# Chapter 6
# Related Search Recommendation with User Feedback Session


**K. R. Venugopal and Sejal Santosh Nimbhorkar**

**Abstract** Keyword-based search is an extensively used method to discover knowledge on the Web. Generally, Web users are unable to arrange and define input queries relevant to their search because of adequate knowledge about the domain. Therefore, the input queries are normally ambiguous and short. Query suggestion is a method to recommend queries related to the user input query that helps them to locate their required information more precisely. It helps the search engine to provide relevant answers and meet users needs. Usually, users query keywords are ambiguous, therefore it is not good to use users query keyword in suggestion. In this chapter, Related Search Recommendation (RSR) framework is presented that determines keywords presented in un-clicked and clicked documents in the feedback session. Feedback sessions are used to retrieve users need in terms of Pseudo documents. Semantic similarity is computed between the terms in the Pseudo document. The semantic terms are used for suggestions. The presented method provides semantically related search queries for the user input query. Results show that the RSR method outperforms Rochios model and Snippet Click Model.

## 6.1 Introduction

Web data keeps expanding and is available in various data forms because of the 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 the future new search.

K. R. Venugopal (✉)
Bangalore University, Jnana Bharathi, Bengaluru 560056, India
e-mail: venugopalkr@gmail.com

S. Santosh Nimbhorkar
BNM Institute of Technology, Banashankari, Bengaluru 560070, India
e-mail: sej_nim@yahoo.co.in



K. R. Venugopal et al., *Web Recommendations Systems*,
https://doi.org/10.1007/978-981-15-2513-1_6

Query suggestion is an efficient way to enhance keyword-based search, which is extensively useful for 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 and (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.

Through query suggestion, search engines have succeeded in obtaining Web information for users, but the keyword-based search is not able to organize 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 the information needed of user search and sometimes ambiguous in meaning expression. Because of insufficient knowledge about domain, users find it difficult to organize and define appropriate input queries. Then the user has to rephrase the query words or query frequently, which affects the search performance.

In ([3–8]), the 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 a 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 the user's information need.

Lu et al. [10] have designed a method to determine the goal of user search for a query. These search goals are obtained by clustering the proposed feedback sessions. The clicked and un-clicked documents with the last clicked document represents feedback session in a user search log. Pseudo documents are obtained by mapping feedback sessions to reflect the information needs of the users effectively. Pseudo documents are clustered using $k$-mean algorithm to derive search goals of users.

In this chapter, 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 the title and snippet of that URL. Pseudo documents are generated by merging all the enriched documents of a feedback session. Finally, the 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.

## 6.2 Related Works

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

### *6.2.1 Measuring Similarity Between Two Words*

Miao et al. [11] have developed a 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 is modelled by *ptf*. Window-based, kernel-based and Hyperspace Analogue to Language (HAL) methods are proposed as proximity measures for evaluating the relationship between query terms and expansion terms. This model achieves better performance over position relevance model and classic Rocchio's model.

Hamai et al. [12] 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. [13] 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. [14] have presented an approach to calculate the 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 multiword expression pairs and Pearson correlation coefficient on word pairs.

Bollegala et al. [15, 16] 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 ([17–22]).

### *6.2.2 Query Recommendation Techniques*

Song et al. [23] 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 to both the graphs. The category of URLs is used to construct correlation matrix for URLs. Optimal query correlation matrix is constructed by combining two query correlation matrices, which is used for query suggestion. This framework gives better performance than pseudo-relevance feedback models ([24–26]) and random walk models.

Kharitonov et al. [27] have focused on contextualization 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 a 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. [28] 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 the relevance of follow-up query, probabilistic utility function is used which relies on the query co-occurrence. To capture the semantic similarity of the suggestions, lexical and result set based characteristics are developed. Gradient Boosted Decision Tree (GBDT) regression is performed to rank the suggestions for input query and remove the irrelevant. This approach shows significant improvement over Mutual Information (MI) method.

Broccolo et al. [29] 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 the most relevant query suggestions.

Zhang et al. [30] 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. [31] 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. This function considers both overlapping removal and preservation of neighbourhood structures. This technique is compared with VPSC, PRISM, Voronoi based and RWordle-C by using Euclidean distance, layout similarity and neighbourhood preservation metrics.

Phan et al. [32] 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 datasets to make documents short, less sparse and more topic oriented. This framework is evaluated for online advertising applications on Web search domain matching/ranking and classification. Precision and recall are used to evaluate hidden topics which are used in the improvement of ranking and matching performance.

He et al. [33] 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. Experiments results show that ordered queries within the same session are highly correlated and should be utilized to understand the user information needs. Coverage and accuracy are used as performance evaluation metrics.

Jiang et al. [34] have presented a query recommendation method based on Query Hashing (QH). QH generates many similar and dissimilar query-pairs as prior knowledge from query sessions. Then QH learns a transformation from the prior knowledge such that after transformation of similar queries tend to have similar hash values. In the recommendation stage, queries that have similar hash values to the given query are ranked and *top K* queries are displayed as the recommendation result. QH model is compared with hashing-based methods, SimHash, Kernelized Locality Sensitive Hashing and Inverted list. This method achieves the best results in terms of efficiency and recommendation performance.

Li et al. [35] 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. The topic distribution is acquired in online query suggestion step for an given input query. The candidate queries and input query similarity is computed by using their corresponding topic distribution. Finally, suggestions are provided by listing candidate queries based on similarity score. Precision and Mean Average Precision (MAP) is used as evaluation metrics.

**Table 6.1** Related work comparison

| Author | Concept | Advantages | Disadvantages |
|---|---|---|---|
| Li et al. [35] | 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. [30] | 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. [11] | 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. [10] | Inferring user search goals with feedback sessions | User search goals can be utilized in query recommendations | Finds personalized search goals |
| Liu et al. [36] | 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 |
| RSR | 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 | |

This approach gives better query suggestions than URL model and comparable results with the term feature model.

Liu et al. [36] have proposed a snippet click model for query recommendation. This model determines the information need of users from search logs. The clicked snippets are used to represent 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 6.1 shows comparison of related works.

## 6.3 Related Search Recommendation Framework and RSR Algorithm

### 6.3.1 Problem Definition

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.

### 6.3.2 Co-occurrence Measures to Compute Semantic Similarity

Co-occurrence measures Dice, Jaccard, Pointwise Mutual Information (PMI) and Overlap are explained to calculate semantic similarity. The notation *P(Q)* is used to represent the page counts for the query Q in the search engine. The *WebJaccard* between terms $T_1$ and $T_2$, (i) *WebJaccard*($T_1$, $T_2$) is defined as

$$WebJaccard(T_1, T_2) = \frac{P(T_1 \cap T_2)}{P(T_1) + P(T_2) - P(T_1 \cap T_2)} \tag{6.1}$$

Here, $P(T_1 \cap T_2)$ denotes the co-occurrence of terms $T_1$ and $T_2$.
(ii) *WebDice*($T_1$, $T_2$)*WebDice* is defined as

$$WebDice(T_1, T_2) = \frac{2P(T_1 \cap T_2)}{P(T_1) + P(T_2)} \tag{6.2}$$

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

$$WebOverlap(T_1, T_2) = \frac{P(T_1 \cap T_2)}{min(P(T_1), P(T_2))} \tag{6.3}$$

Pointwise Mutual Information (PMI) is a measure of association used in statistics and information theory. It reflects the dependencies of two probabilistic events. (iv) *WebPMI* is defined as a modification of pointwise mutual information using page counts as

$$WebPMI(T_1, T_2) = \log_2 \left( \frac{\frac{P(T_1 \cap T_2)}{N}}{\frac{P(T_1)}{N} \frac{P(T_2)}{N}} \right) \tag{6.4}$$

### 6.3.3 WordNet-Based Semantic Similarity

WordNet based measures are discussed to calculate semantic similarity. WordNet [38] developed by Princeton University is a lexical database in English. It is well suited for similarity measures, since it organizes verbs, nouns, adjectives and adverbs with variation in semantic relations into synonym sets (synsets) by representing one concept. It uses *is-a* relation to organize 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 a 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 are also supported by WordNet which are $lesk$ (Banerjee and Pedersen 2003), $hso$ (Hirst and St-onge 1998) and $vector$ (Patwardhan 2003).

### 6.3.4 Rocchio's Model

Rocchio's model [37] and Snippet Click model [36] are compared with RSR algorithm. 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 Eq. 6.5. 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.

$$q_e = aq + \frac{b}{\mid D_r \mid} \sum_{d_r \in D_r} d_r - \frac{c}{\mid D_{ir} \mid} \sum_{d_{ir} \in D_{ir}} d_{ir} \tag{6.5}$$

### 6.3.5 Snippet Click Model

Global-scale snippet click model [36] 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.

Related search recommendation framework is presented as shown in Fig. 6.1. 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 a particular user search knowledge and clicked URLs for Web search [39]. Lu et al. [10] have focused on deriving a feedback session with a single query. In this chapter, query suggestions are generated for a query and hence a single session with a 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 6.2 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 irrelevant 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.

*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 [10] 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 6.1.

**Function 6.1:** 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 removal of stopwords, transforming all letters to lowercase and applying Stemming
- Generate $S_p$ from $S$ after removal of stopwords, 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 Eqs. 6.6 and 6.7 respectively
- Generate Enriched Document $ED$ by the weighted sum of $T_v$ and $S_v$ as shown in Eq. 6.8

**end**

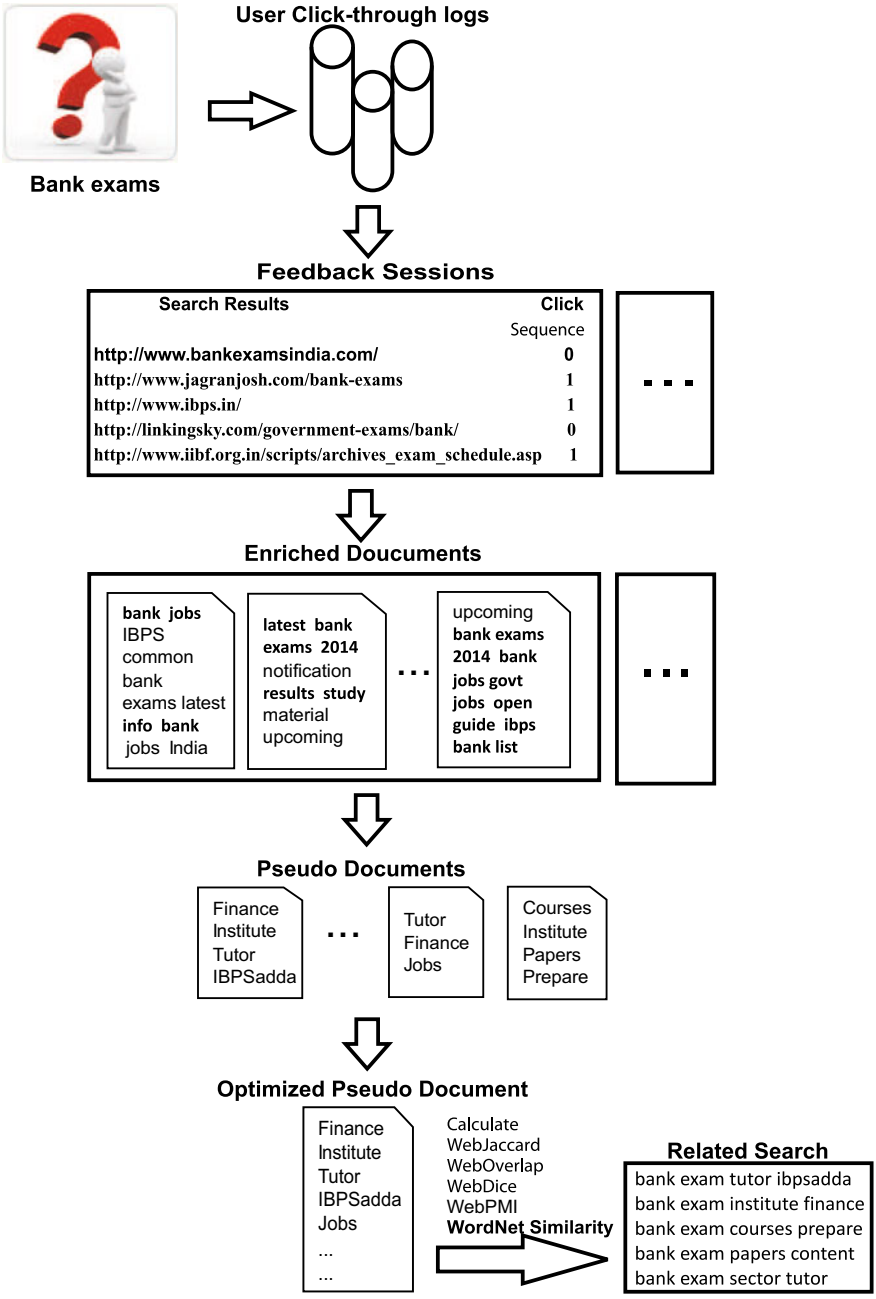


**Fig. 6.1** Related search recommendation framework

| 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-formulae | 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 |

**Fig. 6.2** An example of feedback session for query *bank exam* in rectangular box

$T_v$ and $S_v$ vectors are given in Eqs. 6.6 and 6.7.

$$T_v = [t_{w1}, t_{w2}, ...t_{wm}] \tag{6.6}$$

$$S_v = [t_{w1}, t_{w2}, ...t_{wn}], \tag{6.7}$$

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 Eq. 6.8.

$$ED = w_t T_v + w_s T_s = [ed_{w1}, ed_{w2}....ed_{wk}] \tag{6.8}$$

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 the 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 Fig. 6.1).

*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 6.2.

**Function 6.2:** Pseudo Document

**Function**: PseudoDocument(FeedBack Session $FS$, Enriched Document $ED$)

**for** *each Feedback Session* $FS$ **do**
  Group Enriched Documents of $FS$, as $ED_{clk} = [ed_{w1clk}, ed_{w2clk}, ....ed_{wmclk}]$ and $ED_{unclk}$=$[ed_{w1unclk}, ed_{w2unclk}, ....ed_{wnunclk}]$ of the clicked and un-clicked URLs respectively.
  **for** *each term in* $(ED_{clk}\ U\ ED_{unclk})$ **do**
    Generate Pseudo Document $PD$ by optimizing the value of term such that term $t$ belongs to $ED_{clk}$ gets more importance than $t$ that belongs to $ED_{unclk}$ as given in Eq. 6.9.
  **end**
**end**

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

$$ed_w = \underset{ed_w}{\text{argmin}}\{\sum_{M}[ed_w - ed_{wclk}]^2 - \lambda \sum_{N}[ed_w - ed_{wunclk}]^2\} \tag{6.9}$$

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 $\lambda$ is a parameter balancing the importance of clicked and un-clicked URLs. $\lambda$ is set to 0.5 because if $\lambda$ is set to a small value, then un-clicked URLs importance is reduced and if $\lambda$ 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 Fig. 6.1.

*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 6.3. $N$ is set to 10 as we observe that the top 10 terms are representing the users' information need.

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

### 6.3.6 RSR Algorithm

In this section, we present Related Search Recommendation (RSR) Algorithm as shown in Algorithm 6.1

**Function 6.3:** Optimized Pseudo Document

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

**Algorithm 6.1:** 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 $FS = (fs_1, fs_2...fs_n)$ from user click through log $l$
    **for** *each feedback session* $fs$ *in* $FS$ **do**
      Generate enriched documents $ED = (ed_1, ed_2...ed_m)$ 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 $OPD = (opd_{w1}, opd_{w2},....opd_{wn})$ with *OptPseudoDocument(Pseudocumets PD)*
  **for** *each* $opd_{wi}$ *in* $OPD$ *of size n* **do**
    Calculate semantic similarity of $opd_{wi}$ (1<i<n) AND $opd_{wj}$ (i<j<n) with WebOverlap as discussed in section
    $rq_{Overlapi} = q + opd_{wi} + opd_{wj}$
  $rq = rq_{Overlapi}$

## 6.4 Experiments

### *6.4.1 Data Collection*

To evaluate our proposed method, 95 students participated and each student is assigned 5 queries to collect the feedback session (Permission is taken from the Chairperson, Department of Computer Science and Engineering, UVCE, Bangalore). 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

**Table 6.2** 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 |

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 6.2 shows the statistics of the clicked information of users for this experiment.

### *6.4.2 Experimental 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. Each URL in the feedback session is enriched with title and snippet terms after removing stopwords and applying stemming. Terms are weighed using Term Frequency–Inverse Document Frequency (TF-IDF) as explained in Function 6.1. Enriched documents of a feedback session are classified into clicked and unclicked documents. Pseudo documents are generated by Eq. 6.9. 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 6.3. Optimized Pseudo document has top-10 terms which reflect the user's information need. Semantic similarity between these terms $t_s$ are calculated by WebJaccard, WebDice, WebPMI, WebOverlap methods and WordNet-based similarity measures. Recommendations are generated and ranked by combining query and terms $t_s$ for all the methods.

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. Stopwords removal and stemming are applied for these documents to reduce noise. Expanded queries are generated by Eq. 6.5.

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 the 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 $\lambda$ value to zero in Eq. 6.9 to remove the effect of un-clicked URLs. Optimized Pseudo document is generated by combining all the pseudo documents as shown in Function 6.3. Optimized Pseudo document has top-10 terms. Semantic similarity between these terms $t_s$ are calculated by WebJaccard, WebDice, WebPMI and WebOverlap methods. Recommendations are generated and ranked by combining query and terms $t_s$ for all the methods.

### 6.4.3 Query Recommendation Results

Top-5 recommendation results of Rocchio's model, Snippet Click model, Click-RSR and our RSR algorithm is shown in Table 6.3. 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 the RSR algorithm are *bank exam tutor ibpsadda*, *bank exam institute finance*, *bank exam courses prepare*, *bank exam papers content*, *bank exam sector tutor*.

### 6.4.4 Performance Analysis

From the result shown in Table 6.3, it is observed that RSR algorithm recommends related queries to the given input query. Hundred test queries from various topics like Science, Shopping, Health care have been included.

Lu et al. [10] have discovered different users search goals for a query by using feedback session. This search goals can be utilized in query recommendations. Feedback

**Table 6.3** Related search recommendation results comparison

| Sr. no | Query | Rocchio's model [37] | Snippet click model [36] | Click-RSR | Proposed RSR algorithm |
|---|---|---|---|---|---|
| 1 | Bank exam | Finance | Bank | Question bank | Tutor ibpsadda |
| | | Institute | Competitive | Question tutor | Institute finance |
| | | Tutor | Exam | Papers bank | Courses prepare |
| | | Ibpsadda | Notifications | Shortcuts bank | Papers content |
| | | Gr8ambitionz | Awareness | Bank facebook | Sector tutor |
| 2 | Apartment | Budapest | Budapest | Budapest adina | Realestate properties |
| | | Zillow | Apartment | Zillow rental | Realestate commonfloor |
| | | Decor | Zillow | Zillow genuine | Realestate luxury |
| | | Adina | 123844 | Rental genuine | Properties commonfloor |
| | | Genuine | Luxury | Realestate properties | Properties luxury |
| 3 | Weather | Wiz | Forecast | History weather | BBC forecasts |
| | | Kids | Weather | Wiz kids | BBC animated |
| | | Welcome | Web | Wiz weather | Oceanic atmospheric |
| | | Internet | Local | Web welcome | Forecasts australia |
| | | Temperatures | Dallas | Web weather | Forecasts temperatures |
| 4 | Camera | Nokia | Sony | Nokia grip | Analog lense |
| | | Android | Lines | Nokia 1020 | Analog flash |
| | | Pocket | Cameras | Nokia lumia | CCTV lense |
| | | Grip | Nokia | Grip 1020 | CCTV flash |
| | | 1020 | Github | Grip lumia | Canon lense |
| 5 | Online recharge | Tariffs | MTNL | Payments cellone | Landline cellone |
| | | Cellone | Prepaid | Recharge tariffs | State personal |
| | | Personal | BSNL | Portal cellone | Landline postpaid |
| | | State | Reliance | Prepaid tariffs | Landline huch |
| | | Landline | Services | Banking personal | Landline packs |
| 6 | Free music | Jamendo | Music | Songza worthy | Downloads jango |
| | | Songza | Appears | Composition notation | Downloads limewire |
| | | Composition | Automated | Composition musescore | Beats freeplay |
| | | Archive | Purple | Notation musescore | Beats uncopyrighted |
| | | Jango | Listen | Streaming archive | Beats song |
| 7 | Scholar ships | Reimbursement | Pradesh | Reimbursement fee | Ph.D 2015 |
| | | Federal | IEEE | Federal applying | Ph.D postdoctoral |
| | | Scholarshipporta | Scholarships | Federal finding | Ph.D masters |
| | | Scholarshipexperts | Fellowships | Scholarshipportal solution | 2015 postdoctoral |
| | | Solution | Scholarship | Applying finding | 2015 masters |

(continued)

**Table 6.3** (continued)

| Sr. no | Query | Rocchio's model [37] | Snippet click model [36] | Click-RSR | Proposed RSR algorithm |
|---|---|---|---|---|---|
| 8 | Solar system | Youtube | Tour | Tour solar | Asteroids image |
| | | Wikipedia | Ice | Youtube solar | Kidsastronomy image |
| | | Meteorites | BBC | Youtube witness | Meteorites image |
| | | Characteristics | Phet | Youtube peaceful | Views image |
| | | Astronomy | Velocity | Youtube tues | Visualizer image |
| 9 | Maths | Mathworld | Alpha | Mathworld Webs | Skills watch |
| | | Ask | Wolfram | Mathworld wolfram | American homepage |
| | | Webs | Puzzles | Webs wolfram | Youtube trick |
| | | Level | Guardian | Ask forum | Youtube fast |
| | | Extensive | Drexel | Warwick mathworld | Trick fast |
| 10 | Wedding | Facebook | ANN | Fairy tale | Blog cards |
| | | Fairy | Pretty | Fairy disneys | Fairy tale |
| | | Tale | Wedding | Tale disneys | Registries mywedding |
| | | Disneys | Registry | Gifts fairytale | Blog etiquette |
| | | Registries | Nordstrom | Nordstrom wedfolio | Blog popular |

sessions are utilized in this work and the performance of RSR algorithm is compared with different recommendation methods like classical Rochhio's model [37], Snippet Click Model [36] and modified approach of RSR algorithm considering only clicked URLs. We have adopted Click Through Rate (CTR) method used in [36] 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 WebJaccard, WebDice, WebPMI and WebOverlap methods for RSR algorithm. The average value of CTR and ranked recommendations results are depicted in Fig. 6.3 for all the methods. The average CTR value of Top-5 recommendations are displayed in Table 6.4. CTR is also calculated for WordNet different semantic similarity measures. The average CTR value of Top-5 recommendations are displayed in Table 6.5. 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 Tables 6.4 and 6.5 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 WebJaccard, WebDice, WebPMI and WebOverlap methods for click-RSR algorithm.

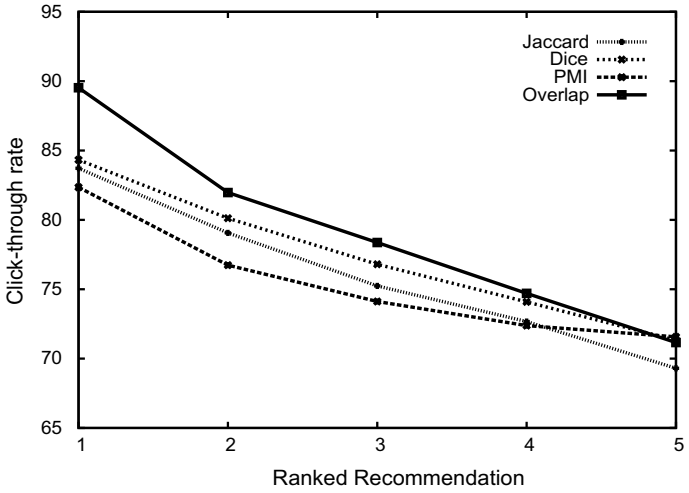


**Fig. 6.3** CTR versus ranked recommendation results

**Table 6.4** 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 6.5** 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 |

The average CTR value of Top-5 recommendations are displayed in Table 6.4. 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 the RSR algorithm with other models, the average CTR value and ranked recommendations are displayed in Fig. 6.4. The average CTR value of Top-5 recommendations for all the models are depicted in Table 6.6. It is observed that the 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 the 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.

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

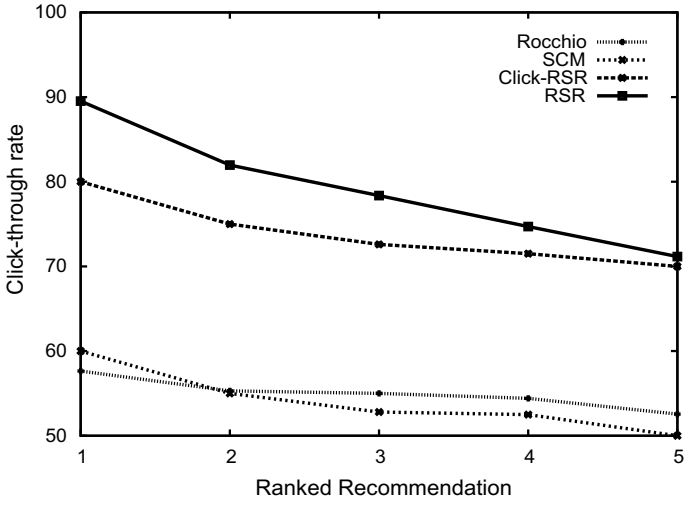


**Fig. 6.4** CTR comparison with other models

**Table 6.6** Average CTR value for Top-5 recommendations for all models

| SCM [36] | Rocchio [37] | Click-RSR | RSR |
|---|---|---|---|
| 54.06 | 55.03 | 73.82 | 79.15 |

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 the 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, the RSR algorithm outperforms the click-RSR.

## 6.5 Summary

In this chapter, 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 WebJaccard, 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 Click-RSR by providing higher CTR value. Further, this work can be extended to classify the search results into different topics.

## References

1. M.P. Kato, S. Tetsuya, T. Katsumi, When do people use query suggestion? A query suggestion log analysis. Journal of Information Retrieval, 16(6), 725–746 December (2013)
2. S. Craig, M. Hannes, H. Monika, M. Michael, Analysis of a very large web search engine query log, in *SIGIR Forum* (1999), pp. 6–12
3. C. Huanhuan, J. Daxin, P. Jian, H. Qi, L. Zhen, C. Enhong, L. Hang, 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* (2008), pp 875–883
4. M. Qiaozhu, Z. Dengyong, C. Kenneth, Query suggestion using hitting time, in *The Proceedings of 17th ACM Conference on Information and Knowledge Management, CIKM '08* (2008), pp. 469–477
5. M. Hao, M.R. Lyu, I. King, Diversifying query suggestion results, in *The Proceedings of 24th AAAI International Conference on Artificial Intelligence, AAAI '10* (2010), pp. 1399–1404
6. J. Guo, X. Cheng, G. Xu, H. Shen, A structured approach to query recommendation with social annotation data, in *The Proceedings of 19th ACM International Conference on Information and Knowledge Management, CKIM '10* (2010), pp. 619–628
7. Y. Song, D. Zhou, L.-W. He, Post ranking query suggestion by diversifying search results, in *The Proceedings of 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '11* (2011), pp. 815–824
8. Z. Kunpeng, W. Xiaolong, L. Yuanchao, A new query expansion method based on query logs mining. Int. J. Asian Lang. Process. **19**(1), 1–12 (2009)
9. Y. Chen, Y.-Q. Zhang, A personalized query suggestion agent based on query-concept bipartite graphs and concept relation trees. Int. J. Adv. Intell. Parad. **1**(4), 398–417 (2009)
10. Z. Lu, H. Zha, X. Yang, W. Lin, Z. Zheng, A new algorithm for inferring user search goals with feedback sessions. IEEE Trans. Knowl. Data Eng. **25**(3), 502–513 (2013)
11. J. Miao, J.X. Huang, Z. Ye, Proximity based Rocchios model for pseudo relevance feedback, in *The Proceedings of 35th ACM International Conference on Research and Development in Information Retrieval, SIGIR' 12* (2012), pp. 535–544
12. M.S. Hamani, R. Maamri, Word semantic similarity based on document's title, in *The Proceedings of 24th IEEE International Workshop on Database and Expert Systems Applications* (2013), pp. 43–47

13. D. Bollegala, Y. Matsuo, M. Ishizuka, Measuring semantic similarity between words using web search engines, in *The Proceedings of 16th International Conference on World Wide Web, WWW '07* (2007), pp. 757–766
14. P. Li, H. Wang, K.Q. Zhu, Z. Wang, X. Wu, Computing term similarity by large probabilistic *isA* knowledge, in *The Proceedings of 22nd International Conference on Information and Knowledge Management, CIKM '13* (2013), pp. 1401–1413
15. B. Danushka, M. Yutaka, I. Mitsuru, A relational model of semantic similarity between words using automatically extracted lexical pattern clusters from the web, in *The Proceedings of International Conference on Empirical Methods in Natural Language Processing, EMNLP '09* (2009), pp. 803–812
16. B. Danushka, M. Yutaka, I. Mitsuru, A web search engine-based approach to measure semantic similarity between words. IEEE Trans. Knowl. Data Eng. **23**(7), 977–990 (2011)
17. P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, in *The Proceedings of the 14th International Joint Conference on Artificial Intelligence* (1995), pp. 448–453
18. D. Lin, An information theoretic definition of similarity, in *The Proceedings of the Fifteenth International Conference on Machine Learning* (1998), pp. 296–304
19. C. Leacock, M. Chodorow, Combining local context and wordnet similarity for word sense disambiguation, in *WordNet: An Electronics Lexical Database* (1998), pp. 265–283
20. J.J. Jiang, D.W. Conrath, Semantic similarity based on corpus statistics and lexical taxonomy, in *The Proceedings of International Conference on Research in Computational Linguistics* (1997), pp. 19–33
21. E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, 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* (2009), pp. 19–27
22. G. Hirst, D. St-Onge, Lexical chains as representations of context for the detection and correction of malapropisms, in *WordNet: An Electronics Lexical Database* (1998), pp. 305–332
23. Y. Song, L.-W. He, Optimal rare query suggestion with implicit user feedback, in *The Proceedings of 19th International Conference on World Wide Web, WWW '10*, pp. 901–910 (2010)
24. N. Craswell, M. Szummer, Random walks on the click graph, in *The Proceedings of 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '07* (2007), pp. 239–246
25. F. Radlinski, T. Joachims, Query chains: learning to rank from implicit feedback, in *The Proceedings of 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD '05* (2005), pp. 239–248
26. G.-R. Xue, H.-J. Zeng, Z. Chen, Y. Yu, W.-Y. Ma, W. Xi, W. Fan, Optimizing web search using web click through data, in *The Proceedings of 13th ACM International Conference on Information and Knowledge Management, CIKM '04* (2004), pp. 118–126
27. E. Kharitonov, C. Macdonald, P. Serdyukov, L. Ounic, Intent estions, in *The Proceedings of 22nd ACM Conference on Information and Knowledge Management, CIKM '13* (2013), pp. 2303–2308
28. U. Ozertem, O. Chapelle, P. Donmez, E. Velipasaoglu, Learning to suggest: a machine learning framework for ranking query suggestions, in *The Proceeding of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12* (2012), pp. 25–34
29. D. Broccolo, L. Marcon, F.M. Nardini, R. Perego, S. Fabrizio, Generating suggestions for queries in the long tail with an inverted index. Int. J. Inf. Process. Manag. **48**(2) (2012)
30. X. Zhang, S. Zilles, R.C. Holte, Improved query suggestion by query search, in *The Proceedings of the 35th Annual German Conference on Artificial Intelligence* (2012), pp. 205–216
31. E. Gomez-Nieto, R.F. San, P. Pagliosa, W. Casaca, E.S. Helou, M.C.F. de Oliveira, L.G. Nonato, Similarity preserving snippet-based visualization of web search results. IEEE Trans. Vis. Comput. Graph. **20**(3), 457–470 (2014)

32. X.-H. Phan, C.-T. Nguyen, D.-T. Le, L.-M. Nguyen, S. Horiguchi, Q.-T. Ha, A hidden topic-based framework toward building applications with short web documents. IEEE Trans. Knowl. Data Eng. **23**(7), 961–976 (2011)
33. Q. He, D. Jiang, Z. Liao, S.C.H. Hoi, K. Chang, E.-P. Lim, H. Li, Web query recommendation *via* Sequential Query Prediction, in *The Proceedings of IEEE 25th International Conference on Data Engineering* (2009), pp. 1443–1454
34. Q. Jiang, M. Sun, Fast query recommendation by search, in *AAAI Conference on Artificial Intelligence* (2011), pp. 1192–1197
35. L. Li, G. Xu, Z. Yang, P. Dolog, Y. Zhang, M. Kitsuregawa, An efficient approach to suggesting topically related web queries using hidden topic model. Int. J. World Wide Web **16**(3), 273–297 (2013)
36. Y. Liu, J. Miao, M. Zhang, S. Ma, L. Ru, How do users describe their information need: query recommendation based on snippet click model. Int. J. Expert. Syst. Appl. **38**(11), 13874–13856 (2011)
37. J.J. Rocchio, Relevance feedback in information retrieval, in *The SMART Retrieval System: Experiments in Automatic Document*, ed. by G. Salton (1971), pp. 313–323
38. C. Fellbaum, *WordNet: An Electronic Lexical Database. Language, Speech, and Communication*. MIT Press, Cambridge (1998)
39. R. Jones, K.L. Klinkner, Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs, in *ACM Conference on Information and Knowledge Management* (2008), pp. 699–708