*I.J. Information Technology and Computer Science,* **2016, 11, 82-91**
Published Online November 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2016.11.10

# Query Recommendation by Coupling Personalization with Clustering for Search Engine

**Dhiliphanrajkumar.Thambidurai**
Department of Computer Science and Engineering, Manonmaniam Sundaranar University,
Tirunelveli, TamilNadu-627012, India
E-mail: dhilipanrajkumar@gmail.com

**Suruliandi. Aandavar and Selvaperumal.Prakasam**
Department of Computer Science and Engineering, Manonmaniam Sundaranar University,
Tirunelveli, TamilNadu-627012, India
E-mail: {suruliandi@yahoo.com,selvaperumal.p@gmail.com}

***Abstract*—**In the present world internet and web search engines have become an important part in one's day-to-day life. For a user query, more than few thousand web pages are retrieved but most of them are irrelevant. A major problem in search engine is that the user queries are usually short and ambiguous, and they are not sufficient to satisfy the precise user needs. Also listing more number of results according to user make them worry about searching the desired results and it takes large amount of time to search from the huge list of results. To overcome all the problems, an effective approach is developed by capturing the users' click through and bookmarking data to provide personalized query recommendation. For retrieving the results, Google API is used. Experimental results show that the proposed method is providing better query recommendation results than the existing query suggestion methods.

***Index Terms*—**Web Search, Personalization, Clustering, Query Recommendation, User Queries.

## I. Introduction

As internet use increases dramatically, so, correspondingly, does the number of pages indexed in search engines. The indexed web contains at least 3.53 billion pages. With such large volumes of data at hand, finding relevant information that satisfies user needs based on simple search queries becomes an increasingly difficult task. Search engines today throw up results based on web popularity rather than user interest. An independent survey of 40,000 web users threw up an interesting finding: after a failed search, 76% of all users merely attempted to rephrase their queries on the same search engine rather than resort to others [3]. A major reason users fail to receive much-needed desired information is that their queries are, more often than not, short and ambiguous. A study by M. Jansen [13] found that the average query size on a search engine was only 2.35 terms. The difficulty with short queries is that the average user is unlikely to be able to accurately express what he actually needs. Ambiguous queries lead to indistinct search results. It is expected that the web, overall, may consist of more than 1 trillion unique URLs. Consequently, lots of pages retrieved are irrelevant to a user's needs because of the ambiguous nature of these queries. Most search engines, therefore, use ranking, clustering and assorted web-mining methodologies to optimize search results, with query recommendation also serving as a method to improve searches. It is an internal component of modern search engines and helps users explore concepts related to their particular needs. Search engines such as Yahoo, Google and Ask suggest both user-specific and user-general queries, yet offer the same suggestions to the same queries without considering individual users' interests. Web page query-clustering identifies meaningful groups of web pages and presents these to the user as clusters. Clustering provides result sets and, when a cluster is selected, the extracted result sets are refined to find the relevant pages. It is necessary to identify user interest by means of their queries and clicks. Hence, in this work, a method is proposed based on query recommendation by coupling personalization-based user clicks and bookmarking with modified query clustering. The results are stored in the form of a query log which provides support in finding and analyzing what users are interested in, and forms a complete record of what they have searched for, given a particular time frame.

## II. Related Work

Search engine results today are based on a reputation on the web rather than user needs. Results that show up on the very first page have greater chances of engaging user interest. Further, users are unsure of how to phrase a query accurately so as to obtain the desired results. There is, consequently, a need for a query recommendation-based search engine coupling personalization with clustering. The goal of query recommendation is to facilitate easier user search for data, allowing users to explore concepts related to their particular information needs. In a personalization search based on user groups, user-profiling strategies such as $P_{Click}$ and $P_{Joshaims}$ are

 

considered and the click-through collected to forecast a user's conceptual needs [8][9] [10]&[11]. Reiner Kraft et al. [14] proposed a method for automatically generating refinements, or related terms to queries, by mining an anchor text for a large hypertext document collection. However, they failed to include additional pre-anchor and post-anchor text. Ricardo Baeza-Yates et al. [15] proposed a method that provides users lists of query suggestions based on previously issued queries, and can be used to fine-tune or redirect the users' search process. Clustering is done from the content in terms of historical preferences and ranking, made with relevance being the criterion. The drawback is that recent search results are not considered and no experiments have been conducted to deal with the issue of ambiguous queries. Kenneth Wai-Ting et al. [9] proposed a personalized content-based clustering of search engine-provided query suggestions for individual users. They extract concepts from web snippets of search results returned from a query. Different users may submit exactly the same query though their needs are different. Clustering categorizes data into groups or clusters such that the data in the same cluster are more similar to each other than those in different clusters. K.W. Ting [10] proposed personalization based on user profiles, employed to group similar queries according to user needs. A user profile is carefully scrutinized, user-searched contents traced, user behavior indicating the time spent on reading online documents and areas of user interest determined. The drawback with the user group method lies in that the user does not always provide the right information and, further, the user's interest may change over time. Keneeth et al. [11] captured users' click-throughs and locations to provide users results, the drawback being that they failed to provide results that matched users' particular needs. Gloria et al. [6] tracked the query behavior of each user, identified those parts of the database that might be of interest for the corresponding data analysis task, and recommended queries that retrieved the relevant data. The problem lies in that the proposed method does not identify similar queries in terms of their structure. Eldar Sadikov et al. [5], in clustering query refinements by user intent, proposed a clustering algorithm for query refinement to improve the selection and placement of query suggestions proposed for the search engine. This algorithm combines information from document clicks and user sessions, the drawback being that it fails to provide appropriate results for ambiguous queries. Avinash et al. [1] proposed an approach that aims to mine a reduced set of effective searches to enhance the search experience, store and maintain users' long-term dynamic profiles based on user search, and subsequently use it to personalize queries. The drawback with this method is that storing users' long-term search data mandates that the memory used for storage must, of necessity, be huge. The base of page ranking was proposed by Brin and Page [3], where each web keyword search is used for page ranking and the process extended by counting links from all pages. The difficulty here is that they failed to develop or model a t for tracing user behavior. Barathi et al. [2], in their study on a topic-based query suggestion using a hidden topic model for effective web search, proposed a novel query suggestion method, providing suggestions related to topics present in the input query and re-ranking the documents retrieved. They do not consider user history while offering query suggestions; instead, they offer suggestions based on the terms in the input query. Consequently this method falls short of providing users the query suggestions needed. Certain researchers have noticed that personalization varies, for different queries, in terms of effectiveness. In our earlier comparative work on the personalization of search engines, we found that personalization based on user groups provides good results [17]. In this work, a query recommendation coupling personalization based on user clicks and bookmarking, and a modified clustering algorithm, is proposed.

### *A. Motivation and Justification*

Search engines often have difficulty in arriving at a short and exact representation of user needs, returning huge numbers of web pages back to user queries, with users ultimately having to waste a lot of time finding the content needed. Further, they fall short of framing the input query in accurate terms, and search engines rely on a measure of accuracy to find the desired search results. In comparison, cluster descriptions in search engines are far more informative, with users also getting an excellent overview of the subjects presented in search results. The aim of clustering is to speed up search time. However, the computing time taken for clustering is high. Besides, the drawback of clustering is that it only takes note of a topical similarity between documents in a ranked list. The problem with ranking is its failure to display a results list based on user requests or preferences. In a traditional approach, a search engine always returns the same rank for the same query submitted at different times or by different users. As massive volumes of data become available, the principal problem becomes the need to scan page after page to find the desired information a user needs. In current search engines, the difficulty is with ordering search results and then presenting the most relevant page first to the user. Motivated by the facts mentioned above, a query recommendation-based search engine coupling personalization with clustering is proposed, in this paper, for improving query results. It is expected that users get relevant results for a given query on personalization, Also, a modified clustering algorithm provides results in a cluster, aiding users retrieve results faster. Further, query-suggested results in response to users' queries are far more useful for finding the results needed. Justified by this fact, a query recommendation technique is combined with personalization and a clustering algorithm, the key purpose being to give users combing the web using a specific query a comprehensive set of recommendations. Recommending the most relevant search keyword sets to users both enhances the search engine hit rate and also helps users find the desired results a lot more quickly.

 

### B. Outline of the Proposed Work

The block diagram in Fig.1 explains the flow of the proposed model. This model addresses query recommendation, clustering and personalized results, the backbone of users looking to search engines for query processing. A user query is given to a user interface, surfed using search engines, and results fetched from the web. The resulting snippets are preprocessed using the Porter stemming algorithm that effectively eliminates irrelevant data such as html tags, stop words and so on. Personalization based on click-through and bookmarking are applied over the preprocessed data and data of interest to the user identified. These results are rearranged and processed using a page-ranking algorithm. The modified agglomerative clustering system is employed to gather similar data into a group so as to fine-tune query results. Results users are interested in, as well as the number of times the contents were clicked, are stored in search engine clicks through logs. The logs help analyze user interest, and query suggestions provided in the query recommendation phase are displayed to users. Query recommendation results provided by the system are evaluated using performance metrics like the precision recall f-measure.

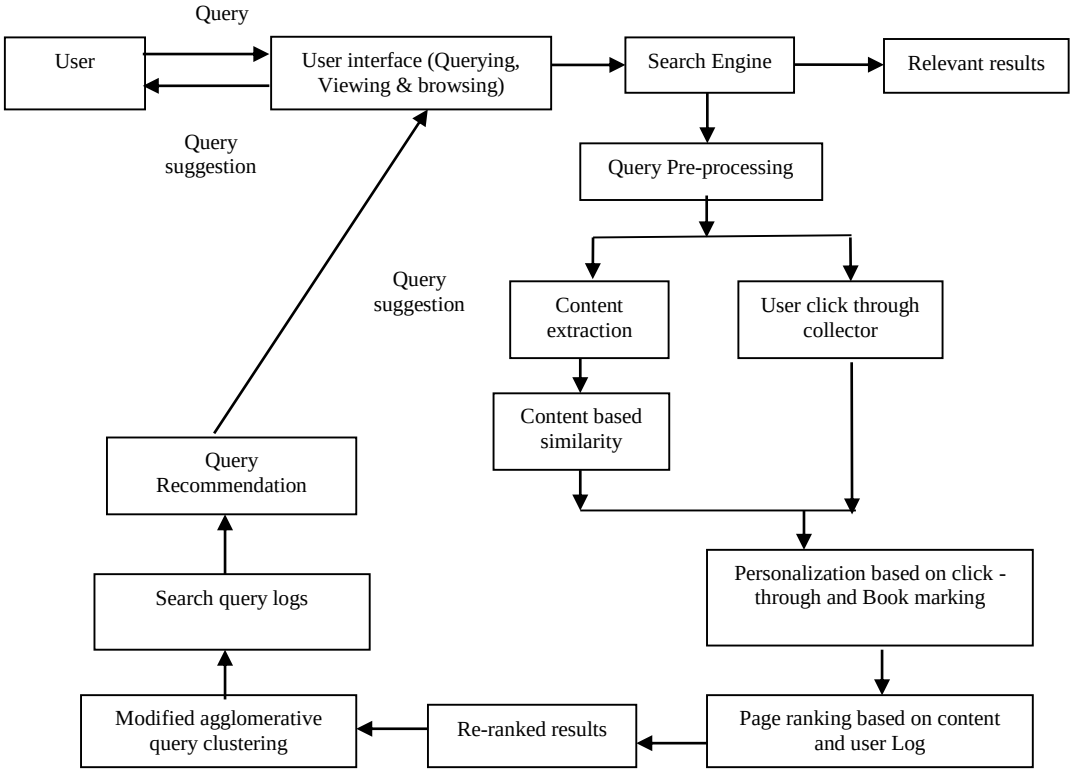


Fig.1. Block Diagram for the Proposed System

### C. Organization Of The Paper

The remaining sections are organized as follows. In Section 2, the methodology of the proposed technique is discussed. In Section 3, detailed statistics of the datasets used in the experiment, the results and a discussion are provided, along with a comparative analysis of the proposed and existing methods. Section 4 focuses on the conclusion and future developments.

## III. Methodology

The process flow of the proposed work is discussed in the following sections.

### A. User Query

A user first queries a search engine to meet his/her particular needs. **A** query interface, which is a graphical user-interface design for providing a system with inputs, comes into play. Inputs can be search terms required to look for data extraction from data sources in the web.

### B. Query Preprocessing

Data in the real world is often dirty and incomplete, lacking both attribute values as well as certain attributes of interest, or containing only noisy and inconsistent aggregate data. There is no quality in data-extracted results and, likewise, no quality mining. Further, queries asked by users are often particularly short. After retrieving results related to user queries, the snippets are mixed with a lot of irrelevant content, including stemming and stop words.

They are preprocessed using information retrieval techniques, the aim being to generate very relevant results for a search query. Fig.2 shows the preprocessing of

 

snippets including the removal of stemming and stop words.

*1. Stemming*

Stemming is a term used in the information-retrieval process to reduce inflected words to their word stem or base. Stemming algorithms are used to transform words in a text and improve the information-retrieval system, the goal being to obtain a one-word description of similar - but not identical - words. The word finally obtained has neither meaning nor is grammatically correct, but contains a description of and bears a similarity to all the other words it represents. For example, "implementing" and "implemented" are described by the word "implement."

*2. Stop Word Elimination*

After stemming, it is necessary to remove expendable words. Stop words are filtered before or after processing data. A stop word is a word that does not have a meaning, so eliminating stop words offers better results in a phrase search. In all languages, certain words are considered stop words, of which there are more than 500 types. For example, words such as "on," "and," "the" and "in," among others, do not provide useful information. After pre-processing such snippets, the results are considered for further processing.

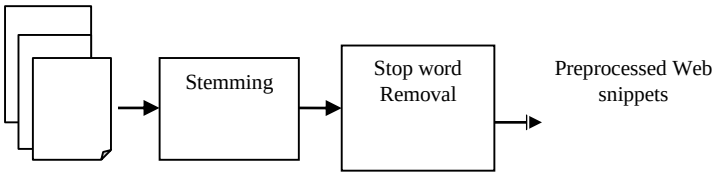


Fig.2. Processing of web snippets

*C. Web Page Content Retrieval*

Following the removal of irrelevant data, contents that frequently occur in snippets are extracted and the relationship between the extracted words analyzed in this phase. Further, user-clicked contents are collected in a user click through a collector.

*1. Extracting Content from Web Snippets*

Content from the refined results is extracted by finding frequent item sets in data mining. When a user types a query, a set of relevant web snippets are returned and if a keyword or phrase crops up frequently in web snippets relating to a particular query, it represents important content related to the query because it exists alongside the query in documents that come up right on top. To measure interest in a particular keyword or phrase $k_i$ extracted from web snippets:

$$\sup port = \frac{sf(k_i)}{n} * |k_i| \tag{1}$$

where $sf(k_i)$ is the snippet frequency of the keyword or phrase $k_i$ , $n$ the number of web snippets returned, and $|ki|$ the number of terms in the keyword or phrase $ki$. If the support of a keyword or phrase $k_i$ is greater than the threshold $s$, then $k_i$ acts as a concept for the query $q$. The maximum length of a concept is limited. This process minimizes processing time and eliminates the extraction of meaningless content.

*2. Content-Based Similarity*

In extracting content-based similarity, a signal-to-noise formula is used to establish the similarity between keywords $k_1$ and $k_2$. The two keywords from a query $q$ are similar if they coexist frequently in web snippets arising from the query $q$.

$$sim(k_1, K_2) = \frac{\frac{n * fd(k_1 \cap k_2)}{df(k_1) * df(k_2)}}{\log n} \tag{2}$$

where $n$ is the number of documents in a mass, $df(k)$ the document frequency of the keyword $k$, and $df(k_1 U k_2)$ the joint document frequency of $k_1$ and $k_2$. The similarity $sim(k_1,k_2)$ obtained using the above formula always lies between [0, 1]. In search engine contexts, two concepts $k_i$ and $k_j$ can coexist in web snippets

$$sim_{R,\, snippet(k_i,\, kj)} = \frac{\log \frac{n * sf_{snippet}(k_i \cup k_j)}{sf_{snippet}(k_i) * sf_{snippet}(k_j)}}{\log n} \tag{3}$$

where $sf_{snippet}(k_i \cup k_j)$ / $sf_{snippet}(k_i \cup k_j)$ are joint snippet frequencies of the concept $k_i$ and $k_j$ in web snippets. $sf_{snippet}(k_i).sf_{snippet}(k_j)$ is the snippet frequency of the concepts $k_i$ and $k_j$ respectively for finding essential features from data word frequency using the following formula

$$Word\ \ Frequency = \frac{No.\ of\ Words\ in\ a\ Document}{Total\ Words\ in\ Document} \tag{4}$$

Now, using the Euclidean distance, the similarity between extracted data and available data is computed and, according to the distance obtained, the data is placed in a similar set where it actually belongs.

*3. User Click-through Collectors*

The relationship that exists between concepts is processed by considering a user's click-through. User-clicked queries are termed user-positive preferences and

 

the rest user-negative ones. When a user clicks on a query, the weight of the extracted concept is incremented by 1 to show user interest. Other concepts related to the user's query are also incremented to a similar score. If the concept is closely related to the user's positive preferences, then it is incremented to a higher value. It is otherwise incremented to a small fraction close to zero, by means of which a user log is created. After finding the data needed using the search engine, it is ranked according to its relevance to the user's query, requiring that a new kind of system be implemented as a consequence. Thus, the proposed work is designed in the manner set forth below. The given ranking system is implemented in components of different text-processing and weight-estimation techniques.

*D. Personalization Based on User Click and Bookmarking*

Personalization (also termed customization) of search engines is meant to present users appropriate and desired results. Web pages are personalized, based on individual interests, social categories and contexts. Personalization implies that changes are based on implicit data, such as items purchased or pages viewed. In most personalized search techniques, the data searched by a user is taken into consideration when creating a user profile. Currently, certain new strategies followed include the liking of a group of users to carry out personalized searches [8,10 &11]. In this paper, user concept-based queries and bookmarking are considered for personalization, and $P_{Click}$, $P_{Joachims}$ and $P_{Click+Joachims}$ used to trace the click histories of a group of users with similar interests [8]. Concept-based users' positive preferences are considered in $P_{Click}$, whereas $P_{Joachims}$ is based on users' document preferences. The concept for a query q using the concept-extraction method provides feasible concepts that may cover more than the user's actual needs. Therefore, the following formulae are used to capture an individual user's degree of interest $w_{ci}$ on the extracted concepts $e_i$, when a web-snippet $W_j$ is clicked by the user (denoted by click ($c_j$)):

$$click(w_j) \rightarrow \forall e_i \in c_j, w_{ci} = w_{ci} + 1 \quad (5)$$

$$click(w_j \rightarrow \forall e_i \in c_j, w_{ci} = w_{ci} + sim_R(e_i, e_j) \quad \text{if } sim_R(e_i,e_j)>0 \quad (6)$$

where $W_j$ is a web-snippet, $w_{ci}$ represents the user's degree of interest on the extracted concept $e_i$, and $e_j$ the neighborhood concept of $e_i$.

When a web-snippet $W_j$ has been clicked by a user, the weight $w_{ci}$ of concepts $e_i$ appearing in $W_j$ is incremented by 1. Joachims et al. [8] introduced a technique entirely based on click-through data to learn the ranking function, and presented an empirical evaluation of understanding clicks through evidence. It is believed that a user would search for results from top to bottom. If a user skipped a document $d_i$ before clicking $d_j$ (where the rank of $d_j$ > rank of $d_i$), one must have searched $d_i$ first and determined not to click on it. A document-based method is converted to a concept-based one. For all the concepts $c_1$, $c_2$.....$c_i$ extracted for a query q, user-selected contents are stored in the corresponding weight values $W_{c1}$, $W_{c2}$.....$W_{cn}$, creating a concept profile for the query q.

$$P_{joachims} = (W_{c1}, W_{c2}.....W_{cn}) \quad (7)$$

It is observed that the $P_{Click}$ method is used to capture a user's positive results. The Joachims et al. method is used to capture negative preferences. Good precision and recall value can be achieved by combining both results [6]. User profiles $P_{Click}$ and $P_{Joachims}$ can be combined with the bookmarked clicked contents using the formula:

$$W(P_c + P_j + P_b) = W(P_c) + W(P_j) + W(P_b), if W(P_j \prec 0) \quad (8)$$

$$W(P_c + P_j) = W(P_c), otherwise \quad (9)$$

where $W(P_C+P_J+P_b) \in P_{Click+Joachims+Bookmark}$, $W(P_C) \in P_{Click}$, , $W(P_J)$ and $W(P_b) \in P_{Joachims}$. The combined user profile method is applied over Google search results and re-ranked, based on user-interested results, with a group level re-ranking being used. The following formula is used for calculating the similarity for a group of users.

$$sim(u_1, u_2) = \frac{c_1(u_1).c_1(u_2)}{\|c_1(u_1)\|\|c_1u_2\|} \quad (10)$$

where u1 and u2 are user1 and user 2 and c1 the category vector of the web page.

*E. Ranking*

User-interested results for a query are stored in a database and, over time, collected using user clicks through a collector. Termed a query log, it provides useful information on searchers' queries and what users are interested in. A problem peculiar to a query log is that it has no relational information other than a query and a click. Considerable portions of queries are rare, with few clicks or even no clicks at all for certain queries [20]. In the proposed work, a combined user profile method is applied to Google search results and the retrieved results re-ranked, based on user-interested results with a level re-ranking being used for this particular group. Both web structure and web content mining are used to give users the relevant results anticipated. Web content mining is used here to get the linking structure of a web page and trace the content and similarity between items on the contents of the said web page. Initially, a user visits a web page at random but this change over time. Here, user interest is calculated using clicks on web links. The quality of user interest changes dynamically with the number of user visits to a particular web page. The probability $P_i$ of a user-taken decision equals user interest relative to the sum of all user-interested values. To find the probability of a user choosing a web page 'i' is

 

$$P_i = \frac{u_i}{\sum_{i=s} u_i} \quad (11)$$

Here $u_i$ is user interest and S the similarity of contents in the web page.

As user interest changes, user interest value $u_i$ is updated accordingly, along with the time spent by the user when the page was visited for the query. As a result, the relevance of the page to the user increases greatly and the probability of its being chosen also increases correspondingly. When the user visits the page, the quantum of user interest is updated. The volume of the web page increases in proportion to its quality.

$$U_i^{t+i} = U_i^t + \Delta U_i^t \quad (12)$$

where $u_i$ is user-interested value at time in second and $\Delta U_i^t$ the amount of user interest saved at time t left by the user. It can be changed, depending on user interest in terms of clicks.

### *F. Modified Agglomerative Clustering Algorithm*

Clustering is the process of organizing objects into groups whose members are similar in some way. A cluster is therefore a collection of objects which are similar, and dissimilar to objects belonging to other clusters. Organizing search results into groups prunes search time. In a modified clustering algorithm, entire paths from a query to the documents should be considered first rather than offering the resultant documents by merely looking at the queries. The second factor to be considered is that a path from a topic to another has to be related to the given query. It is most essential that the concept behind the terms of every document to be clustered is understood. The proposed clustering method uses ontology to understand the concept of a given query. Ontology is a formal representation of a set of concepts inside a domain and the connections between concepts. At its inception, ontology for a domain involved human effort. Today it incorporates domain knowledge into data mining methodology [16, 22].

**Algorithm** Modified agglomerative clustering

Input: Re-ranked prioritized data
Output: Personalized query-clustered results
Steps

1: Find the similarity scores of possible pairs of queries.
2: Merge a pair of the most similar queries ($q_i$,$q_j$).
3: Do not merge the same query from different users.
4: Concept c is coupled with query $q_i$ and $q_j$, and a new link created between c and ($q_i$,$q_j$) with weight
$w = w_i + w_j$
5: Obtain the similarity scores for all possible pairs of concepts using step (4).
6: Use ontology to merge the pair of concepts ($c_i$,$c_j$).
7: Unless termination is reached, repeat steps 1-4.

### *G. Search Query Logs*

Search query logs are popular data sources for query recommendations. A typical log in most search engines, includes fields like the ID, user query, URLs clicked by the user, the rank of the URL clicked for the query, and the time at which the query is submitted for the search. A sample query log used for our experiment is shown below for better understanding.

Table 1. Example illustration of Query log

| ID | Date | URL | Query | Rank |
|---|---|---|---|---|
| 1 | 2015-12-10 10:40:15 | http://www.ccsu.edu/datamining | Data mining | 4 |
| 2 | 2015-12-10 10:50:32 | http://www.apple.com/ | Apple | 6 |
| 3 | 2015-12-10 10:55:42 | http://www.ocpafl.org/ | Orange | 5 |

### *H. Query Recommendation*

Query recommendation is a promising direction for improving web search engine usability. The proposed approach provides recommendations based on user-clicked contents and bookmarking. Recommendations are provided, based on an analysis of the query logs for a user's query and a cluster formed using the logs. A query submitted by a user needs to be matched to its closest cluster to get a query recommendation. Hence it is necessary to compute a query recommendation based on the similarity between input queries, user-clicked content and cluster labels that characterize clusters. All queries in the cluster of highest scores can be served as recommended queries, thereby providing better recommendations than the approach that considers only keyword similarities for doing so.

## IV. Experimental Results and Analysis

In this the dataset used for the experiment, and the different experimental evaluation made all are discussed and the results are analyzed

### *A. Dataset*

Table 2. Statistics of the Tested Queries

| Statistics | |
|---|---|
| Number of users | 30 |
| Number of queries assigned to each user | 5 |
| Number of test queries | 100 |
| Maximum number of retrieved URL for a query | 50 |
| Number of extracted concept for a query | 15656 |

Google API is used for preparing a dataset for user queries. Google search results for 30 days in November 2015 are taken for collecting user-searched data and to validate the work. Default snippet counts are set to 100 and, in the log, user-clicked contents, as well as their

positive and negative preferences, are collected. Table 2 displays statistics for the tested queries.

Certain queries used for the evaluation are ambiguous, and include names of entities and general terms. Table 3 shows queries used for the evaluation of the personalization of search results.

Table 3. Queries Used For Evaluation

| **Types** | **Queries** |
|---|---|
| Ambiguous | apple, tiger, sun, penguin, java |
| Entity names | dell, Disney |
| General terms | maps, flower, music, network |

*B. Performance Metrics*

The proposed method is evaluated using performance measures like precision, recall and f-measure, computed using

*1. Precision*

Precision is the fraction of documents retrieved that are relevant to a user's information needs. It takes all retrieved documents into account.

$$\Pr ecision = \frac{\text{Re}trieved\ Relevant\ Document}{\text{Re}trieved\ Document} \qquad (13)$$

*2. Recall*

Recall is the fraction of documents successfully retrieved and relevant to a query. Also termed sensitivity, it can be looked at as the probability that a relevant document is retrieved by the query.

$$\text{Re}call = \frac{\text{Re}trieved\ \ \text{Re}levant\ Document}{All\ \ \text{Re}levant\ Documents} \qquad (14)$$

It is easy to achieve a recall of 100% by retrieving all documents in response to a query. Hence recall alone is not enough, and the number of non-relevant documents is required to be measured as well.

*3. F-measure*

F-measure is the harmonic mean of precision and recall, and provides good results when precision and recall provide good results.

$$F - measure = 2.\frac{\Pr ecision * \text{Re}call}{\Pr ecision + \text{Re}call} \qquad (15)$$

*C. Experimental Results and Performance Evaluation*

Experiments are conducted using different queries to check the performance of the retrieved results for techniques used in our work such as personalization, page ranking, clustering, and query recommendation, based on metrics like precision, recall, and f-measure. We now turn to the experimental evaluation of finding the relevance of search results, and the performance of the proposed method is calculated using 100 queries.

Table 4. Result for the proposed method with Precision, Recall and f-measure

| Query | Precision | Recall | F-measure |
|---|---|---|---|
| Apple | 0.958 | 0.993 | 0.975 |
| Data mining | 0.870 | 0.977 | 0.920 |
| PHP | 0.893 | 0.983 | 0.935 |
| Web mining | 0.904 | 0.968 | 0.934 |
| Jaguar | 0.919 | 0.984 | 0.950 |
| Google | 0.817 | 0.974 | 0.888 |
| Network | 0.835 | 0.978 | 0.900 |
| Tiger | 0.787 | 0.950 | 0.860 |

From an analysis of the different experiments carried out, we have come to the conclusion that the rates of both precision and recall are high, as seen in Table 4. As precision and recall are higher for the proposed method, the F-score value (being the harmonic mean of precision and recall) is also correspondingly higher.

After personalization, the results are re-ranked based on user interest and their accuracy measured, as shown in Table 5. The query given for personalization is used for ranking as well and the metrics used here are precision, recall and f-measure.

Table 5. Precision, Recall and F-Measure Values for Proposed Page Ranking Algorithms

| Query | Precision for the Proposed Page Ranking | Recall for the Proposed Page Ranking | F-Measure for the Recall Proposed Page Ranking |
|---|---|---|---|
| Apple | 0.991 | 0.989 | 0.99 |
| Data mining | 0.926 | 0.96 | 0.943 |
| PHP | 0.919 | 0.979 | 0.947 |
| Web mining | 0.993 | 0.98 | 0.986 |
| Jaguar | 0.893 | 0.99 | 0.938 |
| Google | 0.885 | 1.0 | 0.938 |
| Network | 0.945 | 1.0 | 0.991 |
| Tiger | 0.932 | 0.952 | 0.941 |

According to the results obtained in Table 5, the value for the query 'apple' is high because most users search for the iconic corporation of the same name, thanks to its popularity on the web. The performance of the proposed technique is optimal, and it is easy to achieve a recall of 100% by retrieving all the documents available in response to a query. Hence recall alone is not enough, and the number of non-relevant documents is required to be measured as well.

The efficiency of the modified agglomerative clustering algorithm using performance metrics is arrived at and the results shown in Table 6, with more than 100 queries being used. The prioritized data (i.e., relevant data) is taken up for clustering, the relevance of the results evaluated and the average values given in Table 6.

 

Table 6. Average Precision, Recall and F-Measure values for Query Clustering

| Clustering Algorithm | Avg. Precision | Avg. Recall | Avg. F-measure |
|---|---|---|---|
| Modified Clustering Algorithm | .98 | 1 | .99 |

We conducted a survey of 30 users, and the number of queries used for the purpose of the search is 5. A comparison is drawn, taking into consideration the Yahoo suggestion list, the Google suggestion list, and our own proposed work, as shown in Table 7. As in the proposed work, suggestions are provided by considering user-clicked and bookmarking contents, given the greater levels of accuracy therein.

The experimental results are measured and compared with a keyword-based query suggestion, a topic-based query suggestion, and the proposed method using precision, recall and f-measure, shown in Table 8.

Table 7. Comparison of Query Recommendation results for Yahoo, Google and the proposed work suggestion list

| Query | Yahoo search | Google search based on web popularity | Proposed work |
|---|---|---|---|
| Apple | Apple, Apple iphone, Apple store, Apple.com, Apple ipad, Apple ipod | Apple, Apple iphone, Applestore, Apple.com, Apple ipad Apple ipod, ITunes | Apple Apple fruit |
| Data mining | Data mining, Data warehousing data mining, tool data mining, techniques data mining software, Armada data mining. | Data mining PPT, Data mining Slides, Data mining Tutorial, Data mining Techniques, Data mining Applications, Data mining Classification, Data Warehousing and Data mining. | Data mining |
| Web mining | Web mining, android based web mining | Web mining PPT Web mining Tools, Web mining Projects, PDF for web mining, web content mining Web Data mining, Web structure mining, Orissa mining | Web mining, Web mining Tools, |
| PHP | php tutorial, facebook login php, php interview questions, php full form, php wiki, mysql, html | php full form, php tutorial php download, php interview questions, learn php, php date php array, php explo | php |

Table 8. Average Precision, Recall and F-Measure values for Query recommendation methods

| Metrics | Keyword based query suggestion | Topic based query suggestion | Proposed keyword, user click and bookmarking based query suggestion |
|---|---|---|---|
| Average Precision | .85 | .88 | .90 |
| Average Recall | .88 | .92 | .97 |
| Average F-Measure | .86 | .918 | .93 |

In a keyword-based query suggestion, all suggestions are provided based on the query's web popularity. The Yahoo search engine uses keyword-based query suggestions. A topic-based method also generates better topic-related query suggestions, yet fails to provide suggestions specific to user needs. The proposed work generates better query suggestions than the other because all recommendations provided are based on user clicks and bookmarking. Also, it considers cluster labels when providing query suggestions. Hence the proposed method helps in minimizing the time spent on web searches. Table 8 makes it plain, that the proposed method compared to the others, provides better precision, recall and f-measure.

## V. Conclusion

As a general rule, user queries are short and ambiguous. Lots of effective methods that provide query suggestions are out there to help users get the desired results. In this paper, a new personalized concept-based modified clustering technique is proposed that obtains personalized query suggestions for individual users. In the proposed method, a concept-based similarity is used for computing similarity between queries. Empirical results from the exhaustive experiments show that the proposed approach successfully generates the needed results, particularly suited to the individual's query. Further, it improves accuracy and reduces computational costs and search time, compared to other methods. The difficulty with personalization, however, is that users are generally reluctant to spend extra time on the specifics of their needs. With privacy concerns looming large, however, users are quite unlikely to be comfortable supplying personal information to search services.

 

## REFERENCES


[1] Avinash A.P, Prashant M. Narayankar, Muniraj.M.Jeevitha, "Clustering Method: Search Result Based Web Personalization", International Conference on Advances in Computer and Electrical Engineering (ICACEE'2012) Nov. 17-18, 2012 Manila (Philippines)

[2] Barathi.M, Valli.S, "Topic Based Query Suggestion Using Hidden Topic Model For Effective Web Search", Journal of Theoretical and Applied Information Technology 31st January 2014. Vol. 59 No.3ISSN: 1992-8645

[3] Brin, Sergey, and Lawrence Page. "Reprint of: The anatomy of a large-scale hypertextual web search engine." Computer networks 56, no. 18 (2012): 3825-3833.

[4] David Hawking, "Web search engines. Part 1." Computer 39.6 (2006): 86-88.

[5] Eldar Sadikov, et al. "Clustering query refinements by user intent."Proceedings of the 19th international conference on World Wide Web. ACM, 2010.

[6] Gloria Chatzopoulou, , Magdalini Eirinaki, and Neoklis Polyzotis. "Query recommendations for interactive database exploration." Scientific and Statistical Database Management. Springer Berlin Heidelberg, 2009.

[7] Hina Agrawal, and Sunita Yadav. "Search Engine Results Improvement--A Review." Computational Intelligence & Communication Technology (CICT), 2015 IEEE International Conference on. IEEE, 2015.

[8] Joachims.T, "Optimizing Search Engines Using Click through Data," Proc. ACM SIGKDD, 2002.

[9] Kenneth Wai-Ting Leung, Wilfred Ng, and Dik Lun Lee. "Personalized concept-based clustering of search engine queries." Knowledge and Data Engineering, IEEE Transactions on 20.11 (2008): 1505-1518.

[10] Kenneth Wai-Ting Leung. Ng, and D.L. Lee, "Personalized Concept-Based Clustering of Search Engine Queries," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 11, pp. 1505-1518, Nov. 2008.

[11] Kenneth Wai-Ting Leung, Dik Lun Lee, and Wang-Chien Lee. "APMSE: A personalized mobile search engine." Knowledge and Data Engineering, IEEE Transactions on 25.4 (2013): 820-834.

[12] Lamberti, Fabrizio, Andrea Sanna, and Claudio Demartini. "A relation-based page rank algorithm for semantic web search engines." Knowledge and Data Engineering, IEEE Transactions on 21, no. 1 (2009): 123-136.

[13] Porter, M. F. 1997, "An algorithm for suffix stripping. In Readings in Information Retrieval", K. S. Jones and P. Willett, Eds. Morgan Kaufmann, 313–316.

[14] Reiner Kraft, and Jason Zien. "Mining anchor text for query refinement."Proceedings of the 13th international conference on World Wide Web. ACM, 2004.

[15] Ricardo Baeza-Yates, Carlos Hurtado, and Marcelo Mendoza. "Query recommendation using query logs in search engines." Current Trends in Database Technology-EDBT 2004 Workshops. Springer Berlin Heidelberg, 2005.

[16] Selvaperumal.P et al, "String Variant Alias Extraction Method using Ensemble Learner", International Journal of Intelligent Systems and Application (IJISA)(2015).

[17] Shen Jiang, et al. "Query suggestion by query search:a new approach to user support in web search", IEEE Transaction (2009): 15-18

[18] Suruliandi.A, et al." Validating The Performance Of Personalization Techniques In Search Engine", Ictact Journal On Soft Computing, April 2015, Volume: 05, Issue: 03.

[19] Vishwas Raval, and Pranaw Kumar. "SEReleC (Search Engine Result Refinement and Classification)-a Meta search engine based on combinatorial search and search keyword based link classification." Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on. IEEE, 2012.

[20] Wenhu Tang, et al. "Improved document ranking in ontology-based document search engine using evidential reasoning." IET software 8.1 (2014): 33-41.

[21] Xiaohui Tao, Yuefeng Li, and Ning Zhong, Senior Member, IEEE," A Personalized Ontology Model for Web Information Gathering", IEEE Transactions On Knowledge And Data Engineering, Vol. 23, NO. 4, April 2011

[22] Xinmei Tian, Qianghuai Jia, and Tao Mei. "Query Difficulty Estimation for Image Search With Query Reconstruction Error." Multimedia, IEEE Transactions on 17.1 (2015): 79-91.

[23] Zhang, Zhiyong, and Olfa Nasraoui. "Mining search engine query logs for query recommendation." Proceedings of the 15th international conference on World Wide Web. ACM, 2006.

[24] Zhicheng Dou, Ruihua Song, Ji-Rong Wen and Xiaojie Yuan "Evaluating the Effectiveness of Personalized Web Search" IEEE Trans. Knowledge and Data Eng., vol. 21, no. 8,Aug. 2009


## Authors' Profiles

**Dhiliphan Rajkumar Thambidurai** is a research scholar in Department of computer Science and Engineering Manonmaniam Sundaranar University Tirunelveli, TamilNadu, India from 2012. He received his B.E (2009) in Computer Science and Engineering from Arulmigu Kalasalingam College of Engineering, KrishnanKovil, Srivilliputhur, TamilNadu, India. He received M.E(2011) in Computer Science and Engineering from Muthayammal Engineering College Rasipuram. He has a strong passion in Web Mining, Pattern recognition and Social networking.

**Dr. A. Suruliandi** received his B.E. (1987) in Electronics and Communication Engineering from Coimbatore Institute of Technology, Coimbatore, Bharathiyar University, Tamilnadu, India. He received M.E. (2000) in Computer Science and Engineering from Government College of Engineering Tirunelveli. He also received Ph.D. in Computer Science (2009) from Manonmaniam Sundaranar University as well. He is having more than 27 years of teaching experience. He is having more than 80 publications in International journals and conferences. His research interests include Pattern recognition, Image processing, Remote sensing and Texture analysis.

**Selvaperumal** is currently pursuing Ph.D degree in Department of Computer Science &Engineering, Manonmaniam Sundaranar University, India. He has a strong passion in Web Mining, Data Mining, Machine learning, NLP and Artificial Intelligence.

**How to cite this paper:** Dhiliphanrajkumar.Thambidurai, Suruliandi. Aandavar, Selvaperumal.Prakasam,"Query Recommendation by Coupling Personalization with Clustering for Search Engine", International Journal of Information Technology and Computer Science(IJITCS), Vol.8, No.11, pp.82-90, 2016. DOI: 10.5815/ijitcs.2016.11.10