at a number of alternatives for this.

Improved similarity formula

fscore(qs, d) = similarity of keywords ×qNorm(qs) × $\sum tf\ (t, d) \times inverse\ df(t)^2$

- tf is the number of times term occurred in the document.
- Idf is the inverse document frequency
- qnorm is the normalization of similar values.
- Similarity of kewords of the documents

VI. CLUSTERING

An important component in this work is the concept of clustering queries in user logs. The query clustering is a preprocessing phase and it can be conducted at periodical and regular intervals. Even though the need for query clustering is somewhat new, there have been general studies on document clustering, which are similar to query clustering. However, it is not reasonable to easily apply any document clustering algorithms to queries due to their own characteristics. It is usually observed that queries submitted to the search engines typically are very short, so the clustering algorithm should be suitable for short texts. Additionally query logs are usually very large, the method should be able of handling a large data set in reasonable time and space constraints. Furthermore, due to the fact that the log data changes daily, the method should incremental.

Clustering Algorithm

Another question involved is the clustering algorithm proper. There are many clustering algorithms available to us. The main characteristics that guide our choice are the following ones:

The algorithm should not require manual setting of the resulting form of the clusters, e.g. the number of clusters. It is unreasonable to determine these parameters manually in advance.Since we only want to find FAQs, the algorithm should filter out those queries with low frequencies. Since query logs usually are very large, the algorithm should be capable of handling a large data set within reasonable time and space constraints.Due to the fact that the log data changes daily, the algorithm should be incremental[4]

Algorithm: Query_Clustering(Q,α,β,τ)Given : A set of n queries and corresponding clicked url'sstored in an array url'sstored in an array

Q[q1,URL1…..URL m] . 1<=i<=n

α=β=0.5

Similarity Threshold τ

Output : Aset C={C1,C2….Ck} of k query clusters

//Start Algorithm

K=1; // k is the number of clusters

For (each query p in Q)

Set ClusterId(p) - Null;

//Initially No Cluster is clustered

For (each p € Q)

{

ClusterId(p) = Ck;

Ck –{ p };

For each q € Q such that p ≠ q

{

Sim(p, q) =

|KW(p, q)|

|kw(p) U kw(q)|

Sim(p,q)=$\frac{KW(p,q)}{KW(p)\cup KW(q)}$

fscore(qs, d) = similarity of keywords ×qNorm(qs) × $\sum tf\ (t,d) \times inverse\ df(t)^2$

α× $similarity\ of\ keywords + \beta \times fscore$

if combine similarity> τ

Set ClusterId(q) = Ck;

Ck = Ck U {k};

Else

Continue;

} // End For

K=K+1;

} //End Outer For

Return Query Cluster Set C

Favored Query Finder

When query clusters are formed, another phase is to find a set of favored queries from each cluster. Query is said to be favored query that occupies the foremost portion of the search requests in a cluster. The process of finding favored queries is which find the favored queries in one cluster. The method is applied in every the clusters and output is stored in the Query Cluster Database[4]

Algorithm: Favored Query Finder()

I/P : A Cluster of Queries

O/P : True or False.

//Start of Algorithm

1. Queries Which are exactly same club them and make a set of the <query,IP addresses> pairs.

2. For(each q € Clusters)

3. Calculate the weights of query as :

Wt = No. Of IP addresses which Fired the

query/Total No. of IP Addresses in that cluster

If (Wt >= threshold Value) then

Return True; //Query is considered as favored query

Else

Return False; //Query is considered as disfavored

VII. CONCLUSION

In this paper, formula of result optimization system has been proposed based on query log for implementing effective web search. The most significant feature is that the result optimization method is based on users' feedback, which determines the relevance between Web pages and user query words.The returned pages are directly matched with the user query. By applying the designed formula in the query log the user get most relevant query related to the input query. Hence, the time user spends for looking for the required information from search result list can be reduced. As the system based on click through data in query log. By applying the information retrieval method in the query log as a result it improves the similarity and gives the highly relevant result As the future work, we can apply a more relevant formulas and algorithms to update the query more efficiently. Although a conclusion may review the main points of the pa-per, do not repeat the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions. As future work, we consider to improve the notion of attention of the suggested queries and to expand other notions of interest for the recommendation algorithm.

REFERENCES

[1] A.K Sharma, Neelam Duhan, Neha Aggarwal, Ranjana Gupta. "**Web Search Result optimization by mining the Search Engine Query Logs,**". Proc. of International Conference on methods and models in Computer Science, Delhi, India, Dec.13-14, 2010.

[2] Neelam Duhan, A.K Sharma."**Rank Optimization and Query Recommendation in Search Engine using Web Log Mining Technique**,". Journal of computing. Vol 2, Issue 12, Dec. 2010.

[3] Murat Ali Bayer, Ismail H. Toroslu, Ahmet Cosar." **A Performance comparison of Pattern discovery methods on web log data,".** Proceedings of AICCSA, pp 445-451. 2006.

[4] Rekha and sushil kumar"**Design of query suggestion using rank updater"Journal of computer trends and technology,**" .Volume 11 number 5-May 2014.

[5] Megha R. Sisode, Ujwala M. Patil ," **A Survey on Query Recommendation Techniques and Evaluation of Snippet based Query Recommendation."** International Journal of Computer Applications (0975 – 8887) National Conference on Emerging Trends in Information Technology (NCETIT-2011)

[6] Dd A. P. Siva kumar1, Dr. P. Premchand2 and Dr. A. Govardhan" **Query-Based Summarizer Based on Similarity of Sentences and Word Frequency,**" International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.1, No.3, May 2011

[7] O.R Zaine and A. Strilets. "**Finding similar queries to satisfy searches based on query traces**,". In proceedings of the international workshops on efficient web based information system, France Sept. 2002

[8] Hamada M.Zahera, Gamal F. El Hady, Waiel.F Abd El-Wahed**" Query Recommendation for Improving Search Engine Results,"** World Congress on Engineering and Computer Science 2010 Vol I

[9] Nikita Taneja, Rachna Chaudhary , "**QueryRecommendation for Optimizing the Search Engine Results,"** International Journal of Computer Applications (0975 – 8887) Volume 50 – No.13, July 2012

[10] Fff A. P. Siva kumar , Dr. P. Premchand and Dr. A. Govardhan" **Query-Based Summarizer Based on Similarity of Sentences and Word Frequency,"** International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.1, No.3, May 2011.

[11] J.XY and W.B. Croft. "**Improving the effectiveness of information retrieval with the local context analysis**." ACM Transaction of information system,79-112,.2000.

[12] Zhi Teng., Ye Liu., Fuji Ren ., Seiji Tsuchiya., and Fuji Ren "**Single Document Summarization Based on Local Topic Identification and Word Frequency**" In Seventh Mexican International Conference on Artificial Intelligence 2008.

[13] E. Peukert, S. Maßmann, and K. K¨onig." **Comparing similarity combination methods for schema matching**. "In GI Jahrestagung (1), 2010.

[14] Magdalini Eirinaki, Suju Abraham, Neoklis Polyzotis, and Naushin Shaikh" **QueRIE: Collaborative Database Exploration"** IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 26, NO. 7, JULY 2014.

[15] Caramia, G. Felici and A. Pezzoli, "**Improving search results with data mining in a thematic search engine**," Computer & Operations Research 31, pp. 2387-2404, (2004) Elsevier.