IADIS International Journal on Computer Science and Information Systems
Vol. 11, No. 1, pp. 81-98
ISSN: 1646-3692

USER FEEDBACK SESSION WITH CLICKED AND UNCLICKED DOCUMENTS FOR RELATED SEARCH RECOMMENDATION

Sejal Desai[1], Vinuth Chandrasheker[1], Vijay Mathapati[1], Venugopal K Rajuk[1], Sundaraja S Iyengar[2] and Lalit M Patnaik[3]
[1]Department of Computer Science and Engineering
University Visvesvaraya College of Engineering, Bangalore University, Bangalore-560001
[2]Florida International University, USA
[3]Indian Institute of Science, Bangalore, India

ABSTRACT

Keyword based search is extensively used method to discover knowledge on the web. Generally, web users unable to arrange and define input queries relevant to their search because of adequate knowledge about domain. Hence, the input queries are normally short and ambiguous. Query recommendation is a method to recommend web queries that are related to the user initial query which helps them to locate their required information more precisely. It also helps the search engine to return appropriate answers and meet their needs. Usually users have ambiguous keywords in their mind to represent their information need. Hence, it is not a good idea to generate relation between user query keywords for recommendations. In this paper, we have presented Related Search Recommendation (RSR) framework, which discovers keywords which are present in snippets clicked and unclicked documents in feedback session. Pseudo documents are generated from feedback sessions which reflect what users wish to retrieve. Finally, semantic similarity is calculated between the terms present in pseudo document and used for recommendations. The proposed method provides semantically related search queries for the given input query. Simulation results show that the proposed framework RSR outperforms Rocchio's model and Snippet Click Model.

KEYWORDS

Pseudo Document, Recommendation, Semantic Similarity, User Feedback Session

1. INTRODUCTION

Web data keeps expanding and is available in various data forms because of rapid growth of online advertising, publishing, e-commerce and entertainment. Although Web search technology provides efficient and effective information access to users, it is still a difficult task to search useful knowledge about user needs from their search queries. Therefore, query suggestion is an important and an essential feature of commercial web search engines. The users can directly use query suggestions results for future new search.

Query suggestion is an efficient way to enhance keyword based search which is extensively useful to web search systems. Users need to modify queries so often because queries are often informational. Users may seek discrete information on a distinct subject, hence may check out various query terms. Users may not have sufficient knowledge on a topic; therefore adequate terms are not known to retrieve the required information.

In Kato et al., [1], Query recommendations are frequently used when (1) a initial query is an exceptional query (2) single term query is used as input query (3) explicit queries are suggested (4) suggestions are provided based on modification of input query (5) various URLs has been clicked by users on the resulting search page.

Query suggestions provided to the user efficiently can reduce the complexity of the search and help them to locate the required information more precisely. This method is extensively accepted by product, music, video search, retrieval of medical information and patent search information. Query suggestions techniques are implemented by commercial search, such as *Searches related* in Google, *Search Assist* in Yahoo!, and *Related Searches* in Bing Search.

Motivation: Through query suggestion, search engines have succeed in obtaining web information for users, but the keyword based search is not able to organise and formulate input queries. Silverstein et al. [2] derived that users' input query's average length is 2.35 terms (AltaVista search engines query log). This shows that most of the user queries are short. A short query cannot describe information need of user search and sometimes ambiguous in meaning expression. Because of insufficient knowledge about domain, users find it difficult to organise and define appropriate input queries. Then user has to rephrase the query words or query frequently, which affects the search performance.

In [3-8] authors have focused on query suggestions by considering users' previous query and click behaviour. There are two major issues with query-URLs recommendations: (i) the common clicks on URLs are limited for various queries (ii) though users may click the same URLs for two different queries, they may be irrelevant as that web documents may have different contents [9]. It is necessary to generate useful suggestions by solving these problems. It is required to discover users' information needs to organize queries with precise meaning. Users' search log provides information needs from users' click behaviour. If a certain retrieved result is clicked by the user, we cannot conclude that the clicked result is completely relevant to the user query since he has not seen the full document. But the brief description of the document i.e., snippet is shown to the user and is read by the user if he decides to click that document. It can be considered that snippet reflects user's information need.

Contribution: In this paper, Related Search Recommendation (RSR) framework is proposed to recommend related queries for user input query. This framework uses user feedback from click through log of search engine. User click through log is converted into feedback session with clicked and un-clicked URLs and it ends with last clicked URL in a session. Each clicked and un-clicked URLs of feedback session is converted into enriched documents by calculating term frequency-inverse document frequency for each term present in title and snippet of that URL. Pseudo documents are generated by merging all the enriched documents of a feedback session. Finally, optimized pseudo document is generated by combining all the pseudo documents for a given input query which reflects the user's information need. Recommendations are generated and ranked by combining query and terms for all the methods.

Organization: This paper is organized as follows: We have reviewed various query recommendation techniques using snippet under section 2. Section 3 describes the Background Work. Section 4 presents Related Search Recommendation Algorithm. Section 5 discusses experiment results, query recommendation results comparison and performance analysis. Finally, conclusions are presented in section 6.

2. RELATED WORK

Mostly, users access webpages by querying through search engines by which the performance of search engines is affected. In this work, we are recommending related search queries with user feedback session. In this sessions, clicked and un-clicked document's snippets are used to formulate related search queries. We need to calculate similarity between different words that exist in snippets to obtain the desired results. We have reviewed several papers related to measuring similarity between words and different techniques used for query recommendations using snippets in this section.

2.1 Measuring Similarity between two Words

Miao et al., [10] have developed query expansion method based on Rocchio's model. In this model, proximity information is modelled by proposed Proximity based Term Frequency *ptf* in the pseudo relevant documents. Expansion terms and their proximity relation with query terms are modelled by *ptf*. This model achieves better performance over position relevance model and classic Rocchio's model. Hamai et al., [11] have discussed a transformation function to measure semantic similarity between two given words. This approach uses page counts of documents title to measure similarity. This approach outperforms similarity measures defined over snippets.

Bollegala et al., [12] have presented an approach to calculate semantic similarity between words. Text snippets are used to obtain Lexico-syntactic patterns from a web search engine. Support vector machine is used to integrate page count based similarity score and lexico-syntactic patterns to generate semantic similarity measure. This method performs better than Information content measures and Edge counting WordNet based methods. Li et al., [13] have presented an approach to calculate semantic similarity between terms and multiword statement. A large web corpus is used to form an *isA* semantic network to provide contexts for the terms. The meaning of input terms is formulated by *K*-Medoids clustering algorithm and similarity is computed with *max-max* similarity function. This algorithm outperforms multi-word expressions pairs and pearson correlation coefficient on word pairs.

Bollegala et al., [14-15] have developed a relational model to calculate the semantic similarity between two words. Snippets of web pages are used to obtain Lexical patterns. Semantically related patterns are identified by extracted clusters from sequential pattern clustering algorithm. Mahalanobis distance is used to calculate semantic similarity between two words. This method outperforms all WordNet-based approaches [16-21].

2.2 Query Recommendation Techniques

Song et al., [22] have designed query suggestion method by using users' feedbacks in the query logs. Query-URL bipartite graph is constructed for click and un-click information. Random Walk with Restart (RWR) technique is applied on both the graphs. This framework gives better performance than pseudo-relevance feedback models ([23-25]) and random walk models. Kharitonov et al., [26] have focused on contextualisation framework for diversifying query suggestion. This framework utilizes the user's history query, the previously clicked and skipped documents and examines query suggestions. Mean Reciprocal Rank (MRR) is used as performance evaluation metric. This framework is compared with non-diversified ranking with the previous query, ranking with the previous query as a context and clicks and skips as context.

Ozetem et al., [27] have developed an approach to learn the probability with machine learning that a user may find a relevant follow up query after executing the input query. To measure relevance of follow-up query probabilistic utility function is used which relies on the query co-occurrence. This approach shows significance improvement over Mutual Information (MI) method. Broccolo et al., [28] have investigated a query suggestion algorithm that can cover long tail queries. This algorithm uses search shortcuts model to process a full text query which is indexed in user sessions recorded in a query log. This algorithm outperforms Query Flow Graph (QFG) and Cover Graph (CG) by providing most relevant query suggestions.

Zhang et al., [29] have developed an approach for query suggestion based on query search. This approach constructs an ordered set of search terms drawn from documents to create candidate query suggestions. It builds query suggestions separately for each potentially relevant document. This approach provides more relevant query suggestions for short queries as well as long queries. Gomex et al., [30] have designed a novel technique to visualize the collection of textual snippets returned from a web query. This technique constructs intuitive and meaningful layouts that optimize the placement of snippets by employing an energy function. Phan et al., [31] have introduced a method to process sparse and short documents by hidden-topic based framework on the web. This framework solves data sparseness and synonyms/homonyms problems of documents. Common hidden topics are determined from data sets to make documents short, less sparse and more topic-oriented.

He et al., [32] have presented a novel sequential query prediction approach for understanding users' search intent and recommending queries. A sequential probabilistic model called Mixture Variable Memory Markov Model is developed for online query recommendation. Jiang et al., [33] have presented query recommendation method based on Query Hashing (QH). QH generates many similar and dissimilar query-pairs as prior knowledge from query sessions. QH model is compared with hashing-based methods, SimHash, Kernelized Locality Sensitive Hashing and Inverted list. This method achieves best results in terms of efficiency and recommendation performance.

Li et al., [34] have proposed a query suggestion approach. In the learning step, a generative probabilistic model is obtained by learning external knowledge gained from the web dataset for web queries. Latent semantic topic model is used to organize the co-occurrence of the web queries. Posterior distribution of hidden topics is obtained for each candidate query with this model. This approach gives better query suggestions than URL model and comparable results with term feature model. Liu et al., [35] have proposed a snippet click model for query recommendation. This model determines information need of users from search logs. The clicked snippets are used to represents the information need of the users and with this judgement snippet click models are constructed. Click-through rate and click amount are used as metrics to evaluate the performance of the algorithm. The proposed algorithm is providing more efficient recommendation than Biadu and Sogou search engines.

Table 1 shows the comparison of closely related works with our proposed approach.

Table 1. Comparisons of Related Works

Author	Concept	Advantages	Disadvantages
Li et al., [34]	Suggest topically related web queries using hidden topic model	Provides better query suggestions than URL model and Comparable Results with term feature model	Training dataset need to be generated to find topic of web queries from external data source.
Zhang et al., [29]	Provide improved query suggestion by query search	Provides more relevant query suggestion for short queries as well as long queries compared to suggestion by query search	User feedback is not considered
Miao et al., [10]	Query expansion based on proximity based Rocchio's model	This model achieves better performance over position relevance model	The exact relationship between the window size factor and information of collection is not fixed
Lu et al., [36]	Inferring User Search goals with Feedback Sessions	User search goals can be utilized in query recommendations	Finds Personalized Search goals.
Liu et al., [35]	Provide query recommendation based on snippet click model	Provides more effective recommendations than Biadu and Sogou search engines	Only click information is used to create model
Rocchio [37]	Query expansion with user feedback	Considers user feedback and generates relevant terms for query expansion	Fails to classify multimodal classes and relationship
Our work	Recommending related search with user feedback session and semantic similarity between words	Provides Semantically related search to inputs and this approach can be extended to generate multiple related words	

3. BACKGROUND

In this section, brief review about feedback session and pseudo-document is presented. Co-occurrence measures Dice, Jaccard, Pointwise Mutual Information (PMI) and Overlap are explained to calculate semantic similarity. WordNet based measures are discussed to calculate semantic similarity. Rocchio's model [37] and Snippet Click model [35] are compared with our work.

3.1 Co-occurrence Measures to Compute Semantic Similarity

The notation $P(Q)$ is used to denote the page counts for the query Q in search engine.

WebJaccard(T_1,T_2) is defined as

$$WebJaccard(T_1,T_2) = P(T_1 \cap T_2) / P(T_1) + P(T_2) - P(T_1 \cap T_2) \qquad \textbf{(1)}$$

Here, $P(T_1 \cap T_2)$ denotes co-occurrence of terms T_1 and T_2.

***$WebDice(T_1,T_2)$* is defined as**

$WebDice(T_1,T_2) = 2\,P(T_1 \cap T_2) / P(T_1) + P(T_2)$ ***(2)***

***$WebOverlap(T_1,T_2)$* is a natural modification to the Simpson coefficient and is defined as**

$WebOverlap(T_1,T_2) = P(T_1 \cap T_2) / min(P(T_1), P(T_2))$ ***(3)***

Point-wise mutual information (PMI) is a measure of association used in information theory and statistics. It is intended to reflect the dependence between two probabilistic events.

***WebPMI* is defined as a variant form of point-wise mutual information using page counts as**

$WebPMI(T_1,T_2) = log_2[(P(T_1 \cap T_2)/N) / (P(T_1)/N)(P(T_2)/N)]$ ***(4)***

3.2 WordNet based Semantic Similarity

WordNet [38] developed by Princeton University is a lexical database in English. It is well suited for similarity measures, since it organises verbs, nouns, adjectives and adverbs with variation in semantic relations into synonym sets (synsets) by representing one concept. It uses *is-a* relation to organise noun and verbs into hierarchies. Semantic relations used by WordNet are autonomy, synonymy, member, hyponymy, domain, relation, cause and similar and so on. *wup*(Wu and Palmer 1994) , *lch* (Leacock and Chodorow 1998) and *path* calculates similarity with path length. *lin* (Lin 1998), *res* (Resnik 1995) and *jcn* (Jiang and Conrath 1997) measures similarity with information content which is corpus based measure of the specificity of concept. WordNet also provides *is-made-of*, *has-part*, *is-an-attribute-of* etc. non-hierarchical relations. With this additional relations, measures of relatedness is also supported by WordNet which are *lesk*(Banerjee and Pedersen 2003), *hso*(Hirst and St-onge 1998) and *vector*(Patwardhan 2003).

3.3 Rocchio's Model

Rocchio's Model [37] uses relevant and irrelevant URLs identified by users in search log to extend the input query. The extended query is used to carry out retrieval again. These URLs are converted into documents with title and snippet. Let the input query be q, the set of related documents accepted by users be D_r and the set of non-related documents be D_{ir}. The expanded query q_e is computed by using equation 5. Here, a, b and c are parameters and their traditional values are 1, 0.8 and 0.1 respectively. Related documents are given more importance than non-related documents. The importance of terms which are present in both related and non-related documents and only in non-related documents is reduced by subtraction.

$$q_e = aq + \frac{b}{\mid D_r \mid}\sum_{d_r \in D_r} d_r - \frac{c}{\mid D_{ir} \mid}\sum_{d_{ir} \in D_{ir}} d_{ir} \qquad (5)$$

3.4 Snippet Click Model

Global scale snippet click model [35] uses clicked URLs CLK_{url} from the user search log for a given input query q. Snippets are extracted for CLK_{url} and converted into documents D. Each keyword Term Frequency (TF) is calculated in documents D. Top N keywords with largest

TFs is used as recommendation candidates. These N keywords are combined with the input query q and displayed as recommendations.

4. RELATED SEARCH RECOMMENDATION FRAMEWORK AND ALGORITHM

4.1 Problem Definition and Assumptions

Given a user input query q and user click through log lg from the web search engine S, our objective is to recommend expanded queries q_e. It is assumed that the user is online while entering input query and considers only top-50 retrieved search results.

4.2 Related Search Recommendation Framework

In this section, related search recommendation framework is presented as shown in Figure. 1. Feedback sessions are generated for a given query from the user search logs and pseudo-documents are mapped to it.

Feedback Sessions: Generally, a session can be defined as a list of consecutive queries to correlate particular user search knowledge and clicked URLs for web search [39]. Lu et al. [36] have focused on deriving a feedback session with single query. In this work, query suggestions are generated for a query and hence a single session with single query is suitable and is different from the traditional session.

The feedback session is defined with both clicked and un-clicked documents and it ends with last clicked documents in a session. This feedback session gives information that all the URLs have been examined and assessed by users before the last click. Figure 2 shows an example of feedback session for query *bank exam*. The left part is the 19 search results of the query *bank exam* and the right part is a user's click series, with 1 as clicked URLs by user and 0 as un-clicked. Here, a single session includes 19 URLs, while the feedback session includes only 15 URLs. The feedback session consists of four clicked and six un-clicked URLs. Inside this session, the clicked URLs display that is relevant to the users and the un-clicked URLs display that is ir-relevant to the users. The un-clicked URLs followed by the last clicked URL are ignored in the feedback session since it is not assured that users have scanned or not.

Search Results	Click Sequence
Http://www.bankexamsindia.com/	0
http://www.jagranjosh.com/bank-exams	1
http://www.ibps.in/	1
http://linkingsky.com/government-exams/bank/	0
http://www.bankexamstoday.com/	0
http://www.freejobalert.com/upcoming-exam-dates-of-various-jobs/1835/	1
http://www.freejobalert.com/upcoming-notifications/21614/	0
http://www.time4education.com/bankexams/	0
http://www.bankjobsindia.net/upcoming-bank-exams-2014-in-india-and-latest-bank-jobs/	0
http://www.successcds.net/Bank-PO/	0
https://bankerschoice.talentsprint.com/	0
https://bankerschoice.talentsprint.com/indian-bank-exams-ibps-sbi/quant-formula	0
https://www.sbi.co.in/user.htm?action=viewsection&lang=0&id=015110	0
http://www.bankingexamseasy.com/	0
http://www.iibf.org.in/scripts/archives_exam_schedule.asp	1
https://www.facebook.com/IbpsExamGuru	0
http://www.tcyonline.com/exam-preparation-bank-po-preparation-tests/100241/bank-po-clerical	0
http://www.eenadupratibha.net/Pratibha/ibps.aspx	0
http://www.sbirecruitment2014.org/	0

Figure 1. An example of feedback session for query *bank exam* in rectangular box.

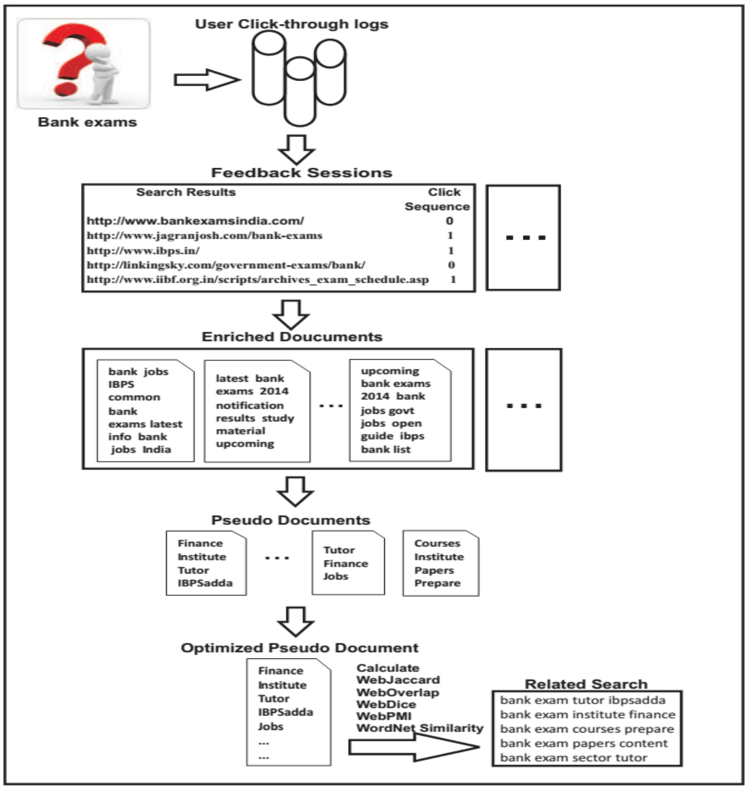

Figure 2. Related Search Recommendation Framework

Generate Enriched Documents from Feedback Sessions: It is not suitable to use feedback sessions directly to obtain meaningful information for suggestions as it may differ for different search history and queries. Usually, users have ambiguous keywords in their minds to represent their information need. Hence, it is not a good idea to generate relation between the user query keywords for recommendations. Enriched documents [36] are generated from feedback sessions and this enriched document is used to locate keywords that appear in snippets clicked and un-clicked documents in feedback session. The method of generating enriched document is given in function 1. T_v and S_v vectors are given in given in equation 6 and 7.

Function 1 : Enriched Document

Function : EnrichedDocument(Feedback Session *FS*)
for *each URL u in Feedback Session FS* **do**
Extract Title *T* and Snippet *S*
Generate T_p from *T* after stopwords removal, transforming all letters to lowercase and applying stemming
Generate S_p from *S* after stopwords removal, transforming all letters to lowercase and applying stemming
Generate T_v and S_v vector by calculating Term Frequency-Inverse Document Frequency (TF-IDF) for T_p and S_p as shown in Equation 5 and 6 respectively
Generate Enriched Document *ED* by the weighted sum of T_v and S_v as shown in Equation 7
end

$$T_v = [t_{w1}, t_{w2}, \ldots, t_{wm}] \quad (6)$$
$$S_v = [t_{w1}, t_{w2}, \ldots, t_{wn}] \quad (7)$$

Where, t_{wm} = Term Frequency Inverse Document Frequency (TF-IDF) value of the mth term in URL's title and t_{wn} = TF-IDF value of the nth term in the URL's snippet. The enriched document is defined as given in equation 8.

$$ED = w_tT_v + w_sT_s = [ed_w1, ed_w2 \ldots. ed_{wk}] \quad (8)$$

Where, w_t is the weight of the title, w_s is the weight of the snippet, and ed_{wi} indicates the importance of ith term in the URL. As title directly represents the URL information, it is necessary to give more importance to title terms than the snippet terms, and therefore w_t is set to 2 and w_s is set to 1. Five enriched documents are generated for five URLs of feedback session (see Figure. 1).

Generate Pseudo-Documents from Enriched Documents: For a feedback session, each URL is converted into enriched document. This document contains frequent terms that appears in clicked and un-clicked documents. For each feedback session, a Pseudo-Document is generated from its enriched documents. The method of generating Pseudo-Document(PD) is shown in function 2.

Function 2 : Pseudo Document

Function : PseudoDocument(FeedBack Session *FS*, Enriched Document *ED*
for *each FeedBack Session FS* **do**
Group Enriched Document of *FS* as $ED_{clk} = [ed_{w1clk}, ed_{w2clk}, \ldots, ed_{wmclk}]$ and $ED_{unclk} = [ed_{w1unclk}, ed_{w2unclk}, \ldots, ed_{wnunclk}]$ of the clicked and un-clicked URLs respectively.
for *each term in* $(ED_{clk} \cup ED_{unclk})$ **do**
Generate Pseudo Document *PD* by optimizing the value of term such that *t* belongs to ED_{clk} get more importance than *t* belongs to *EDunclk* as given in Equation 8.
end
end

The generated $PD = [ed_{w1}, ed_{w2}, \text{---}, ed_{wp}]$.

$$Ed_w = \underset{edw}{arg\ min} \{\sum_{M} [ed_w - ed_{wclk}]^2 - \lambda \sum [ed_w - ed_{wunclk}]^2\} \quad (9)$$

Here, ed_w is the optimized term in Pseudo Documents, ed_{wclk} is the term from clicked enriched documents, ed_{wunclk} is the term from un-clicked enriched documents and λ is a parameter balancing the importance of clicked and un-clicked URLs. λ is set to 0.5 because if λ is set to a small value, then un-clicked URLs importance is reduced and if λ value is too large then un-clicked URLs dominates the value of ed_w. A pseudo document generated from five enriched documents is shown in Figure. 1.

Generate Optimized Pseudo-Document from Pseudo Documents: The pseudo document reflects both the relevant and irrelevant documents to the users. Optimized Pseudo-document is generated by combining all the pseudo documents for an input query. The method for generating optimized pseudo-document is shown in function 3. N is set to 10 as we observe that the top 10 terms are representing the users' information need.

Function 3 : Optimized Pseudo Document

Function : OptPseudoDocument(*Pseudo Document PD*)
    for *each PD* **do**
        Select top *N* terms
        Compute occurrence of each term in all the PD_s
        Arrange the terms in descending order of occurrence and select top *N* terms to optimized *PD*
    end

Semantic similarity is calculated between optimized pseudo-document terms by WebJaccard, WebDice, WebPMI, WebOverlap methods and WordNet based similarity measures as discussed in section 3. Recommendation results are generated and ranked by combining query and terms for all the methods. These results are evaluated in section 5 performance evaluation.

4.3 Related Search Recommendation Algorithm

In this section, we present Related Search Recommendation (RSR) Algorithm as shown in algorithm 1

Algorithm 1: RSR : Related Search Recommendation

Input : input query *q*, user click through log *l*
Output : related queries *rq* = <*1 ...k*>
 	
    begin
      for *input query q* **do**
          Select Feedback Sessions $FS = \{fs1, fs2, \ldots, fsn\}$ from user click through log *l*
          for *each feedback session fs in FS* **do**
              Generate Enriched Document $ED = (ed_1, ed_2, \ldots, ed_m)$ by *EnrichedDocument(Feedback Session fs)*
              Generate Pseudo document *pd* with *PseudoDocument(Feedback Session fs, Enriched Document ED)*
              Add *pd* to *PD* <*1 ... l*>
          Generate Optimized Pseudo Document $OPD = (opd_{w1}, opd_{w2}, \ldots, opd_{wn})$ with *OptPseudoDocument (Pseudo Document PD)*
      for *each* opd_{wi} *in OPD of size n* **do**
          Calculate semantic similarity of opd_{wi} $(1 < i < n)$ and opd_{wj} $(1 < j < n)$ with WebOverlap
          $rq_{Overlapi} = q + opd_{wi} + opd_{wj}$
      $rq = rq_{Overlapi}$

5. EXPERIMENTS

5.1 Data Collection

To evaluate our proposed method 95 students participated and each student is assigned 5 queries to collect the feedback session. A Google middleware is implemented to monitor the user clicks. The top 50 search results from Google are retrieved for the submitted query. The title and web-snippets of resulting search are presented to the user as the snippets provide more information about the documents and help them to guide to the click URLs. Feedback sessions are generated through the clicked information of a user for a given input query. Table 2 shows the statistics of the clicked information of users for this experiment.

5.2 Experiment Setup

The setup of Related Search Recommendation (RSR) framework is as follows : Feedback sessions are generated for a given input query from the user click through *log* as discussed in section 5.1. Each URL in the feedback session is enriched with title and snippet terms after removing stop words and applying stemming. Terms are weighed using Term Frequency-Inverse Document Frequency (TF-IDF) as explained in function 1. Enriched documents of a feedback session are classified into clicked and un-clicked documents. Pseudo documents are generated by the equation 9. Similarly, Pseudo documents are generated for all the feedback sessions for an input query. Optimized Pseudo document is generated by combining all the pseudo documents as shown in function 3. Optimized Pseudo document has top-10 terms which reflect the user's information need. Semantic similarity between these terms t_s are calculated by WebJaccrad, WebDice, WebPMI, WebOverlap methods and WordNet based similarity measures as discussed in section 3. Recommendations are generated and ranked by combining query and terms t_s for all the methods.

Table 2. Statistics of Clicked Information of Users

Total users	95
Total queries allocated to each user	5
Total test queries	100
Total unique queries	100
Total URLs retrieved for a query	50
Total URLs retrieved	5000
Average feedback sessions for a query	5
Average clicked URLs for a query	9.732
Average un-clicked URLs for a query	40.268
Total words extracted from title for a query	23048
Average words extracted from title for a query	230
Total words extracted from snippet for a query	38098
Average words extracted from snippet for a query	380
Total words extracted	61146
Average words extracted for a query	611

The setup of Rocchio's model is as follows: User identified relevant and irrelevant URLs are partitioned from the user click through *log* for a given input query. These URLs are converted into documents with title and snippet. Stop words removal and stemming is applied for these documents to reduce noise. Expanded queries are generated by equation 5.

The setup of Snippet Click Model (SCM) is as follows: All the clicked URLs from user click through *log* are obtained for a given input query. Snippets are extracted from these URLs. Top-10 keywords are extracted by calculating term frequency of the terms present in snippets. Query recommendations are generated by combining the input query with extracted keywords.

To examine the effectiveness of considering only clicked URLs in our proposed method (click-RSR), enriched documents are generated with only clicked URLs. Pseudo documents are generated by setting λ value to zero in equation 9 to remove the effect of un-clicked URLs. Optimized Pseudo document is generated by combining all the pseudo documents as shown in function 3. Optimized Pseudo document has top-10 terms. Semantic similarity between these terms t_s are calculated by WebJaccrad, WebDice, WebPMI and WebOverlap methods as discussed in section 3. Recommendations are generated and ranked by combining query and terms t_s for all the methods.

5.3 Query Recommendation Results

Top-5 recommendation results of Rocchio's model, Snippet Click model, Click-RSR and our RSR algorithm is shown in Table 6. Only terms are displayed in recommendation results due to space restriction. The actual recommendations for all models are query + terms. For query *bank exam*, recommendations for Rocchio's model are *bank exam finance*, *bank exam institute*, *bank exam tutor*, *bank exam ibpsadda* and *bank exam gr8ambitionz*. Recommendations for Snippet Click Model are *bank exam bank*, *bank exam competitive*, *bank exam exam*, *bank exam notification*, *bank exam awareness*. Recommendations for Click-RSR are *bank exam question bank*, *bank exam question tutor*, *bank exam papers bank*, *bank exam shortcuts bank*, *bank exam bank facebook*. Recommendations for RSR algorithm are *bank exam tutor ibpsadda, bank exam institute finance, bank exam courses prepare, bank exam papers content*, *bank exam sector tutor*.

5.4 Performance Analysis

From the result shown in Table 6, it is observed that RSR algorithm recommends related queries to the given input query. Hundred test queries from various topics like Science, Shopping, and Healthcare have been included.

Lu. et al. [36] have discovered different users search goals for a query by using feedback session. These search goals can be utilized in query recommendations. Feedback sessions are utilized in our work and the performance of RSR algorithm is compared with different recommendation methods like classical Rochhio's model [37], Snippet Click Model [35] and modified approach of RSR algorithm considering only clicked URLs. The setup of the experiment is discussed in section 5.2. We have adopted Click Through Rate (CTR) method used in [35] to evaluate related search recommendations. CTR is the percentage of ever clicked recommendations in all recommendations for a given query. The set of students who participated in collecting click through log also participated in computing CTR as they can judge the recommendation results effectively. CTR is used to evaluate whether the recommendation is clicked by the user and a higher CTR value proves the effectiveness of the algorithm.

CTR is calculated for top-5 recommendations results generated with WebJaccrad, WebDice, WebPMI and WebOverlap methods for RSR algorithm. The average value of CTR and ranked recommendations results is depicted in Figure. 3 for all the methods. The average CTR value of Top-5 recommendations is displayed in Table 3. CTR is also calculated for WordNet different semantic similarity measures. The average CTR value of Top-5 recommendations is displayed in Table 4. It is observed from WordNet similarity measures that few terms are not available in WordNet database, hence are not able to find out similarity between two terms. It is observed from Table 3 and 4 that recommendations ranked with WebOverlap method have higher CTR value. Hence, WebOverlap method is adopted to rank RSR recommendations.

Similarly, CTR is calculated for top-5 recommendations results generated with WebJaccrad, WebDice, WebPMI and WebOverlap methods for click-RSR algorithm. The average CTR value of Top-5 recommendations is displayed in Table 3. It is observed that recommendations ranked with WebOverlap method have higher CTR value. Hence, WebOverlap method is adopted to rank click-RSR recommendations.

To compare RSR algorithm with other models, the average CTR value and ranked recommendations are displayed in Figure 4. The average CTR value of Top-5 recommendations for all the models are depicted in Table 5. It is observed that RSR algorithm has highest CTR value in comparison with other models.

It is observed that the CTR value of the RSR algorithm increases by 25% in comparison with SCM. The major difference between our algorithm and SCM is consideration of un-clicked URLs along with clicked URLs, while SCM considers only clicked URLs. Even the weighing of terms in SCM is limited to term frequency which is further optimized in RSR algorithm.

Table 3. Average CTR value for Top-5 Recommendations for RSR and Click-RSR Algorithm

	WebPMI	WebJaccard	WebDice	WebOverlap
RSR	75.43	76.00	77.33	79.15
Click-RSR	73.72	72.72	71.20	74.02

Table 4. Average CTR value for Top-5 Recommendations for WordNet similarity measures

lch	wup	path	res	lin	jcn	hso	lesk	vector
73.36	74.87	67.89	73.76	62.18	45.4	70.70	76.36	67.10

Table 5. Average CTR value for Top-5 Recommendations for all models

SCM [35]	Rocchio [37]	Click -RSR	RSR
54.06	55.03	73.82	79.15

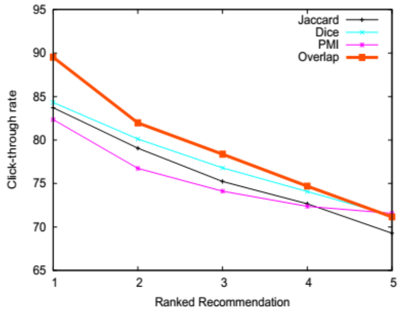

Figure 3. CTR vs. Ranked Recommendation Results

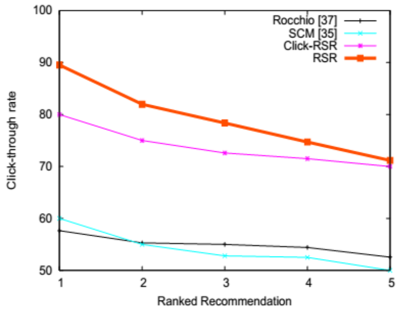

Figure 4. CTR Comparison with other Models

The CTR value of the RSR algorithm increases by 24% in comparison with Rocchio's model. The difference between two approaches are as follows : 1) In our approach, feedback sessions are limited to the last clicked URL as the left-out URLs may not be of user's interest. 2) Click through data is considered as sessions in RSR algorithm while in Rocchio's model it is treated as group of clicked/un-clicked URLs.

The CTR value of RSR algorithm increases by 5% in comparison with click-RSR. The major difference between RSR algorithm and click-RSR is consideration of only clicked URLs in the feedback session. It is observed from the recommendations result from RSR algorithm that the terms from un-clicked URLs are also present. It is observed that top-5 recommendations from RSR algorithm for 100 test queries consists of about 23.5% of overall terms from the un-clicked URLs in the feedback sessions, which shows the importance of the un-clicked URLs scanned by users. Thus, RSR algorithm outperforms the clickRSR.

Table 6. Related Search Recommendation Results Comparison

Sr.No	Query	Rocchio's model [37]	Snippet Click Model [35]	Click-RSR	Proposed RSR algorithm
1	bank exam	finance institute tutor ibpsadda gr8ambitionz	bank competitive exam notifications awareness	question bank question tutor papers bank shortcuts bank bank facebook	tutor ibpsadda institute finance courses prepare papers content sector tutor
2	apartment	budapest zillow decor adina genuine	budapest apartment zillow 123844 luxury	budapest adina zillow rental zillow genuine rental genuine realestate properties	realestate properties realestate commonfloor realestate luxury properties commonfloor properties luxury
3	weather	wiz kids welcome internet temperatures	forecast weather web local dallas	history weather wiz kids wiz weather web welcome web weather	bbc forecasts bbc animated oceanic atmospheric forecasts australia forecasts temperatures
4	camera	nokia android pocket grip 1020	sony lines cameras nokia github	nokia grip nokia 1020 nokia lumia grip 1020 grip lumia	analog lense analog flash cctv lense cctv flash canon lense
5	online recharge	tariffs cellone	mtnl prepaid	payments cellone recharge tariffs	landline cellone state personal

		personal state landline	bsnl reliance services	portal cellone prepaid tariffs banking personal	landline postpaid landline huch landline packs
6	free music	jamendo songza composition archive jango	music appears automated purple listen	songza worthy composition notation composition musescore notation musescore streaming archive	downloads jango downloads limewire beats freeplay beats uncopyrighted beats song
7	Solar system	youtube wikipedia meteorites characteristics astronomy	tour ice bbc phet velocity	tour solar youtube solar youtube witness youtube peaceful youtube tues	asteroids image kidsastronomy image meteorites image views image visualizer image
8	maths	mathworld ask webs level extensive	Alp ha wolfram puzzles guardian drexel	mathworld webs mathworld wolfram webs wolfram ask forum warwick mathworld	skills watch american homepage youtube trick youtube fast trick fast
9	wedding	facebook fairy tale disneys registries	ann pretty wedding registry nordstrom	fairy tale fairy disneys tale disneys gifts fairytale nordstrom wedfolio	blog cards fairy tale registries mywedding blog etiquette blog popular

6. CONCLUSIONS

In this work, we have presented Related Search Recommendation (RSR) algorithm to suggest related queries to given input query by using feedback session from user click through log. Each feedback session is converted into enriched documents. Pseudo Documents are generated by combining all the enriched documents of a feedback session. Optimized Pseudo Document is generated by combining all the Pseudo Documents for a given input query, which reflects the user's information need. Semantic similarity is calculated by WebJaccrad, WebDice, WebPMI and WebOverlap methods for terms present in the optimized Pseudo Document. Recommendations are generated and ranked by combining query and terms for all the methods. Simulations are performed on click through log generated by displaying title and snippet to the students of our college and compared with Rocchio's model, Snippet Click Model and Click-RSR. Click Through Rate (CTR) is used as a performance evaluation metric. Simulation results show that RSR algorithm outperforms Rocchio's model, Snippet Click Model and ClickRSR by providing higher CTR value. Further, this work can be extended to classify the search results into different topics.

REFERENCES

[1] M. P. Kato, T. Sakai, and K. Tanaka, "When Do People Use Query Suggestion? A Query Suggestion Log Analysis," *Journal of Information Retrieval*, vol. 16, no. 6, pp. 725–746, December 2013.

[2] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz, "Analysis of a Very Large Web Search Engine Query Log," *In SIGIR Forum*, pp. 6–12, 1999.

[3] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen, and H. Li, "Context-Aware Query Suggestion by Mining Click-Through and Session Data," *KDD '08 : In the Proceedings of 14th International ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 875–883, 2008.

[4] Q. Mei, D. Zhou, and K. Church, "Query Suggestion Using Hitting Time," *CIKM '08: In the Proceedings of 17th ACM Conference on Information and Knowledge Management*, pp. 469–477, 2008.

[5] H. Ma, M. R. Lyu, and I. King, "Diversifying Query Suggestion Results," *AAAI '10 : In the Proceedings of 24th AAAI International Conference on Artificial Intelligence*, pp. 1399–1404, 2010.

[6] J. Guo, X. Cheng, G. Xu, and H. Shen, "A Structured Approach to Query Recommendation with Social Annotation Data," *CKIM '10 : In the Proceedings of 19th ACM International Conference on Information and Knowledge Management*, pp. 619–628, 2010.

[7] Y. Song, D. Zhou, and L.-w. He, "Post Ranking Query Suggestion by Diversifying Search Results," *SIGIR '11 : In the Proceedings of 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 815–824, 2011.

[8] Z. Kunpeng, W. Xiaolong, and L. Yuanchao, "A New Query Expansion Method based on Query Logs Mining," *International Journal on Asian Language Processing*, vol. 19, no. 1, pp. 1–12, 2009.

[9] Y. Chen and Y.-Q. Zhang, "A Personalized Query Suggestion Agent based on Query-Concept Bipartite Graphs and Concept Relation Trees," *International Journal of Advanced Intelligence Paradigms*, vol. 1, no. 4, pp. 398–417, June 2009.

[10] J. Miao, J. X. Huang, and Z. Ye, "Proximity based Rocchios Model for Pseudo Relevance Feedback," *SIGIR' 12: In the Proceedings of 35th ACM International Conference on Research and Development in Information Retrieval*, pp. 535–544, August 2012.

[11] M. S. Hamani and R. Maamri, "Word Semantic Similarity based on Document's title," *In the Proceedings of 24th IEEE International Workshop on Database and Expert Systems Applications*, pp. 43–47, 2013.

[12] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring Semantic Similarity between Words using Web Search Engines," *WWW '07: In the Proceedings of 16th International Conference on World Wide Web*, pp. 757–766, 2007.

[13] P. Li, H. Wang, K. Q. Zhu, Z. Wang, and X. Wu, "Computing Term Similarity by Large Probabilistic *isa* Knowledge," *CIKM '13 : In the Proceedings of 22nd International Conference on Information and Knowledge Management*, pp. 1401–1413, 2013.

[14] B. Danushka, M. Yutaka, and I. Mitsuru, "A Relational Model of Semantic Similarity between Words using Automatically Extracted Lexical Pattern Clusters from the Web," *EMNLP '09 : In the Proceedings of International Conference on Empirical Methods in Natural Language Processing*, pp. 803–812, 2009.

[15] D. Bollegala, Y. Matsuo, and M. Ishizuka, "A Web Search Engine-Based Approach to Measure Semantic Similarity between Words," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 977–990, July 2011.

[16] P. Resnik, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy," *IEEE Transactions on Knowledge and Data Engineering*, pp. 448–453, 1995.

[17] D. Lin, "An Information Theoretic Definition of Similarity," *IEEE Transactions on Knowledge and Data Engineering*, pp. 296–304, 1998.

[18] C. Leacock and M. Chodorow, "Combining Local Context and Wordnet Similarity for Word Sense Disambiguation," *WordNet : An Electronics Lexical Database*, pp. 265–283, 1998.

[19] J. J. Jiang and D. W. Conrath, "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy," *In the Proceedings of International Conference on Research in Computational Linguistics*, pp. 19–33, 1997.

[20] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pas¸ca, and A. Soroa, "A Study on Similarity and Relatedness using Distributional and Wordnet-Based Approaches," *In the Proceedings of Human*

Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 19–27, 2009.

[21] G. Hirst and D. St-Onge, "Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms," *WordNet : An Electronics Lexical Database*, pp. 305–332, 1998.

[22] Y. Song and L.-w. He, "Optimal Rare Query Suggestion with Implicit User Feedback," *WWW '10: In the Proceedings of 19th International Conference on World Wide Web*, pp. 901–910, April 2010.

[23] N. Craswell and M. Szummer, "Random Walks on the Click Graph," *SIGIR '07: In the Proceedings of 30th Annual International ACM SIGIR Conference on Research and Development in Information Retreival*, pp. 239–246, 2007.

[24] F. Radlinski and T. Joachims, "Query Chains : Learning to Rank from Implicit Feedback," *KDD '05: In the Proceedings of 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 239–248, 2005.

[25] G.-R. Xue, H.-J. Zeng, Z. Chen, Y. Yu, W.-Y. Ma, W. Xi, and W. Fan, "Optimizing Web Search using Web Click Through Data," *CIKM '04: In the Proceedings of 13th ACM International Conference on Information and Knowledge Management*, pp. 118–126, 2004.

[26] E. Kharitonov, C. Macdonald, P. Serdyukov, and L. Ounic, "Intent Models for Contextualizing and Diversifying Query Suggestions," *CIKM '13: In the Proceedings of 22nd ACM Conference on Information and Knowledge Management*, pp. 2303–2308, 2013.

[27] U. Ozertem, O. Chapelle, P. Donmez, and E. Velipasaoglu, "Learning to Suggest: A Machine Learning Framework for Ranking Query Suggestions," *SIGIR '12: In the Proceeding of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 25 –34, August 2012.

[28] D. Broccolo, L. Marcon, F. M. Nardini, R. Perego, and F. Silvestri, "Generating Suggestions for Queries in the Long Tail with an Inverted Index," *International Journal on Information Processing and Management*, vol. 48, no. 2, March 2012.

[29] X. Zhang, S. Zilles, and R. C. Holte, "Improved Query Suggestion by Query Search," *In the Proceedings of the 35th Annual German Conference on Artificial Intelligence*, pp. 205–216, September 2012.

[30] E. Gomez-Nieto, F. San Roman, P. Pagliosa, W. Casaca, E. S. Helou, M. C. F. de Oliveira, and L. G. Nonato, "Similarity Preserving Snippet-Based Visualization of Web Search Results," *IEEE Transactions on Visualization and Computer Graphic*, vol. 20, no. 3, pp. 457–470, March 2014.

[31] X.-H. Phan, C.-T. Nguyen, D.-T. Le, L.-M. Nguyen, S. Horiguchi, and Q.-T. Ha, "A Hidden Topic-Based Framework toward Building Applications with Short Web Documents," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 961–976, July 2011.

[32] Q. He, D. Jiang, Z. Liao, S. C. Hoi, K. Chang, E.-P. Lim, and H. Li, "Web Query Recommendation *via* Sequential Query Prediction," *In the Proceedings of IEEE 25th International Conference on Data Engineering*, pp. 1443–1454, April 2009.

[33] Q. Jiang and M. Sun, "Fast Query Recommendation by Search," *AAAI Conference on Artificial Intelligence*, pp. 1192–1197, 2011.

[34] L. Li, G. Xu, Z. Yang, P. Dolog, Y. Zhang, and M. Kitsuregawa, "An Efficient Approach to Suggesting Topically Related Web Queries using Hidden Topic Model," *International Journal on World Wide Web*, vol. 16, no. 3, pp. 273–297, May 2013.

[35] Y. Liu, J. Miao, M. Zhang, S. Ma, and L. Ru, "How do Users Describe their Information Need : Query Recommendation based on Snippet Click Model," *International Journal on Expert Systems with Applications*, vol. 38, no. 11, pp. 13 874–13 856, October 2011.

[36] Z. Lu, H. Zha, X. Yang, W. Lin, and Z. Zheng, "A New Algorithm for Inferring User Search Goals with Feedback Sessions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 502–513, March 2013.

[37] J. J. Rocchio, "Relevance Feedback in Information Retrieval," *In G. Salton, The SMART Retrieval System : Experiments in Automatic Document*, pp. 313–323, 1971.

[38] C. Fellbaum, "WordNet: An Electronic Lexical Database," *Language, Speech, and Communication. MIT Press, Cambridge, USA*, 1998.

[39] R. Jones and K. L. Klinkner, "Beyond the Session Timeout: Automatic Hierarchical Segmentation of Search Topics in Query Logs," *ACM Conference on Information and Knowledge Management*, pp. 699–708, 2008.