Recommending High Utility Query via Session-Flow Graph

Xiaofei Zhu[1,2], Jiafeng Guo[1], Xueqi Cheng[1],
Yanyan Lan[1], and Wolfgang Nejdl[2]

[1] Institute of Computing Technology, Chinese Academy of Sciences, BeiJing, China
zhuxiaofei@software.ict.ac.cn, {guojiafeng,cxq,lanyanyan}@ict.ac.cn
[2] L3S Research Center, Leibniz Universität Hannover, Hannover, Germany
nejdl@L3S.de

Abstract. Query recommendation is an integral part of modern search engines that helps users find their information needs. Traditional query recommendation methods usually focus on recommending users relevant queries, which attempt to find alternative queries with close search intent to the original query. Whereas the ultimate goal of query recommendation is to assist users to accomplish their search task successfully, while not just find relevant queries in spite of they can sometimes return useful search results. To better achieve the ultimate goal of query recommendation, a more reasonable way is to recommend users high utility queries, i.e., queries that can return more useful information. In this paper, we propose a novel utility query recommendation approach based on absorbing random walk on the session-flow graph, which can learn queries' utility by simultaneously modeling both users' reformulation behaviors and click behaviors. Extensively experiments were conducted on real query logs, and the results show that our method significantly outperforms the state-of-the-art methods under the evaluation metric QRR and MRD.

Keywords: Query Recommendation, Absorbing Random Walk, Session-Flow Graph.

1 Introduction

Search engines have become an essential way for satisfying users' daily information needs, such as finding particular Web pages, locating target resources, or accessing information of certain topics. However, formulating a proper query for search is difficult for users. Most of them need to reformulate their queries several times before satisfaction.

To alleviate users' search burden, query recommendation has been proposed and considered as a prominent ingredient of modern search engines. Query recommendation aims to provide users alternative queries, which can represent their information needs more clearly in order to return better search results [8,22]. Previous research on query recommendation focuses on recommending users relevant queries to their initial queries. Different ways for measuring the query relevance

P. Serdyukov et al. (Eds.): ECIR 2013, LNCS 7814, pp. 642–655, 2013.

are employed, for example, common query terms [19], same clicked documents [4,13,15] or occurring in same search sessions [20,5], to calculate the relevance between queries, and then recommend users these most relevant queries. The basic assumption of this type of methods is that more useful search results will be returned if the relevant recommendations are used.

However, the problem of this assumption is that even some very relevant queries may have few or even no useful search results, while other comparatively less relevant queries may produce more useful search results. For example, given a user's initial query "iphone available time market" which tends to find "what's the time of iphone to sell on the market", the candidate recommendations may include "iphone market sale time", "iphone start selling market" and "iphone release date". Obviously, the three recommended queries are all relevant to the user's initial query, especially the former two queries are more relevant in terms of textual similarity, but the search results show that the last one can find better search results for satisfying users' needs. If the we recommend the user the former two queries, she may feel disappointed about their search results. The reason behind this problem is that these methods only take into account the relationship between queries for recommendation according to some similarity metrics, but ignore the utility of queries themselves, i.e. how much useful information can this query return to satisfy users' information needs?

Since 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 relevant queries is apparently not directly toward this goal. Therefore, it is necessary to further recommend users high utility queries, 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 emphasize users' *post-click experience*, i.e., whether users will be satisfied by the recommendation after clicking it.

The central challenge in high utility query recommendation is how to learn the query utility according to users' original information needs. In [23], Zhu et al. proposed a Dynamic Bayesian Network to mining query's utility from users' collective search behaviors. This method has certain fundamental limitations as it cannot make full use of the click-through information. Specifically, it only considers whether the search results of a reformulated query have some clicked documents or not, but does not take individually clicked document into consideration. In this paper, we propose a novel method referred to as Two-phase model based on Absorbing Random Walk (TARW), to further capture these specific clicked documents for modeling query utility. With the learned query utility in hand, we can provide users query recommendations with high utility to help them better accomplish their search task.

The main contributions of this paper are three folds: First, we introduce the session-flow graph to capture both users' click behaviors and reformulation behaviors, and we also import failure nodes into this graph to further capture the behaviors that users give up their search tasks. Second, we proposed a novel model tailed to infer query utility on the session-flow graph. Finally, we conduct an empirical study on publicly released query logs, and the results show that our method performs significantly better than the state-of-the-art methods in recommend high utility queries, thus can better satisfy users' information needs.

The rest of this paper is structured as follows. Section 2 reviews the related work. In Section 3, we introduce the session-flow graph. Our proposed model is described in Section 4. In Section 5, we report on experiments on publicly released query logs. Finally, we conclude and describe future work in Section 6.

2 Related Work

In this section, we review two research topics which are relevant to our work: query recommendation and absorbing random walk.

Query Recommendation. Query recommendation plays a critical role in modern industry search engines. Most of the previous work on query recommendation sheds light on relevant query recommendation, i.e. recommending alternative queries similar to the user's initial query. In these literatures, both click-through logs [4,19,13,15,14] and query session logs [20,5] are the two most commonly used information embedded in the search logs. Wen et al. [19] 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. [4] applied agglomerative clustering algorithm over the click-through bipartite graph to identify related queries for recommendation. Zhang et al. [20] 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. [5] 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. There are also some studies [15,8,22] taking into account recommending diverse queries to satisfy users' multiple search intents.

Recently, several research work [2,10,23] proposed to recommend high utility queries to users. Both studies [2,10] defined a global utility function over the recommendation set, which emphasize either the diversity [10] or the expected click-through rate [2] of the recommendations. They did not define and learn the query utility toward users' post-click experience as ours. The paper most closely related to our work is by Zhu et al [23]. Indeed, their paper employed a generative way to mine query utility from users' search behaviors. This approach differs from ours in that it uses only partial information of users' click behaviors, while leaves other information, such as the individually clicked documents, unconsidered. Our method further takes these information into consideration, thus can better infer query utility by making full use of users' search behaviors.

Absorbing Random Walk. Recently, absorbing random walk has been widely used in many popular research domains, such as expert finding [17], text summarization [21], recommendation system [18], and query refinements clustering [16]. The literatures most closely related to our work are [18,16]. In [18], Singh et al.

performed an absorbing random walk on an augmented bipartite graph, which combines both user-item graph and user-user social links, for helping user find interesting items. In [16], Sadikov et al. aimed to cluster the refinements of an original query into a set of different search intents. They modeled user behavior as a graph, and leverage absorbing random to describe the original query as a probabilistic distribution over the documents.

Our method differs from them mainly in two aspects: (1) our method can automatically set specific transition probability for each query node by considering their performance in the search process, while they leverage a common transition probability setting to different types of nodes. (2) we propose a novel graph referred to as session-flow graph to depict users' search behaviors, and apply a variant absorbing random walk on this graph to make query recommendation.

3 Session-Flow Graph

In [5], Boldi et al. first proposed the concept of query-flow graph, in which a directed edge from query q to query q' represents that the query q' is a reformation of the query q. The query-flow graph can effectively describe users' query reformulation behaviors. However, the limitation of this graph is that it cannot capture other types of users' search behaviors, such as users' click behaviors or users give up their search tasks.

In order to better model users' search behaviors, in this paper, we propose a new concept referred to as the session-flow graph, which expands the traditional query-flow graph by introducing other two types of nodes, i.e. document nodes and failure nodes. The document nodes represent the corresponding clicked documents for each issued query by users, and the failure nodes represent the situation that users' search tasks are unsuccessful[1]. In contrast to the query-flow graph, session-flow graph can simultaneously describe both users' reformulation behaviors and click behaviors. Meanwhile, it can also take into account the situation that users' information needs have not been satisfied. Figure 1 shows an example of the session-flow graph.

The session-flow graph can be defined as a directed graph $G_{sf} = (Q, D, S, E, F, G)$, where:

- $Q = \{q_1, q_2, \cdots, q_n\}$ is the set of distinct queries issued to the search engine;
- $D = \{d_1, d_2, \cdots, d_m\}$ is the set of distinct documents clicked by users after they submit queries to the search engine;
- $S = \{s_1, s_2, \cdots, s_n\}$ is the set of failure nodes, where each failure node corresponds to a query and represents the situation that users' information needs cannot be satisfied by this query.
- $E \triangleq \{(q, q') : q \in Q, q' \in R(q)\}$ is the edges from queries to their reformulations, where $R(q)$ denotes the set of reformulations of a query q;

[1] Here we assume a user search task, i.e. a search session, is successful if the last submitted query has some clicked documents, otherwise unsuccessful.

- $F \triangleq \{(q,d) : q \in Q, d \in C(q)\}$ is the edges from queries to their clicked documents, where $C(q)$ is the set of clicked documents of a query q;
- $G \triangleq \{(q,s) : q \in Q, s \in S\}$ is the edges from queries to their failure nodes.

Let W_E, W_F and W_G be the adjacency matrix corresponding to the edges E, F and G, respectively. Let $w(q,q') \in W_E$ be a weighting function for an edge $(q,q') \in E$, e.g., the number of reformulations that query q' reformulates the query q. Let $w(q,d) \in W_F$ be a weighting function for an edge $(q,d) \in F$, e.g., the number clicks on a document d of a query q. Let $w(q,s) \in W_S$ be a weighting function for an edge $(q,s) \in G$, e.g., the number of times that users' search tasks are unsuccessful when they use the query q. Then we get the corresponding adjacency matrix for the session-flow graph , which can be written as:

$$W = \begin{bmatrix} W_E & W_F & W_G \\ W_F^T & 0 & 0 \\ W_G^T & 0 & 0 \end{bmatrix} \tag{1}$$

4 Proposed Approach

After constructing the session-flow graph, we propose a novel two-phase model based on absorbing random walk. This model consists two steps: (1) forward utility propagation, and (2) backward utility propagation. In the forward utility propagation, we treat the query nodes[2] in the session-flow graph as transient states, and both document and failure nodes as absorbing states. Then an absorbing random walk will be run on the session-flow graph. By this procedure, we can learn the utility of each document. And with the learned document utility in hand, a backward utility propagation procedure will be followed. In this procedure, the utility will be reversely propagated from document nodes to these reformulation nodes and we can then learn the utility of each candidate query.

4.1 Forward Utility Propagation

In the forward utility propagation, a random walker starts from the original query which we want to generate recommendations for, then she either visits reformulation nodes, or be absorbed by document nodes or failure nodes. For each query node, the walker will choose to go to its reformulation nodes with probability α_1, or to its clicked document nodes with probability α_2, or to its failure node with probability α_3 (where $\alpha_1 + \alpha_2 + \alpha_3 = 1$). We can regard α_1 as the importance of reformulation for satisfying users' information needs, a larger α_1 represents the walker believes that the reformulation can better satisfy

[2] Note that the query nodes include both the original query nodes and the reformulation query nodes if there is no specification. We use the termination 'original query' to denote the query for which we generate recommendations, and the termination 'reformulation query' (or 'reformulation' for short) to denote the query which will be used as candidates for recommendation.

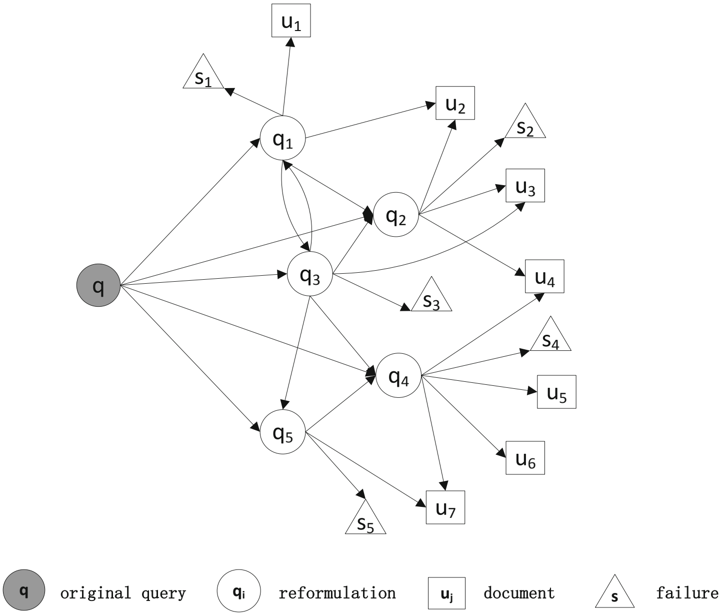

Fig. 1. An Example of the Session-Flow Graph

users' information needs, and she will walk to reformulation nodes with a larger probability. α_2 can be considered as the importance of clicked documents for satisfying users' information needs, a larger α_2 represents the walker believes that the clicked documents can provide more useful information for satisfying users' information needs, and she will walk to document nodes with a larger probability. α_3 depicts the situation that users' information needs has not been satisfied, i.e. users give up their search tasks. A larger α_3 represents the walker believes that her information needs cannot be satisfied, and she will choose to give up the search task, i.e., with a larger probability to the failure node.

In the existing absorbing random walk literatures [16,18], all query nodes share the same transition probability setting to different types of nodes (i.e. reformulation node, document node and failure node[3]). However, just setting common transition probability does not reflect the characteristics of each query. For example, if a query is more likely to be reformulated but with less click operation, then a higher probability α_1 should be assigned to encourage transition to its reformulation nodes. And if the search result of a query is frequently clicked by users but with less reformulation operation, then this type of query should be assigned with higher probability α_2 for transiting its document nodes. Moreover, if a query is difficult and users cannot satisfy their information needs, then it should be assigned with a high probability α_3 for transiting to a failure node ($\alpha_1 + \alpha_2 + \alpha_3 = 1$).

[3] In existing literatures [16,18], they have not considered the failure nodes, which equals to set $\alpha_3 = 0$ in our model.

To this end, we further make use of the observed transition probability β_1^i, β_2^i, β_3^i of query q_i and obtain a posterior transition probability for each query. Formally, the posterior transition probability for query q_i can be written as:

$$\alpha_k^i = (1-\lambda)\alpha_k + \lambda\beta_k^i, k = 1, 2, 3, \tag{2}$$

where $\lambda \in [0,1]$ controls the importance of prior transition probability $\alpha_k(k = 1, 2, 3)$ (all queries share the same prior transition probability) and observed transition probability $\beta_k^i(k = 1, 2, 3)$. If $\lambda = 1$, then the transition probability only depends on observed knowledge. Otherwise, the transition probability only depend on prior knowledge[4] if $\lambda = 0$. Here we treat them equally, thus set $\lambda = 0.5$.

When the walker chooses to transit to a reformulation, the transition probability from query q_i to its reformulation q_j is defined as the fraction of the number of time q_i that was reformulated by q_j over the total times q_i that was reformulated, formally:

$$P(q_j|q_i) = \alpha_1 \times \frac{w(q_i, q_j)}{\sum_{q_k \in R(q_i)} w(q_i, q_k)}, \tag{3}$$

where $w(q_i, q_j)$ denotes the number of times the query q_i was reformulated by query q_j.

When the walker transits from query q_i to its clicked document d_j, the corresponding transition probability is the fraction of the document d_j' click frequency over the total document click frequency of query q_i, formally:

$$P(d_j|q_i) = \alpha_2 \times \frac{w(q_i, d_j)}{\sum_{d_k \in C(q_i)} w(q_i, d_k)}, \tag{4}$$

where $w(q_i, d_j)$ denotes the number of times the document d_j was clicked by users when they issued the query q_i.

When the walker transits from q_i to its failure node s_i, the corresponding transition probability is $P(s_i|q_i) = \alpha_3$. Since both document and failure nodes are absorbing states, the walker can never transit to other nodes if they reach these nodes, thus $P(d_j|d_j) = 1$ and $P(s_i|s_i) = 1$. Besides, we will process queries without any reformulation or clicked documents. Specifically, if a query has no reformulation, we will assign α_1 value uniformly to all reformulation nodes in the graph. Similarly α_2 value will be uniformly assigned to all document nodes in the graph if a query has no clicked documents.

[4] We empirically set $(\alpha_1, \alpha_2, \alpha_3) = (0.95, 0.05, 0)$. Here we let $\alpha_3 = 0$, so our method will regress to the existing absorbing random walk [16,18] if we ignore the effect of the observed transition probaility (i.e., $\lambda = 0$).

4.2 Computing the Distribution

In the forward utility propagation, the corresponding transition matrix can be represented as:

$$P = \begin{bmatrix} P_Q & P_D & P_S \\ 0 & I_D & 0 \\ 0 & 0 & I_S \end{bmatrix}, \tag{5}$$

where P_Q is a $n \times n$ transition matrix on queries, P_D is a $n \times m$ matrix of transition from query to document , P_S is a $n \times n$ matrix of transition from query to failure node, I_D is a $m \times m$ identity matrix and I_S is a $n \times n$ identity matrix.

Since the above transition matrix is reducible, there is no stationary distribution. An alternative way to compute the absorbing distribution is by using an iterative way

$$P^t = \begin{bmatrix} P_Q^t & \sum_{k=0}^{t-1} P_Q^k P_D & \sum_{k=0}^{t-1} P_Q^k P_S \\ 0 & I_D & 0 \\ 0 & 0 & I_S \end{bmatrix}, \tag{6}$$

where $P^t[i,j]$ represents the probability of node i to node j after t step walk. Here we need to compute the probability from query to document, i.e. computing the upper middle matrix $\sum_{k=0}^{t-1} P_Q^k P_D$ of P^t, and the computing complex is $O(tn^3 + n^2m)$. Moreover, in the recommendation scenario, we only need to compute the probability from the original query to its clicked documents, i.e., computing the distribution of the matrix row corresponding to the original query. Let v (a $1 \times m$ row vector) denote the corresponding row of the original query, thus we will compute vP_Q^{k-1} instead of P_Q^k, and the computation complex is $O(tn^2 + nm)$.

4.3 Backward Utility Propagation

With the document utility learned, the next step is reversely propagate the utility from document nodes to the reformulation nodes[5]. Since a query's utility is represented by the documents which users will click after issuing the query, an intuitive way to infer a query's utility is to aggregate the utility of its clicked documents. Although this method is simple, the experimental results show that it is effective and robust. We can also use other ways to calculate query's utility, e.g., run an random walk on the click-through graph [7].

After learning each candidate queries' utility, they are ranked in descending order of the utility, which represent the amount of useful information they can provide for satisfy users' information needs. Candidate queries with the highest utilities will be recommended to users.

[5] The utility absorbed by failure nodes denotes the information which cannot be satisfied, thus it is unnecessary to propagate it back to the reformulation nodes.

5 Experimental Results

To demonstrate the effectiveness of our proposed high utility query recommendation approach, we conducted experiments on publicly available query logs and compared our method with six baselines. Furthermore, we also evaluated the learned document utility in the forward utility propagation of our method.

5.1 Dataset

Our experiments are based on publicly available query logs, namely UFindIt log data [1]. There are totally 40 search tasks represented by 40 test queries. We process the data by ignoring some interleaved sessions, where the participants search for multiple information needs in one search session. 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 documents. For each test query, the average number of search sessions is 32 and the average number of distinct reformulation queries is 26.

5.2 Evaluation of Query Utility

Metrics. We evaluate the effectiveness of different approaches with manual judgements, where all users' clicked search results have been labelled as relevant or irrelevant with respect to their original information needs. Here we use two evaluation metrics proposed in [23], namely the Query Relevant Ratio (QRR) and the Mean Relevant Document (MRD), to measure the performance of the recommendations. For a specific information need, the metric QRR is defined as:

$$QRR(q) = \frac{RQ(q)}{N(q)}, \tag{7}$$

where $RQ(q)$ denotes the total frequency of query q with relevant results 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[6]. A higher QRR means that users will be more likely to find useful results with respect to the original information needs. Besides, for a specific information need, the metric MRD is defined as:

$$MRD(q) = \frac{RD(q)}{N(q)}, \tag{8}$$

where $RD(q)$ denotes the total frequency of relevant results 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 users will find more relevant results in terms of the original information needs.

[6] 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.

Baseline Methods. To evaluate the performance of our TARW method, we compare it with six baseline query recommendation methods: (1) Adjacency (ADJ): given a test query q, the most frequent queries in the same session adjacent to q are recommended to users [12]. (2)Co-occurrence (CO): given a test query q, the most frequent queries co-occurred in the same session with q are selected as recommendations [9]. (3) Query-Flow Graph (QFG): this is a state-of-the-art method. It constructs a query-flow graph based on collective search sessions, and then a random walk is performed on this graph for query recommendation [5]. (4) Click-through Graph (CTG): this is also a state-of-the-art method. It creates a query-URL bipartite graph by mining query logs [15]. Then it performs a random walk and employs the hitting time as a measure to select queries for recommendation. (5) Click-through Rate(CTR): it employs [2] the expected click-through rate of the search results of a query for recommendation. The basic assumption is that a user is more likely to click the search results of a query if she believes this query is relevant, which reflect the query's perceive relevance. (6) Query Utility Model(QUM): this is another state-of-the-art method. It uses a Dynamic Bayesian Network to learn query utility based on users search behaviors [23].

Overall Evaluation Results. Figure 2(a) and Figure 2(b) show the performance of top recommendations from different methods under the metric QRR and MRD, respectively. From Figure 2, 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 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., QFG and CTG, 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 CTR method, which solely relies on the expected click-through rate over their search results. Since users are more likely to click the search results that they deem relevant. However, only after inspecting the content of the clicked results, users can decide whether the results are truly relevant. Therefore, the queries with high click-through rate are not necessary to be highly useful. That is the reason why the CTR method cannot always show high performance according to the two metrics. Moreover, the QUM method shows better performance as compared with the above baseline methods under both metrics. It demonstrates the importance to take into account the posterior effect of each query.

Finally, as we can see from Figure 2, our TARW method performs better than all the baseline recommendation methods. We conduct t-test ($p\text{-}value \leq 0.05$) over the results and find that the performance improvements are significant as compared with all the baseline methods. It shows that by further modelling each clicked document of queries, our method can better learn the utility of queries and thus help users finding their desired informations.

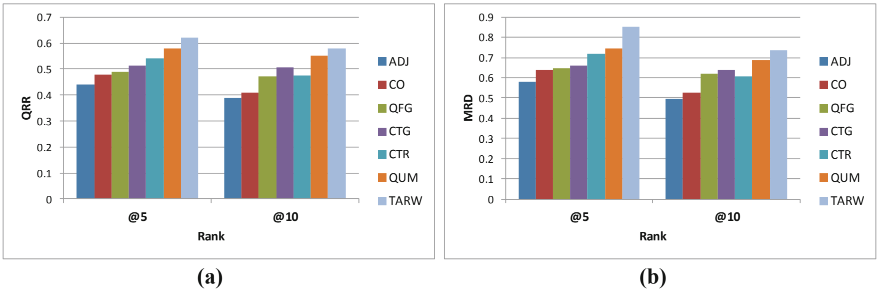

Fig. 2. Comparison of the performance of all approaches (ADJ, CO, QFG, CTG, CTR, QUM, TARW) in terms of (a) Query Relevant Ratio (QRR) and (b) Mean Relevant Document(MRD)

5.3 Evaluation of Document Utility

Since the performance of our two-phase utility model relies on the inferred utility of the documents in the forward utility propagation procedure, in this subsection, we will evaluate the learned documents' utility in our method. Specifically, we compare it against three baseline approaches[7]: (1) Document Frequency Based Method (DF): it is based on the click frequency of a document when users browse the search results of the original query. This method assumes that the click frequency of a document reflects users preference for that document when they search with the original query, thus it can be used to describe the relevance of the document to the original query. (2) Session Document Frequency Based Method (SDF): since document clicks of a query is sparse, where some of relevant documents may have no clicks. To alleviate this problem, another baseline method referred to as SDF is used, which is based on the document click frequency within the same search session (since the separation of sessions is out of the scope of this paper, here we assume an ideal search session separation method exists). SDF assumes that documents within the same search session convey the similar search intent, thus aggregated click frequency can be used to reflect their relevance to the original query. (3) Markov-model Based Method (MM): we employ the method in [16] as a baseline method, which is a state-of-the-art method, and the learned document distribution for the original query are employed for ranking.

Since the utility of a document is the useful information it can provide for satisfy users' information needs, the traditional document labelling strategy, e.g., labelling as relevant or irrelevant , can be employed. There are many metrics to evaluate the performance of methods on documents' utility learning, and here we employ three metrics: Precision at position k(P@k)[3] , Normalized Discounted Cumulative Gain(NDCG) [11], and Mean Average Precision (MAP) [3].

[7] Notice that here we don't employ click models [6] as baselines since these models also rely on knowing the search results without clicks, which is only available within some commercial search engine companies.

Table 1. Comparison of document relevance of four approaches (The percentages in the parentheses are the improvements of our TARW method over the corresponding methods)

Method	P@5	P@10	MAP	NDCG@5	NDCG@10
DF	0.460(32.6%)	0.330(47.7%)	0.468(33.5%)	0.524(27.3%)	0.487(35.2%)
SDF	0.555(9.9%)	0.448(8.9%)	0.568(10.1%)	0.612(8.9%)	0.610(8.0%)
MM	0.590(3.4%)	0.463(5.4%)	0.597(4.7%)	0.648(2.9%)	0.632(4.3%)
TRAW	**0.610**	**0.488**	**0.625**	**0.667**	**0.659**

Table 1 presents the performance of different approaches on learning document utility. Among all methods, DF performs worst than the other methods. This is not surprising given that DF suffers from the problem of click sparsity. SDF outperforms DF since it aggregates the document click information within the same search session to alleviate the sparsity problem. MM not only considers the document click information within a session, but also utilizes its corresponding query's position information in the search session, thus it can better infer document utility. The limitation of MM is that all query nodes share the same transition probability setting to different type of nodes, while never considering their performance in the search process. TARW improvements over all baselines by using an adaptive transition probability setting to different types of nodes, and it also models users' behaviors of giving up their search tasks by introducing the failure nodes.

6 Conclusion

In this paper we investigated the problem of how to recommend high utility queries to users. To this end, we first propose the concept of session-flow graph to capture users search behaviors, including both reformulation behaviors and click behaviors. Then a novel two-phase model based on absorbing random walk is proposed, which is tailed to the session-flow graph, to effectively learn queries' utility from Web users' search behaviors. Experimental results on publicly released query logs show that our proposed approach achieves statistically significant improvements over the baselines.

There still exists some interesting problems needed to be addressed in the future work: 1) Our approach dedicates to mining each query's utility separately. However, when we recommend a set of high utility queries to users, this method will suffer from the redundant utility in these queries, e.g., two recommended queries may return same relevant documents to users. One important future research work is to reduce the redundant utility in the recommendation set. 2) Another interesting work, which is out of the scope of this paper as mentioned before, is how to detect the query session boundary. In this paper, we leverage control query logs, in which all sessions are well segmented. However, in real Web search query logs, identifying search task boundaries is not a trivial work. When we apply our proposed method in large scale Web search logs, a reliable session boundary detection will be valuable.

Acknowledgments. This research work is funded by the European Commission under Grant No. 287704 (CUBRIK) and the National Natural Science Foundation of China under Grant No. 60933005, No. 61173008, No. 61003166 and 973 Program of China under Grants No. 2012CB316303.

References

1. Ageev, M., Guo, Q., Lagun, D., Agichtein, E.: Find it if you can: a game for modeling different types of web search success using interaction data. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 345–354 (2011)
2. Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A.: An optimization framework for query recommendation. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 161–170 (2010)
3. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Boston (1999)
4. Beeferman, D., Berger, A.: Agglomerative clustering of a search engine query log. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 407–416 (2000)
5. Boldi, P., Bonchi, F., Castillo, C., Donato, D., Gionis, A., Vigna, S.: The query-flow graph: model and applications. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 609–618 (2008)
6. Craswell, N., Zoeter, O., Taylor, M., Ramsey, B.: An experimental comparison of click position-bias models. In: Proceedings of the International Conference on Web Search and Web Data Mining, pp. 87–94 (2008)
7. Craswell, N., Szummer, M.: Random walks on the click graph. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 239–246 (2007)
8. Guo, J., Cheng, X., Xu, G., Shen, H.: A structured approach to query recommendation with social annotation data. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 619–628 (2010)
9. Huang, C.-K., Chien, L.-F., Oyang, Y.-J.: Relevant term suggestion in interactive web search based on contextual information in query session logs. Journal of the American Society for Information Science and Technology 54(7), 638–649 (2003)
10. Jain, A., Ozertem, U., Velipasaoglu, E.: Synthesizing high utility suggestions for rare web search queries. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information, pp. 805–814 (2011)
11. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Transactions on Information Systems 20(4), 422–446 (2002)
12. Jones, R., Rey, B., Madani, O., Greiner, W.: Generating query substitutions. In: Proceedings of the 15th International Conference on World Wide Web, pp. 387–396 (2006)
13. Li, L., Yang, Z., Liu, L., Kitsuregawa, M.: Query-url bipartite based approach to personalized query recommendation. In: Proceedings of the 23rd National Conference on Artificial Intelligence, pp. 1189–1194 (2008)
14. Ma, H., Yang, H., King, I., Lyu, M.R.: Learning latent semantic relations from clickthrough data for query suggestion. In: Proceeding of the 17th ACM Conference on Information and Knowledge Management, pp. 709–718 (2008)

15. Mei, Q., Zhou, D., Church, K.: Query suggestion using hitting time. In: Proceeding of the 17th ACM Conference on Information and Knowledge Management, pp. 469–477 (2008)
16. Sadikov, E., Madhavan, J., Wang, L., Halevy, A.: Clustering query refinements by user intent. In: Proceedings of the 19th International Conference on World Wide Web, pp. 841–850 (2010)
17. Serdyukov, P., Rode, H., Hiemstra, D.: Modeling expert finding as an absorbing random walk. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 797–798 (2008)
18. Singh, A.P., Gunawardana, A., Meek, C., Surendran, A.C.: Recommendations using absorbing random walks. In: North East Student Colloquium on Artificial Intelligence (NESCAI) (2007)
19. Wen, J., Nie, J., Zhang, H.: Clustering user queries of a search engine. In: Proceedings of the 10th International Conference on World Wide Web, pp. 162–168 (2001)
20. Zhang, Z., Nasraoui, O.: Mining search engine query logs for query recommendation. In: Proceedings of the 15th International Conference on World Wide Web, pp. 1039–1040 (2006)
21. Zhu, X., Goldberg, A.B., Gael, J.V., Andrzejewski, D.: Improving diversity in ranking using absorbing random walks. In: Proceedings North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT), pp. 97–104 (2007)
22. Zhu, X., Guo, J., Cheng, X., Du, P., Shen, H.: A unified framework for recommending diverse and relevant queries. In: Proceedings of the 20th International Conference on World Wide Web, pp. 37–46 (2011)
23. Zhu, X., Guo, J., Cheng, X., Lan, Y.: More than relevance: high utility query recommendation by mining users' search behaviors. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 1814–1818 (2012)