More Than Relevance: High Utility Query Recommendation By Mining Users' Search Behaviors

Xiaofei Zhu, Jiafeng Guo, Xueqi Cheng, Yanyan Lan

Institute of Computing Technology, Chinese Academy of Sciences, BeiJing, P.R. China
zhuxiaofei@software.ict.ac.cn, {guojiafeng, cxq, lanyanyan}@ict.ac.cn

ABSTRACT

Query recommendation plays a critical role in helping users' search. Most existing approaches on query recommendation aim to recommend relevant queries. However, the ultimate goal of query recommendation is to assist users to reformulate queries so that they can accomplish their search task successfully and quickly. Only considering relevance in query recommendation is apparently not directly toward this goal. In this paper, we argue that it is more important to directly recommend queries with high utility, i.e., queries that can better satisfy users' information needs. For this purpose,we propose a novel generative model, referred to as *Query Utility Model* (QUM), to capture query utility by simultaneously modeling users' reformulation and click behaviors. The experimental results on a publicly released query log show that, our approach is more effective in helping users find relevant search results and thus satisfying their information needs.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithm, Experimentation, Performance, Theory

Keywords

Query Recommendation, Utility, Generative Model, Query Logs

1. INTRODUCTION

Nowadays, most users leverage search engines as an important tool to accomplish various information seeking tasks, e.g., finding particular Web pages, locating target resources, or accessing information of certain topics. However, it is often difficult for users to formulate a query that can describe their information needs properly, and find the right results quickly. Therefore, how to assist users to better formulate queries becomes a valuable and challenging problem for modern search engines.

Query recommendation has been widely recognized as a key technique to alleviate users' reformulation burden and improve the usability of search engines. A major approach on query recommendation is relevant query recommendation, which focuses on providing alternative queries similar to a user's initial query. However, the ultimate goal of query recommendation is to assist users to reformulate queries so that they can acquire their desired information successfully and quickly. Only recommending similar queries, as in relevant query recommendation approaches, is apparently not directly toward this goal.

Therefore, in this paper, we argue that it is more important and beneficial to directly recommend queries with high utility, i.e., queries that can better satisfy users' information needs. Formally, query utility is defined as *the information gain that a user can obtain from the search results of the query according to her original search intent.* By recommending high utility query, we actually emphasize users' *post-click satisfaction*, i.e., whether users will be satisfied by the recommendation after clicking it. We argue that users' post-click satisfaction reflects the true effectiveness of query recommendation.

The central challenge in our recommendation problem is how to learn the query utility according to users' original information needs. The basic idea is that users' collective search behaviors, especially their reformulation and click behaviors in search sessions, embed rich information on the usefulness of queries. Therefore, we propose a novel dynamic Bayesian network, referred to as *Query Utility Model* (QUM), to capture query utility by simultaneously modeling users' reformulation and click behaviors. By learning query utility in hand, we can provide query recommendations with high utility to help users better accomplish their search tasks.

To evaluate the performance of our approach, we compare it with the state-of-the-art query recommendation approaches based on a publicly released query log. We propose two evaluation metrics, namely Query Relevant Ratio (QRR) and Mean Relevant Document (MRD), to evaluate the effectiveness of recommendations with respect to users' post-click satisfaction. The experimental results show that, by recommending high utility queries, our approach is far more effective in helping users find relevant search results and thus satisfy their information needs.

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...$15.00.

2. RELATED WORK

Query recommendation plays a core role for large industrial search engines. Traditional query recommendation approaches usually focus on relevant query recommendation [2, 3, 9, 13, 14, 16, 17, 18]. For example, Wen et al. [16] attempted to find similar queries by clustering queries in query logs based on both query content information and user click-through data. Beeferman et al. [2] applied agglomerative clustering algorithm over the click-through bipartite graph to identify related queries for recommendation. Zhang et al. [18] first proposed to model users' sequential querying behaviors as a query graph and calculate query similarity based on search sessions for recommendations. Boldi et al. [3] further introduced the concept the query-flow graph by aggregating the session information of users, and then performed a random walk on this graph to find relevant queries. Based on the relevance query recommendation, some researchers also proposed to take into account diversity in query recommendation [15, 8, 19]. Recently, some studies consider the utility in query recommendation [11, 1], and formalize query recommendation as a problem of optimizing a global utility function. However, the utility defined in these works is largely different from ours. For example, In [11], the utility actually refers to the diversity of the recommendations. While in [1], the query utility is defined as the possibility that users will be satisfied by its search results, and simply calculated by its click-through rate.

Our model is inspired by the click model [4, 5, 6, 7] in Web search ranking, and the most related work to ours is the DBN model proposed in [4]. Our approach differs from DBN in two significant ways: (1) The problem studied and the data set are completely different. Their goal is to estimate the relevance of URLs for Web search ranking based on the search results and clicks given a query, while we consider the problem of estimating the utility of queries for query recommendation based on the reformulations and clicks given an initial query. (2) The model is also different. In DBN, the satisfaction event only depends on the current state, while in our case, we assume that it is dependent on all previous states.

3. OUR APPROACH

In our work, we propose to recommend high utility queries, i.e., queries that can better satisfy users' information needs, to users. The major problem is how to learn the query utility according to users' original information needs. In this section, we first take a look at a typical users' search session to show what reveals query utility, and give the definition of the query utility. We then introduce a novel dynamic Bayesian network, referred to as Query Utility Model (QUM), to capture query utility by simultaneously modeling users' reformulation and click behaviors. Finally, we show how to estimate the parameters in our model and infer the query utility.

3.1 Search Session and Query Utility

Here we investigate search sessions to see what makes a query useful for users according to their original information needs. A search session, or session, is defined as the sequence of submitted queries and the corresponding clicked URLs within a time limit for a particular information needs [3]. A typical search session is illustrated in Figure 1. Given some information needs, a user submits an initial query to the search engine, and obtains some returned results. Here the documents[1] which are relevant to the user's information needs are shown in red, otherwise in black. The user then goes through the search results, and clicks the second and third results for further inspecting (i.e., denoted by ticks before the results) as they seem to be relevant. The user obtains some useful information from the third result but is not satisfied yet. Therefore, the user reformulates the second query and submits it to the search engine. However, this time she finds no results are related and thus no clicks are performed on the results. Therefore, the third query is generated by the user. From the returned results, she finds two possibly relevant results, i.e., the first and third results, and clicks them for further inspecting. The user then accumulates enough information she needs and finishes the search process.

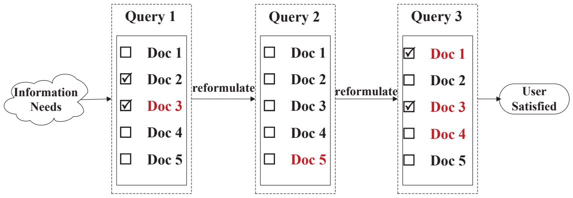


Figure 1: A typical user search session. Tick '$\surd$' denotes a user click on a document, and red color denotes the relevance of a document.

From the above search session, we can see that the first and third queries are useful for the user according to her original information needs. It shows that the usefulness of a query should satisfy two conditions: 1) The query's search results should be *attractive* to the user, so that she would like to click the results for further inspecting, such as the first and third queries. If the search results do not attract the user to click, as in the second query case, the query will not be useful for the user even it contains relevant search results. 2) The clicked search results can actually *satisfy* the user to some extent by providing relevant information to her original needs, e.g., the first and third search results under the third query. Otherwise, the user cannot obtain any useful information after clicking, e.g., the second search result of the first query.

Based on the above analysis, we further divide the utility of a query into two components. One is related to the attractiveness of the query's search results before the user actually inspects the contents of the corresponding URLs, referred to as *perceived utility*, which models the probability that users click the search results of a query given their original needs. The other is related to the satisfaction from the clicked search results, referred to as *posterior utility*, which captures the actual information gain users obtain from the clicked search results given their original needs. The query utility is then defined as the product of the two components, which is the expected information gain users obtain from the search results of the query according to their original information needs.

The remaining problem is how to learn the query utility automatically. From the above search session, we can see that the problem would be quite simple if we have observed

[1] We use the terms URL and document interchangeably throughout this paper.

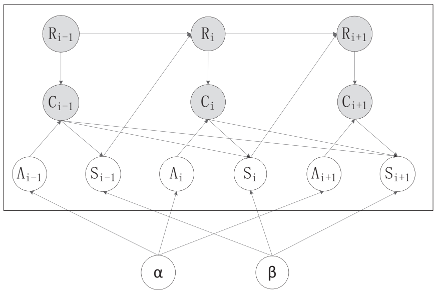


Figure 2: Graphical model representation of Query Utility Model. Observed click and reformulation variables are shaded.

all the search behaviors from users and the exact relevance labels of the search results. However, the true relevance labels are not observed in real search logs, which makes such a learning problem non-trivial. Without loss of generality, here we assume that a user will be more likely to click the search results of a query if she deems the results relevant to their information needs. We also assume that the user will acquire some useful information if the clicked results are relevant to their information needs, and will choose to reformulate the next query if she has not been satisfied. Based on these assumptions, we are able to infer queries' utility from the collective search sessions automatically.

3.2 Query Utility Model

Based on the above analysis, we now introduce a novel dynamic Bayesian network, referred to as Query Utility Model (QUM), to learn query utility based on users' search behaviors. Suppose there are N search sessions starting from the same information needs , and there are a total of T distinct reformulated queries occurring in these N search sessions, denoted by $q_t (1 \leq t \leq T)$.

For a particular search session, we define four binary random variables, R_i, C_i, A_i and S_i to model reformulation, click, attractiveness and satisfaction events at the i-th reformulation position:

- R_i: whether there is a reformulation at position i;
- C_i: whether the user clicks on some of the search results of the reformulation at position i;
- A_i: whether the user is attracted by the search results of the reformulation at position i;
- S_i: whether the user's information needs have been satisfied at position i;

where the first two events are observable from search sessions and the last two events are hidden.

Figure 2 plots the graphical model of QUM for a particular search session. The full model specification that accompanies Figure 2 is as follows:

$$P(C_i = 1 | R_i = 1, A_i = 1) = 1, \tag{1}$$

$$P(A_i = 1) = \alpha_{\phi(i)}, \tag{2}$$

$$P(S_i = 1 | C_{1:i}) = \sigma(\sum_{k=1}^{i} \beta_{\phi(k)} \cdot I(C_k = 1))), \tag{3}$$

$$P(R_i = 1 | R_{i-1} = 1, S_{i-1} = 1) = 0. \tag{4}$$

The above equations describe our model in the following way: The user will reformulate queries sequentially, and stop reformulation if her information needs have been satisfied. There are some clicks on the search results of a reformulated query if and only if the user reformulates the query and is attracted by the search results (1). The probability of the attractiveness only depends on the search results of the reformulated query (2), which is controlled by the variable $\alpha_{\phi(i)}$, i.e., the perceived utility of the query at position i. Note here $\phi(i)$ denotes the index of the query at the position i in a search session. After the user clicks and inspects the search results, there is certain probability she will be satisfied at the current position. The probability of users' satisfaction at position i depends on how much utilities she has accumulated[2] from the previously clicked search results (3). Note here $C_{1:i}$ denote the vector $(C_1, C_2, \cdots, C_i)$, $\beta_{\phi(i)}$ denotes the posterior utility of the query at position i, $I(C_k = 1)$ is an indicator function, and $\sigma(x)$ is the logistic function, i.e. $\sigma(x) = \frac{1}{1+e^{-x}}$. Once the user is satisfied, she will not reformulate the next query (4).

In our model, there are two parameters α_t and β_t $(1 \leq t \leq T)$ related to the utility of a query. The first one models the perceived utility, as it reflects the attractiveness of a query's search results (i.e., the probability that users click the search results of the query given their original needs). The second one models the posterior utility, since it captures the satisfaction of a user from the clicked search results (i.e., the information gain users obtain from the clicked search results given their original needs). As aforementioned, we define the utility of a query as the expected information gain users obtained from the search results of the query according to their original information needs, which is the product of the two facts: $u_t = \alpha_t * \beta_t$.

So far as we know, this is the first dynamic Bayesian network which attempts to model the query utility from users' sequential search behaviors. To simplify our query utility model, in this work we only consider whether the search results of a reformulated query have some clicks or not, but do not specify which URL has been clicked. We may further extend our utility model to capture the specific clicked URLs for finer modeling. Moreover, we take a search session terminated with a user clicking some results as a successful session, otherwise a failure. It is clear that this assumption may not be realistic since users may also stop searching after clicking some results because of impatience. However, even under this simple assumption, we find our model can capture the utility of queries very well as shown in the experimental section. In the future work, we may further improve our model by introducing a persistent parameter to capture the impatience.

3.3 Parameter Estimation

The overall log-likelihood of the N search sessions with the same search intent is:

$$\begin{aligned}\mathcal{L} = \sum_{j=1}^{N} \sum_{i=1}^{N_j} & A_i^j \cdot \log(\alpha_{\phi_j(i)}) + (1 - A_i^j) \cdot \log(1 - \alpha_{\phi_j(i)}) \\ & + S_i^j \cdot \log(\sigma(\sum_{k=1}^{i} \beta_{\phi_j(k)} \cdot I(C_k^j = 1))) \\ & + (1 - S_i^j) \cdot \log(1 - \sigma(\sum_{k=1}^{i} \beta_{\phi_j(k)} \cdot I(C_k^j = 1))),\end{aligned} \tag{5}$$

[2] We assume the utilities of a set of queries can be addictive.

where N_j is the number of reformulations in the j-th session. The click events C_i^j are observed in the j-th search session. If $C_i^j = 1$, we can infer that the results of the reformulation at position i in the j-th session is attractive (i.e., $A_i^j = 1$); Otherwise, $A_i^j = 0$. Besides, based on our assumption, we have that $S_{1:M-1}^j = 0$ and $S_M^j = 1$ if the j-th search session ends with a user clicking some results (i.e., a successful search session); Otherwise, we have that $S_{1:M}^j = 0$.

To obtain an estimate of the parameters $\alpha = \{\alpha_t | 1 \leq t \leq T\}$ and $\beta = \{\beta_t | 1 \leq t \leq T\}$, we can maximize $\mathcal{L}$ with respect to α and β. Due to the space limitation, here we skip the details of the optimization process.

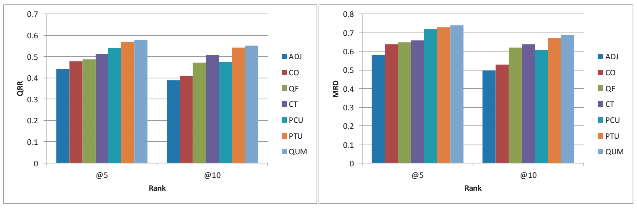
3.4 High Utility Recommendation

Based on the above query utility model, we finally obtain our high utility query recommendation approach. Given the original information needs, typically represented by an input query, we collect all the search sessions starting from the same given query to approximate the search behaviors from that information needs. All the queries in these search sessions other than the input query become the recommendation candidates. We then apply our query utility model and infer the utilities of the recommendation candidates according to the original information needs. Finally, we rank all the candidate recommendations according to their utilities and provide the top ranked ones to users.

4. EXPERIMENTS

In this section, we empirically evaluate the effectiveness of our proposed high utility recommendation method.

Dataset. Our experiments are based on a publicly available dataset, namely UFindIt log data[3], which was collected over a period of 6 months. We process the data by ignoring some interleaved sessions, where the participants search for multiple information needs in one search process. We also remove sessions which have no reformulations, and sessions started without queries. After processing, we obtain 1,298 search sessions, 1,086 distinct queries and 1,555 distinct clicked URLs. For each initial query, the average number of search sessions is 32 and the average number of distinct reformulated queries is 26.



(a) QRR

(b) MRD

Figure 3: Comparison of the performance of all approaches (ADJ,CO,QF,CT,PCU,PTU,QUM) in terms of QRR and MRD.

Evaluation Metrics. In this paper, we resorted to using the manually judgements for evaluation. Specifically, assessors browsed the content of each clicked document of a recommended query and determined whether it can provide useful information, i.e. relevant or irrelevant to the original query. Since this type of information has been kindly labeled in the UFindIt search logs, we will use it for our evaluation, and it is also easier for others to reproduce the results. With the manually labeled data, here we measure the quality of the recommendations with the following two metrics:(1)Query Relevant Ratio (QRR), which and (2) Mean Relevant Document (MRD). For a specific information need, the metric QRR is defined as:

$$QRR(q) = \frac{RQ(q)}{N(q)}, \tag{6}$$

where $RQ(q)$ denotes the total frequency of query q with relevant results obtained(clicked) by users, and $N(q)$ denotes the total frequency of query q issued by users. This metric measures the probability that a user finds relevant results when she uses query q for her search task[4]. A higher QRR means that a user will be more likely to find useful results with respect to the original information needs.

Moreover, for a specific information need, the metric MRD is defined as:

$$MRD(q) = \frac{RD(q)}{N(q)}, \tag{7}$$

where $RD(q)$ denotes the total frequency of relevant results obtained(clicked) by users when they use query q for their search tasks, and $N(q)$ denotes the total frequency of query q issued by users. This metric measures the average number of relevant results a user finds when she uses query q for her search task. A higher MRD means that a user will find more relevant results according to the original information needs.

Baseline Methods. To evaluate the performance of our QUM method, we compare it with four baseline query recommendation methods: (1)Adjacency (ADJ): given a test query q, the top frequent queries in the same session adjacent to q are recommended to users [12]. (2)Co-occurrence (CO): given a test query q, the top frequent queries co-occurred in the same session with q are selected as recommendations [10]. (3)Query-Flow Graph (QF): it constructs a query-flow graph based on collective search sessions, and then a random walk is performed on this graph for query recommendation [3]. (4)Click-through Graph (CT): this method creates a query-URL bipartite graph by mining query logs [15]. It performs a random walk and employs the hitting time as a measure to select queries for recommendation. The former two methods can be regarded as frequency-based methods and the latter two methods can be regarded as graph-based methods. Besides, to further investigate the influence of the two component utilities (i.e., perceived utility and posterior utility) in our QUM method, we also use them separately to recommend queries as two baselines, namely Perceived Utility method (PCU) and Posterior Utility method (PTU).

4.1 Results

Figure 3(a) and Figure 3(b) show the performance of top recommendations from different methods under the metric QRR and MRD, respectively. From Figure 3, we can see that the two frequency-based methods ADJ and CO perform poorly under the two metrics. It shows that by simply considering the most frequently adjacent or co-occurring

[3] http://ir-ub.mathcs.emory.edu/uFindIt/

[4] In our experiment, we use $QRR(q) = (RQ(q)+1)/(N(q)+2)$ to reduce the influence of observing frequency in computing QRR. Similarly for the metric MRD.

queries in the same session with the given query (which are usually highly relevant), we can't guarantee to recommend useful queries to satisfy users' information needs. The two graph-based methods, i.e., QF and CT, show better performance than the frequency-based methods. It indicates that by leveraging the local relationships (i.e., either the co-click or the reformulation relationship) between query pairs to collectively reveal the global relationships between queries, we are able to find better query recommendations.

The PCU method, which only relies on queries' perceived utility for recommendation, presents a comparable performance with the two graph-based methods. As we know, the PCU method actually recommends queries according to the expected click-through rate over their search results (i.e., perceived utility). Since users are more likely to click the search results that they deem relevant, the perceived utility implicitly convey the utility information of the queries. However, only after inspecting the content of the clicked results, users can decide whether the results are truly relevant. Therefore, the queries with high perceived utility are not necessary to be highly useful. That is the reason why the PCU method cannot always show high performance according to the two metrics.

Moreover, the PTU method, which takes queries' posterior utility for recommendation, shows better performance as compared to the above baseline methods under both metrics. It indicates that by simultaneously consider users' reformulation and click behaviors, the learned posterior utility can better reflect the usefulness of the queries. Moreover, when compared with the PCU method, we can find that the PTU method shows constantly better performance. It further demonstrates the importance to take into account users' reformulation behaviors (i.e., satisfaction event) to capture the utility of queries.

Finally, as we can see from Figure 3, our QUM method performs better than all the baseline recommendation methods. We conduct t-test (*p-value* $<= 0.05$) over the results and find that the performance improvements are significant as compared with both the frequency-based and graph-based baselines. It shows that by leveraging our query utility model, we are able to infer and recommend high utility queries for users. Meanwhile, as compared with the two component utility methods (i.e., PCU and PTU methods), the QUM methods can also constantly outperform the two baselines. It demonstrates that to recommend high utility queries, we need to find those queries that can not only attract users to click their search results, but also provide useful information with the clicked search results. The two aspects are complementary and neither can be ignored for high utility recommendation.

5. CONCLUSIONS

This paper proposes a novel dynamic Bayesian network to infer query utility from users' search behaviors for recommendation. We evaluate the performance on a real query log, and the experimental results show that the proposed approach can outperform the state-of-the-art baselines in providing useful recommendations.

6. ACKNOWLEDGMENTS

This research work was funded by the National Natural Science Foundation of China under Grant No. 60933005, No. 61173008, No. 61003166 and 973 Program of China under Grants No. 2012CB316303.

7. REFERENCES

[1] A. Anagnostopoulos, L. Becchetti, C. Castillo, and A. Gionis. An optimization framework for query recommendation. In *WSDM*, pages 161–170, 2010.

[2] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *SIGKDD*, pages 407–416, 2000.

[3] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The query-flow graph: model and applications. In *CIKM*, pages 609–618, 2008.

[4] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *WWW*, pages 1–10, 2009.

[5] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *WSDM*, pages 87–94, 2008.

[6] G. Dupret and C. Liao. A model to estimate intrinsic document relevance from the clickthrough logs of a web search engine. In *WSDM*, pages 181–190, 2010.

[7] L. A. Granka, T. Joachims, and G. Gay. Eye-tracking analysis of user behavior in www search. In *SIGIR*, pages 478–479, 2004.

[8] J. Guo, X. Cheng, G. Xu, and H. Shen. A structured approach to query recommendation with social annotation data. In *CIKM*, pages 619–628, 2010.

[9] J. Guo, X. Cheng, G. Xu, and X. Zhu. Intent-aware query similarity. In *CIKM*, pages 259–268, 2011.

[10] C.-K. Huang, L.-F. Chien, and Y.-J. Oyang. Relevant term suggestion in interactive web search based on contextual information in query session logs. *JASIST*, 54(7):638–649, 2003.

[11] A. Jain, U. Ozertem, and E. Velipasaoglu. Synthesizing high utility suggestions for rare web search queries. In *SIGIR*, pages 805–814, 2011.

[12] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *WWW*, pages 387–396, 2006.

[13] L. Li, Z. Yang, L. Liu, and M. Kitsuregawa. Query-url bipartite based approach to personalized query recommendation. In *AAAI*, pages 1189–1194, 2008.

[14] H. Ma, H. Yang, I. King, and M. R. Lyu. Learning latent semantic relations from clickthrough data for query suggestion. In *CIKM*, pages 709–718, 2008.

[15] Q. Mei, D. Zhou, and K. Church. Query suggestion using hitting time. In *CIKM*, pages 469–477, 2008.

[16] J.-R. Wen, J.-Y. Nie, and H.-J. Zhang. Clustering user queries of a search engine. In *WWW*, pages 162–168, 2001.

[17] X. Yan, J. Guo, and X. Cheng. Context-aware query recommendation by learning high-order relation in query logs. In *CIKM*, pages 2073–2076, 2011.

[18] Z. Zhang and O. Nasraoui. Mining search engine query logs for query recommendation. In *WWW*, pages 1039–1040, 2006.

[19] X. Zhu, J. Guo, X. Cheng, P. Du, and H.-W. Shen. A unified framework for recommending diverse and relevant queries. In *WWW*, pages 37–46, 2011.