Enriching Query Flow Graphs with Click Information

M-Dyaa Albakour[1], Udo Kruschwitz[1], Ibrahim Adeyanju[2], Dawei Song[2], Maria Fasli[1], and Anne De Roeck[3]

[1] University of Essex, Colchester, UK
malbak@essex.ac.uk
[2] Robert Gordon University, Aberdeen, UK
[3] Open University, Milton Keynes, UK

Abstract. The increased availability of large amounts of data about user search behaviour in search engines has triggered a lot of research in recent years. This includes developing machine learning methods to build knowledge structures that could be exploited for a number of tasks such as query recommendation. Query flow graphs are a successful example of these structures, they are generated from the sequence of queries typed in by a user in a search session. In this paper we propose to modify the query flow graph by incorporating clickthrough information from the search logs. Click information provides evidence of the success or failure of the search journey and therefore can be used to enrich the query flow graph to make it more accurate and useful for query recommendation. We propose a method of adjusting the weights on the edges of the query flow graph by incorporating the number of clicked documents after submitting a query.

We explore a number of weighting functions for the graph edges using click information. Applying an automated evaluation framework to assess query recommendations allows us to perform automatic and reproducible evaluation experiments. We demonstrate how our modified query flow graph outperforms the standard query flow graph. The experiments are conducted on the search logs of an academic organisation's search engine and validated in a second experiment on the log files of another Web site.

Keywords: Search Log Analysis, Query Suggestions, Automatic Evaluation.

1 Introduction

User interfaces of modern search engines have evolved rapidly in recent years. Modern web search engines do not only return a list of documents as a response to a user's query but they also provide various interactive features that help users in quickly finding what they are looking for or assist them in browsing the information. Google, for example, provides a list of query suggestions while a user is typing in her queries in the search box. Beyond Web search we also observe more interaction emerging as illustrated by the success of AquaBrowser[1] as

[1] http://serialssolutions.com/aquabrowser/

M.V.M. Salem et al. (Eds.): AIRS 2011, LNCS 7097, pp. 193–204, 2011.

a navigation tool in digital libraries. Such interfaces rely on a wealth of knowledge that characterise the domain and specify relations between the different concepts and entities. A number of approaches have been developed to extract knowledge structures that could be exploited to enrich these interfaces. One promising approach is to perform search log analysis which captures the community knowledge about the domain. Query flow graphs extracted from query logs are an example of these approaches which have proven to be useful for providing query recommendations.

In this study, we extend the query flow graph model which relies on query flows as implicit source of feedback by incorporating the post-query user browsing behaviour in the form of clicks. We explore various settings of this model by running an automatic evaluation on actual search logs to understand the impact of various interpretations of click information on the quality of query recommendations.

The paper is structured as follows. We will give a short review of related work in Section 2. Section 3 will describe how we extend the query flow graph model by adding click information using query logs. The experimental setup is explained in Section 4. Results are presented and discussed in Section 5. We will draw conclusions in Section 6 and outline future work in Section 7.

2 Related Work

Query recommendations have become ubiquitous in modern search engines. This is true for Web search engines but also for more specialised search engines. The challenge is to identify the right suggestions for any given search request, and this may depend on a number of factors such as the actual user who is searching, the context, the time of the day etc. A promising route for deriving query recommendations appears to be the exploitation of past interactions with the search engines as recorded in the logs. Several approaches have been proposed in the literature to provide query modification suggestions. Studies have shown that users want to be assisted in this manner by proposing keywords [19], and despite the risk of offering wrong suggestions they would prefer having them rather than not [16].

With the increasing availability of search logs obtained from user interactions with search engines, new methods have been developed for mining search logs to capture "collective intelligence" for providing query suggestions as it has been recognised that there is great potential in mining information from query log files in order to improve a search engine [9,15].

Given the reluctance of users to provide explicit feedback on the usefulness of results returned for a search query, the automatic extraction of implicit feedback has become the centre of attention of much research. Clickthrough data is one form of the implicit feedback left by users which can be used to learn the retrieval ranking function [10], [11], [1]. Queries and clicks can be interpreted as "soft relevance judgements" [6] to find out what the user's actual intention is and what the user is really interested in. Query recommendations can then be derived, for

example, by looking at the actual queries submitted and building query flow graphs [4], [5], query-click graphs [6], cover graphs [3] or association rules [8]. Jones *et al.* combined mining query logs with query similarity measures to derive query modifications [12].

Mining post-query click behaviour has also been studied and applied in information retrieval tasks. For example, Cucerzan *et. al.* [7] used landing page information to derive query suggestions. White *et. al.* [18] mined user search trails for search result ranking, where the presence of a page on a trail increases its query relevance. Click graphs were used by White and Chandrasekar to derive labels to shortcut search trails to help users reach target pages efficiently [17].

Given the successful application of both the query flow graph model as well as post-query click information we explore the potential of extending the query flow graph with click information for deriving query recommendation suggestions.

3 The Model

3.1 The Query Flow Graph

The query flow graph was introduced in Boldi *et al.* [4] and applied for query recommendations.

The query flow graph G_{qf} is a directed graph $G_{qf} = (V, E, w)$ where:

- V is a set of nodes containing all the distinct queries submitted to the search engine and two special nodes s and t representing a *start state* and a *terminate state*;
- $E \subseteq V \times V$ is the set of directed edges;
- $w : E \rightarrow (0..1]$ is a weighting function that assigns to every pair of queries $(q, q') \in E$ a weight $w(q, q')$.

The graph can be built from the search logs by creating an edge between two queries q, q' if there is one session in the logs in which q and q' are consecutive. A session is simply defined as a sequence of queries submitted by one particular user within a specific time limit.

The weighting function of the edges w depends on the application. Boldi *et al.* [4] developed a machine learning model that assigns to each edge on the graph a probability that the queries on both ends of the edge are part of the same chain. The chain is defined as a topically coherent sequence of queries of one user. This probability is then used to eliminate less probable edges by specifying some threshold. For the remaining edges the weight $w(q, q')$ is calculated as:

$$w(q, q') = \frac{freq(q, q')}{\Sigma_{r \in R_q} freq(q, r)} \quad (1)$$

Where:

- $freq(q, q')$ is the number of the times the query q is followed by the query q'.
- R_q is the set of all reformulations of query q in the logs.

Note that the weights are normalised so that the total weights of the outgoing edges of any node is equal to 1.

3.2 Enriching the Query Flow Graph

In this section we explain how we extend the query flow graph model with click data. The intuition here is to use implicit feedback in the form of clickthrough data left by users when they modify their queries which has been shown to be powerful feedback, e.g. [6]. We consider the number of clicked documents by a user after submitting a query as an indication of how useful the results are. This is line with previous work on evaluating search engines with clickthrough data [14].

Let $\phi(q, q') = \{\varphi_0(q, q'), \varphi_1(q, q'), \varphi_2(q, q'), ..\}$ be an array of the frequencies of the reformulation (q, q'), where $\varphi_k(q, q')$ is the number of the times the query q is followed by the query q' and the user has clicked k (and only k) documents on the result list presented to the user after submitting query q'. We aggregate over all users here.

We modify the weighting function in equation 1 to incorporate the click information as follows

$$w(q, q') = \frac{\Sigma_i C_i.\varphi_i(q, q')}{\Sigma_{r \in R_q} \Sigma_i C_i.\varphi_i(q, r)} \tag{2}$$

Where C is an array of co-efficient factors for each band of click counts. Choosing different values for C_i allows us to differentiate between queries that resulted in more or fewer clicks. For example queries which result in a single click might be interpreted as more important than the ones which resulted in no clicks or more than one click as the single click may be an indication of quickly finding the document that the user is looking for.

In our experiments we investigate how different values of the co-efficient C_i affect the quality of the query recommendations. Note that the weighting function of the standard graph in Equation 1 is the special case where $C_0 = C_1 = C_2 = .. = 1$.

3.3 Query Recommendations

Query recommendation is the problem of finding for a given query q relevant query suggestions. If we want to recommend only a single query, then we try to identify the "most important" query q'. The query flow graph can be used for this purpose by ranking all the nodes in the graph according to some measure which indicates how reachable they are from the given node (query). Boldi *et al.* [4] proposed to use graph random walks for this purpose and reported the most promising results by using a measure which combines relative random walk scores and absolute scores. This measure is

$$\overline{s}_q(q') = \frac{s_q(q')}{\sqrt{r(q')}} \tag{3}$$

where:

- $s_q(q')$ is the random walk score relative to q i.e. the one computed with a preference vector for query q.

– $r(q')$ is the absolute random walk score of q' i.e. the one computed with a uniform preference vector.

In our experiments, we adopted this measure for query recommendation and used the random walk parameters reported by Boldi *et al.*

4 Experimental Setup

The aim of the experiments is to investigate whether the query flow graph can be enhanced and how the performance of query recommendations can be affected by different values of the coefficient factors of click counts presented in Equation 2.

The experiments conducted try to answer these questions:

1. Using search logs of a local search engine[2], can we achieve better query recommendations over the standard query flow graph by boosting certain co-efficient factors of click counts and eliminating others?
2. Does the same observation hold true when we use the search of another organisation?

In this section we first provide a description of the search logs used in these experiments. Then we introduce our experimental design and illustrate the different models being tested.

4.1 Search Logs

The main search log data in our experiments are obtained from the search engine of the Web sites of the University of Essex (UOE). In this search log we can obtain the query that has been entered, a time stamp of the transaction and the session identifier. In addition to that the clicked documents from the result lists by users following each query can also be obtained. We used a period of 10 weeks of logs between February and May 2011. During this period a total number of 142,231 queries were submitted to the search engine in 90,684 user sessions and 99,733 clicks on the results were logged. Figure 1 illustrates a histogram of the frequency of queries corresponding to the resulting number of clicks following each query as recorded in the logs of that search engine.

To validate the findings of our experiments on those search logs we conducted further experiments on search logs of another academic institution, the Open University (OU), where the same sort of data can be obtained. Figure 2 shows the corresponding histogram for the logs of the OU search engine using exactly the same 10-week period. It has a similar shape with much higher values of counts. In both histograms, for most cases the users either click on one result or do not click at any.

[2] Here we investigate a search engine of an academic organisation.

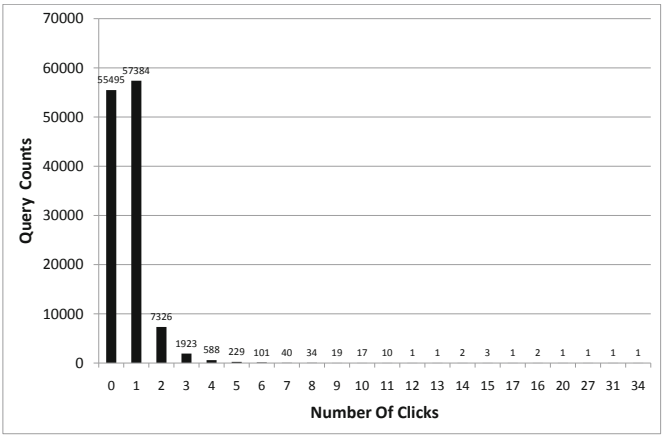

Fig. 1. Frequency of queries for each click counts band - UOE Search Engine

4.2 Query Flow Graphs

To assess the quality of query recommendations that can be achieved using our enriched query graph model we used an automatic evaluation approach based on the search logs to compare the quality of recommendations for various combinations of co-efficient factors of click counts.

Based on the fact that less than 2% of all queries result in more than 2 clicks, we simplified Equation 2 for the experiments as follows:

$$w(q, q') = \frac{C_0.\varphi_0(q, q') + C_1.\varphi_1(q, q') + C_k.\varphi_k(q, q')}{\Sigma_{r \in R_q} \Sigma_i C_i.\varphi_i(q, r)} \tag{4}$$

where C_k is the co-efficient factor of all click counts which are larger than 1. i.e. no matter whether a query has resulted in 2 or more clicks on resulting documents we treat all cases the same.

Table 1 lists all the combinations we considered in running the automatic evaluation framework.

We adopted the frequency weighting used by Boldi *et al.* [4] without incorporating the learning step as our goal is to show how we can enrich the query flow graph with click data. The learning step can always be added to the enriched version of the graph.

$QFG_{standard}$ is the standard query flow graph where no click information are incorporated. QFG_{no_zero} is an enriched query flow graph where reformulations which result with no clicks on the presented document list to the user are not considered. Both QFG_{boost_one} and $QFG_{boost\_one\_more}$ are enriched graphs that boost queries with a single click on the presented list. $QFG_{penalise\_many}$ penalises queries which attract 2 clicks or more.

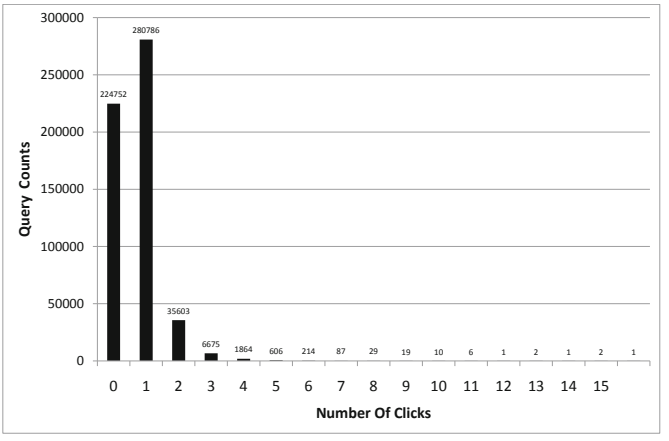

Fig. 2. Frequency of queries for each click counts band - OU Search Engine

Table 1. Experimental Graphs

	C_0	C_1	C_k
$QFG_{standard}$	1.0	1.0	1.0
QFG_{no_zero}	0.0	1.0	1.0
QFG_{boost_one}	1.0	2.0	1.0
$QFG_{boost\_one\_more}$	1.0	3.0	1.0
$QFG_{penalise\_many}$	1.0	2.0	0.5

4.3 The Evaluation Framework

The automatic evaluation framework assesses the performance of query recommender systems over time based on actual query logs by comparing suggestions derived from a query recommender to query modifications actually observed in the log files. The validity of the framework has been confirmed with a user study [2].

The evaluation is performed on arbitrary intervals, e.g. on a weekly basis. For all Q query modifications in a given week, we can calculate the system's Mean Reciprocal Rank (MRR) score as

$$MRR_w = (\sum_{i=1}^{Q} \frac{1}{r_i})/Q \tag{5}$$

where r_i is the rank of the actual query modifications in the list of modifications recommended by the system. Note that in the special case where the actual query modification is not included in the list of recommended modifications then $1/r$ is set to zero. The above evaluation process results in a score for each logged

week. So overall, the process produces a series of scores for each query recommendation system being evaluated. These scores allow the comparison between different system. One query recommender system can therefore be considered superior over another if a statistically significant improvement can be measured over the given period.

In our experiments we start with an empty query flow graph and we go through the search log data. At the end of each interval, we calculate the MRR score for that interval by producing a ranked list of query suggestions using the process described in Section 3.3 and then we use that interval data to update the graph adding necessary edges and adjusting the weights.

Producing query suggestions from the graph is computationally expensive as it requires performing a random walk on the nodes in the graph. Due to computing limitations, when calculating the MRR score we consider only a sample of the query modifications in the batch by taking every tenth query modification.

5 Results and Discussion

The automatic evaluation framework has been run on the various enriched query flow graphs listed in Table 1. We used the log files collected on the UOE search engine for our first experiments. We ran the evaluation on the entire 10-week period and used weekly batches to calculate the MRR scores for each graph.

Using the MRR scores, we can assess the graph performance over time in generating query recommendations and compare the performance of different graphs.

Table 2. Average Weekly MRR scores obtained for the query flow graphs in UOE search logs. The graphs are ordered by their scores.

Graph	Avg. Weekly Score
QFG_{boost_one}	0.0820
$QFG_{boost\_one\_more}$	0.0817
$QFG_{penalise\_many}$	0.0812
$QFG_{standard}$	0.0789
QFG_{no_zero}	0.0533

Table 2 presents the average weekly MRR scores obtained (ordered by average score). We observe that the enriched query flow graphs are outperforming the standard query flow graph. Apart from QFG_{no_zero} all enriched graphs are producing higher average MRR scores. To perform a statistical analysis on the differences between the enriched query flow graphs, in Table 3 we compare the query flow graphs using the average percent increase of MRR scores and the p value of a two-tailed t-test.

We observe that when boosting the co-efficient factor of single clicks, statistically significant improvements are obtained. Both QFG_{boost_one} and

Table 3. Comparison of the query flow graphs (UOE search engine)

	per. increase(%)	paired t-test
QFG_{boost_one} vs. $QFG_{standard}$	2.3%	< 0.05
$QFG_{boost\_one\_more}$ vs. $QFG_{standard}$	2.2%	< 0.05
$QFG_{boost\_one\_more}$ vs. QFG_{boost_one}	-0.1%	0.91
$QFG_{penalise\_many}$ vs. QFG_{boost_one}	-0.8%	0.16
QFG_{no_zero} vs. $QFG_{standard}$	-61.2%	< 0.01

$QFG_{boost\_one\_more}$ are significantly better than the standard query flow graph $QFG_{standard}$. However no further improvement can be observed when we further boost the co-efficient factor of single clicks. In fact $QFG_{boost\_one\_more}$ is slightly worse than QFG_{boost_one}.

Comparing $QFG_{penalise\_many}$ to QFG_{boost_one} would inform us about the impact of reducing the co-efficient factor of more than one click counts. The results show that this does not have a positive impact on the quality of recommendations. $QFG_{penalise\_many}$ is worse than QFG_{boost_one}.

Only enriched graph QFG_{no_zero} failed to improve the MRR scores, and in fact it was significantly worse than the standard graph with a high average percentage decrease. This appears to be counter-intuitive as we would assume that queries resulting in no clicks are not good candidates for query recommendation suggestions, and this finding warrants further analysis in future experiments.

In any case, this last finding suggests that completely eliminating reformulations with no user clicks affects the query recommendation quality negatively. Note that in QFG_{boost_one} and $QFG_{boost\_one\_more}$ we are considering these reformulations but we are also penalising them as they have a smaller co-efficient factor.

To validate the findings we obtained the log files of another academic search engine. To get a comparable number of interactions we decided to run this experiment in daily batches over 10 days of the April 2011 logs, i.e. we now use daily intervals to update the graph and calculate the MRR scores.

Table 4 presents the results obtained in this experiment. The corresponding t-test results can be found in Table 5.

Table 4. Average Daily MRR scores obtained for the query flow graphs in OU search logs. The graphs are ordered by their scores.

Graph	Avg. Daily Score
QFG_{boost_one}	0.0488
$QFG_{penalise\_many}$	0.0480
$QFG_{boost\_one\_more}$	0.0478
$QFG_{standard}$	0.0476
QFG_{no_zero}	0.0425

Table 5. Comparison of the query flow graphs (OU search engine).

	per. increase(%)	paired t-test
QFG_{boost_one} vs. $QFG_{standard}$	2.1%	0.15
$QFG_{boost\_one\_more}$ vs. $QFG_{standard}$	0.1%	0.88
$QFG_{boost\_one\_more}$ vs. QFG_{boost_one}	-2.0%	< 0.05
$QFG_{penalise\_many}$ vs. QFG_{boost_one}	-1.4%	< 0.05
QFG_{no_zero} vs. $QFG_{standard}$	-9.9%	< 0.01

Despite some minor differences we can see the same pattern. The ordering of the graphs according to their average MRR scores is similar. Only positions 2 and 3 ($QFG_{boost\_one\_more}$ and $QFG_{penalise\_many}$) are swapped. The enriched query flow graphs are outperforming the standard query flow graph but no statistical significant was observed this time.

Like before, reducing the co-efficient factor of many click counts did not have a positive impact on this dataset either. In fact $QFG_{penalise\_many}$ is now significantly worse than QFG_{boost_one}. Again, we find that eliminating queries that result in no clicks does not improve performance but instead results are significantly worse.

6 Conclusions

Query flow graphs built from query logs are a common and efficient technique to learn useful structures that can be utilised in query recommendation. We presented a new approach for incorporating user post-query browsing behaviour in the query flow graph. This is done by taking into account the number of documents that have been clicked by the user after submitting a query.

In this paper we explored variations of interpreting the number of clicked documents by conducting controlled, deterministic and fully reproducible experiments. which are based on an automatic evaluation framework that uses real world data to assess the performance of different models. Our experiments allowed us to quantitatively answer our research question and to draw very useful conclusions.

Boosting queries which result in a single document click has a positive impact on query recommendation. A single click can be interpreted as quickly reaching a landing page and rewarding these queries significantly improved the automatic evaluation scores. This is line with previous findings on using landing pages to generate query recommendations [7].

Eliminating queries which result in no clicks negatively impacted query recommendation. One possible explanation (but certainly only one single aspect) could be that some users found what they are looking for in the result snippets and as a result they would not continue clicking on the right document. Therefore, the graph will miss those useful suggestions. Penalising these reformulations without completely eliminating them though would have a positive

effect as graphs QFG_{boost_one}, $QFG_{boost\_one\_more}$ have a smaller co-efficient factors for zero clicks.

We also show the observation made on one dataset was similar on a different dataset. The performance of the experimented graphs was similar on both datasets. However no statistical significance was observed for the enriched query flow graph over the standard query flow graph on the OU search engine. This may be due to the higher sparsity of the OU search engine logs.

7 Future Work

There is much room for future work. One area we will investigate is to automatically optimise the parameters. An extension of that work will then also allow us to look at building a machine learning model which can be trained on actual search log data taking as features the post query browsing behaviour including the click information to optimise the graph weighting function. Other browsing behaviour features can be further explored.

The appeal of an automated evaluation framework is that we can re-run experiments and explore a large search space without any user intervention. The shortcoming is that any automated evaluation makes some simplifying assumptions, and end users will ultimately need to be involved to assess the real impact of the query recommendation suggestions being employed. We see our evaluation as a first step in assessing what methods are promising and select those that promise the highest impact. We are about to incorporate a number of these models in a live Web site where we interleave recommendations coming from different models in the spirit of the active exploration approach presented by Radlinksi *et al.* [13]

Acknowledgments. This research is part of the AutoAdapt[3] research project. AutoAdapt is funded by EPSRC grants EP/F035357/1 and EP/F035705/1.

References

1. Agichtein, E., Brill, E., Dumais, S.: Improving web search ranking by incorporating user behavior information. In: Proceedings of SIGIR 2006, pp. 19–26. ACM, New York (2006)
2. Albakour, M.-D., Kruschwitz, U., Nanas, N., Kim, Y., Song, D., Fasli, M., De Roeck, A.: AutoEval: An Evaluation Methodology for Evaluating Query Suggestions Using Query Logs. In: Clough, P., Foley, C., Gurrin, C., Jones, G.J.F., Kraaij, W., Lee, H., Mudoch, V. (eds.) ECIR 2011. LNCS, vol. 6611, pp. 605–610. Springer, Heidelberg (2011)
3. Baeza-Yates, R., Tiberi, A.: Extracting semantic relations from query logs. In: Proceeding of KDD 2007, San Jose, California, pp. 76–85 (2007)
4. Boldi, P., Bonchi, F., Castillo, C., Donato, D., Gionis, A., Vigna, S.: The query-flow graph: model and applications. In: Proceeding of CIKM 2008, pp. 609–618. ACM, New York (2008)

[3] http://autoadaptproject.org

5. Bordino, I., Castillo, C., Donato, D., Gionis, A.: Query similarity by projecting the query-flow graph. In: Proceedings of SIGIR 2010, Geneva, pp. 515–522 (2010)
6. Craswell, N., Szummer, M.: Random Walks on the Click Graph. In: Proceedings of SIGIR 2007, Amsterdam, pp. 239–246 (2007)
7. Cucerzan, S., White, R.W.: Query suggestion based on user landing pages. In: Proceedings of SIGIR 2007, pp. 875–876. ACM, New York (2007)
8. Fonseca, B.M., Golgher, P.B., de Moura, E.S., Ziviani, N.: Using association rules to discover search engines related queries. In: Proceedings of the First Latin American Web Congress, Santiago, Chile, pp. 66–71 (2003)
9. Jansen, J., Spink, A., Taksa, I. (eds.): Handbook of Research on Web Log Analysis. IGI (2008)
10. Joachims, T., Granka, L., Pan, B., Hembrooke, H., Gay, G.: Accurately interpreting clickthrough data as implicit feedback. In: Proceedings of SIGIR 2005, Salvador, Brazil, pp. 154–161 (2005)
11. Joachims, T., Radlinski, F.: Search engines that learn from implicit feedback. IEEE Computer 40(8), 34–40 (2007)
12. Jones, R., Rey, B., Madani, O.: Generating query substitutions. In: Proceedings of WWW 2006, pp. 387–396 (2006)
13. Radlinski, F., Joachims, T.: Active exploration for learning rankings from clickthrough data. In: Proceedings of KDD 2007, pp. 570–579. ACM, New York (2007)
14. Radlinski, F., Kurup, M., Joachims, T.: How does clickthrough data reflect retrieval quality? In: CIKM, pp. 43–52 (2008)
15. Silvestri, F.: Mining query logs: Turning search usage data into knowledge. Foundations and Trends in Information Retrieval 4, 1–174 (2010)
16. White, R.W., Bilenko, M., Cucerzan, S.: Studying the Use of Popular Destinations to Enhance Web Search Interaction. In: Proceedings of SIGIR 2007, Amsterdam, pp. 159–166 (2007)
17. White, R.W., Chandrasekar, R.: Exploring the use of labels to shortcut search trails. In: Proceeding of SIGIR 2010, pp. 811–812. ACM, New York (2010)
18. White, R.W., Huang, J.: Assessing the scenic route: measuring the value of search trails in web logs. In: Proceeding of SIGIR 2010, pp. 587–594. ACM, New York (2010)
19. White, R.W., Ruthven, I.: A Study of Interface Support Mechanisms for Interactive Information Retrieval. JASIST 57(7), 933–948 (2006)