Generating Simulated Relevance Feedback: A Prognostic Search approach

Nithin Kumar M and **Vasudeva Varma**
Search and Information Extraction Lab,
International Institute of Information Technology Hyderabad,
nithin_m@research.iiit.ac.in and vv@iiit.ac.in

Abstract

Implicit relevance feedback has proved to be a important resource in improving search accuracy and personalization. However, researchers who rely on feedback data for testing their algorithms or other personalization related problems are loomed with problems like unavailability of data, staling up of data and so on. Given these problems, we are motivated towards creating a synthetic user relevance feedback data, based on insights from query log analysis. We call this simulated feedback. We believe that simulated feedback can be immensely beneficial to web search engine and personalization research communities by greatly reducing efforts involved in collecting user feedback. The benefits from "Simulated feedback" are - it is easy to obtain and also the process of obtaining the feedback data is repeatable, customizable and does not need the interactions of the user. In this paper, we describe a simple yet effective approach for creating simulated feedback. We have evaluated our system using the clickthrough data of the users and achieved 77% accuracy in generating click-through data.

1 Introduction

Implicit relevance feedback serves as a great source of information about user behaviour and search context. A lot of research went through in the recent past in making use of this great pool of information. Relevance feedback is proven to significantly improve retrieval performance (Harman, 1992; Salton and Buckley, 1990). It has also been successfully used to improve searching ranking, query expansion, personalization, user profiling et cetera (Steve Fox et al.,, 2005; Rocchio, 1999; Xuehua et al., 2005).

Clickthrough data is the most prevalent form of implicit feedback used by researchers for personalization purposes. Click log data provides valuable information about the interests, preferences and semantic search intent of the user (Daniel and Levinson, 2004; Kelly and Belkin, 2001). Unlike explicit feedback, clicks logs do not require any special effort from the user (Rocchio, 1999). It is collected in the background while the user interacts with the search engine to quench his information need. Hence, it is easy and feasible to collect large amounts of clickthrough data.

However, using clickthrough data has its own share of problems. Firstly, it is not available for public or even research communities at large for reasons like being a potential threat to privacy of web users. Secondly, it only contains the URLs of the results that the user clicked and does not contain the documents that the user has chosen. Given the dynamic nature of the web, content of many of the urls is prone to change and in some cases it might not exist. In other cases, even if the old expected results remain good resources, search engines might not retrieve them in response to queries. It will return near-duplicate pages that have equivalent content but different URLs. Thus feedback data may rapidly become stale with new pages replacing old ones as more approporiate resources. And also, given the rapidly changing ranking algorithms of web search engines, feed-

Coling 2010: Poster Volume, pages 597–604,
Beijing, August 2010

back data collected from the users becomes outdated. Hence researchers who rely on feedback data either for testing their algorithms or other personalization related problems are faced with the problems of non-availability of user feedback data.

In this paper, we strive to address the above problems by generating simulated relevance feedback using prognostic search techniques. *Prognostic search* is a process of simulating user's search process and emulating their actions, through preferences captured in their profile. Such generated feedback can be used for research in personalization techniques and analyzing personalization algorithms and search ranking functions(Harman, 1988). The main advantage with this system is that we can create data on the fly and hence not fear of it becoming stale. Since it does not involve user's actions, it is feasible to generate large amounts of data in this way.

2 Contributions and Organization

In this paper, we propose a novel way of creating simulated feedback. The data thus produced can be used for evaluating/training personalization systems. Using our proposed method, given a user's training data, we can produce synthetic implicit feedback data - simulated feedback data on the fly. We also propose a novel user browsing model which extends the high performing cascade model of (Craswell et al., 2008). Our *Patience* parameter can be used to build more complex user browsing models to bring the whole process of generating implicit feedback data a step nearer to the real world mechanisms.

In section 3, we describe our approach to generate simulated feedback data. In sections 3.2.3 and 3.2.4, we describe the process of browsing results and generating clicks which form the crux of our approach. We evaluate our system and prove the usefulness of it in section 4. Section 5 and 6 give an account of our experiments and the study of works related to ours already present in the literature. We conclude that our proposed approach can be highly useful in personalization research and give an account of our future directions in section 7.

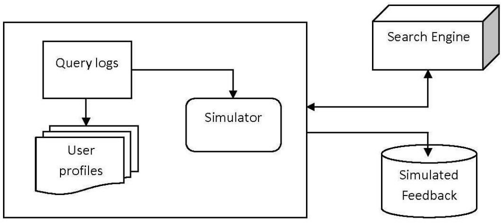

Figure 1: System architecture

3 Proposed Approach

Simulated feedback is a new type of feedback similar to implicit and explicit relevance feedback. Simulated feedback is created by observing and analyzing real world search log data. We propose a two phase process to create simulated relevance feedback as follows: In phase 1, we process real world click-through data of a search engine and build user profiles using the data. In phase 2, we simulate a user's search process and emulate their actions based on their profile. We call this process as "*Prognostic Search*".

3.1 Creating Profiles

After closely examining and analyzing the semantics of the query log, we have chosen the following parameters to characterize a user: an anonymous user-id, *perceived relevance threshold*, *patience*, previous queries issued and search history of the user.

A user-id is used to distinguish and uniquely identify each and every user. *Perceived relevance* is the relevance estimate of the result according to the user on examining the title, snippet and the url of the result. And *Perceived relevance threshold* is the threshold limit of *perceived relevance* of a result for the user to click it. *Patience* of the user is the trait which determines the number of clicks and the depth to which the user examines the results. We explain the process of computing a user's patience parameter in detail in section 3.2.3. We stored the previous queries and clicks of the user to capture the preferences of the user.

To make use of the search history, we used the previous queries issued and previous results clicked by the user. We store the titles and snippets of those results to capture the interests of the user. Here, our aim is to generate implicit relevance feedback which is very close to the real world data. To generate synthetic relevance feedback, we instantiate these parameters with appropriate values using real world data.

3.2 Prognostic Search

Prognostic search is simulation of a user's search process and emulating their actions based on their interests and preferences captured in their user profile. Simulating search process involves four steps viz., i)Query formulation, ii)Searching, iii)Browsing results and iv)Generating Clicks. Each of these processes are explained below.

3.2.1 Query Formulation

Query formulation involves cognitive process of the user and requires background knowledge about the user like their interests, preferences and their knowledge base. It is highly impossible to capture the cognitive thought process of a user and emulate their method of generating a query. To solve this problem, we randomly select a search session from a user's history and send all the queries in it sequentially to the search engine. This helps us to preserve the inter query relations that naturally exist between the subsequent queries in a session.

3.2.2 Searching

This step involves retrieving documents relevant to the query generated in the previous step. We used yahoo search engine which is very much similar to the search engine from which the training data is collected.

3.2.3 Browsing results

In this step, we simulate the manner in which a user browses the results in the real world. Based on the observations in (Granka et al., 2004; Filip and Joachims, 2005), we assume that the user in the real world follows the browsing model explained in Algorithm 1. In real world, a user may follow more complex browsing models, but presently we have considered this browsing model to simplify things.

Algorithm 1 User browsing model in real world

```
Step1: Start browsing with the top-most result.
Step2: Examine title, snippet and URL of the result.
Step3: Click if the result looks promising.
Step4: If(user has patience) go to step 5, else go to step 6.
Step5: Select next result and go to step 2.
Step6: Start examining the clicked results.
Step7: If(information need satisfied) end the process, else go to step 8.
Step8: Reformulate the query and go to step 1.
```

Accordingly, to simulate the browsing process of the user explained in algorithm 1, we followed the Algorithm 2.

Algorithm 2 Simulated User browsing model

```
Step 1: Determine the number of results to be browsed based on patience parameter.
Step 2: Browse the results in increasing order of their ranks and examine them.
Step 3: Compute the perceived relevance score of the results.
Step 4: In the same order, generate clicks based on the perceived relevance scores of the results.
Step 5: If(session has more queries) go to step 6, else end the process.
Step 6: Select next query in the session and go to step 1.
```

Thus based on the *patience* parameter, we determine the number of results that the user browses. In our analysis of query log parameters, we learned that the *patience* value of a user can be characterized by the following parameters: number of clicks per session, maximum rank of the result clicked in a session, time spent in a session, the number of queries issued per second and the average semantic relevance of the top ten results of that session to the user. We found out that the patience of the user is directly proportional to the maximum rank of the result he has clicked in a session. We also found out that the number of clicks a user generates is inversely proportional to the number of queries he issues per second and directly proportional to the amount of time he spends per session. Thus, a user with more *patience* tends to examine more search results and thus generate more clicks based on their relevance. We explain these dependencies in detail in the experiments section. So in order to learn the Patience parameter of the user, we devised the following formula:

$$\text{Patience} = \alpha \times \frac{(MR \times T \times C \times S_{q_i})}{Q} \quad (1)$$

Here MR denotes the average of maximum rank of the results clicked by the user in a session, T denotes the average time spent in a session, C is the average number of clicks in a session and Q denotes the average number of queries issued per session and S_{q_i} is the average semantic distance of the top ten results of the query 'q_i'. Here, "α'' is an equalization constant.

3.2.4 Generating clicks

This is the most important step in our simulation process. Typically, a user observes the visual information viz., title, snippet and the URL of a result(Joachims et al., 2005). Then based on their interests, they choose the results relevant to them. Similarly, we closely examine the results selected in the previous step and then score them according to their relevance to the user. We consider the title, snippet and the page-rank of the result and determine its relevance to the user known as *perceived relevance* score.

We first compute the semantic distance between the title and snippet of the present result from the titles and snippets of previously clicked results of the user. The results already clicked by the user serve as a knowledge base of the interests and preferences of the user. Thus, the semantic distance between the present result and the previous result gives us an account of the relevance that the present result carries to the user.

We used latent semantic analysis (LSA) to compute the semantic distance between the results. LSA does not take the dictionary meaning of the words as input; it rather extracts the contextual meaning of the word with respect to all other words in semantic space(Landauer et al., 2007). This property of LSA is very much useful in the present context. A particular word may have a lot of meanings but we are concerned about only those meanings of the word which the user interprets, which are captured in the sentences present in the user's click history. Hence, we used LSA to compute the semantic distance between the results.

We also consider the page-rank of the result, which has proven to be an important factor in making the decision of a click. In our study, we found that for about 89% of the queries with clicks, the top ranked document has been clicked and for 56% of the queries second ranked document has been clicked. In Figure 3, we show the click ratio for each of the top ten ranked documents[1]. Thereby, we derive that the rank of the result is also a very important factor in deciding whether a result has to be clicked or not. We also consider the distance of the present result from the previous click of the user. In (Joachims et al., 2005), it is shown that the user is more biased to click the result that immediately follows the result he previously clicked. In our simulation process, if this distance for any result exceeds 10, then we terminate the browsing process and reformulate the query. We believe that when this distance exceeds 10, it signifies that the quality of the results is low and hence can be ignored.

We used the bayesian probabilistic techniques to calculate the probability of the user clicking a result based on the above discussed factors. Hence *Click* being a Bernoulli variable, we have

$$P(c/R, q, u) = \alpha_{R,q,u}^{c} \, (1 - \alpha_{R,q,u})^{1-c} \quad (2)$$

Where $\alpha_{R,q,u}$ is the probability that user 'u' clicks the result 'R' for a query 'q'. We model the probability of a click, $P\,(c/R, q, u)$ as a joint probability of P(c,r,Rel,D) where 'r' denotes the rank of the result, 'Rel' denotes the semantic relevance score of the result to the user – precisely to his previous clicks – and 'D' denotes the distance of the previous click of the user. We use this probability of the result as the *Perceived relevance* score of the result. Thus, we have:

[1] In figure 3, we have normalized the clicks statistics with the number of clicks for top ranked document. So, the click-ratio for the top ranked document will be 1.

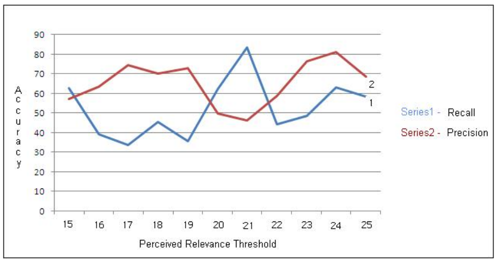

Figure 2: Graph showing Precision and Recall of generating clicks for a particular user

$$\text{Perceived relevance} = P\left(c/R, q, u\right) = P\left(c/r, Rel, D\right) \propto ln\left[P\left(r/c\right)\right] + ln\left[P\left(Rel/c\right)\right] + ln\left[P\left(D/c\right)\right] + ln\left[P\left(c_{i+1}\right)\right] \quad (3)$$

Here, 'r' denotes the rank of the result, 'Rel' denotes the *perceived relevance* of the result to the user and 'D' denotes the distance of the result from the user's previous clicked result. Prior probablities of each of these factors are calculated from the data stored in the user profile. We used Laplace smoothing techniques to deal with zero probability entries. $P(c_{i+1})$ is the probability that the user may click a result after clicking 'i' results. We also believe that the behaviour of the user changes with each click he generates in a session. Hence we used the factor $P(c_{i+1})$ in determining the probability of the click[2]. Then, we compare this score with the *Perceived relevance threshold* of the user and generate the clicks accordingly.

Computing *Perceived Relevance Threshold*: Using the above formula, we generated clicks for different values of *Perceived Relevance Threshold* for a user. Figure 2 show the precision and recall values of generating clicks for different values of Perceived Relevance Threshold of a user. Thus, we plot the accuracy of our system for different values of *Patience Relevance Threshold* and accordingly set the threshold selecting the best values for precision and recall of the system.

4 Experiments

Clickthrough data is a valuable source of user information. In our statistical analysis of clickthrough data, we have found that the page-rank of a result can highly influence the user to make a click which can be seen in figure 3.

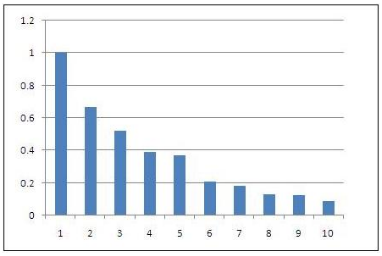

Figure 3: Ranks Vs Clicks-ratio

In our definition of *Patience*, we termed it as parameter to denote the depth to which the user examines the results and the number of clicks he generates. In equation 1, we show that the patience value is inversely proportinal to the number of queries the user issues in a session. To prove this fact, we made a statistical analysis on the real world querlogs[3]. From the graphs shown in figure 4, it can be clearly seen that the *Patience* of the user is inversely proportional to the user's number of Queries/sec. These graphs show the influence of the factor Queries/sec on the number of clicks the user generates for a query and the maximum rank clicked by the user in a session. We drew the graphs averaging the different queries/sec value of a user in a session for each value of MR and number of clicks respectively. It is evident that both the graphs are weakly decreasing functions. Since maximum rank clicked and the number of clicks per session directly affect the *Patience* parameter, we can say that Queries/sec is inversely proportional to the *Patience* of the user.

Both the graphs show occasional phases of increasing behaviour which can be attributed to a variety of reasons. While plotting the graphs, for a given value of MR/number of clicks, we take observations from numerous sessions of the user and average the queries/sec value. Thus, presence of some outlier values may affect the overall out-

[2] We used laplace smoothing technique to negate the effect of zero probability instances.

[3] We performed these experiments on the query log data of a popular commerical search engine. The data consists of 21 million web queries collected from 650,000 users. The query log data consists of anonymous id given to the user, query, the time at which the query was posed, rank of the clicked URL (if any) and the URL of the document clicked by the user (if any).

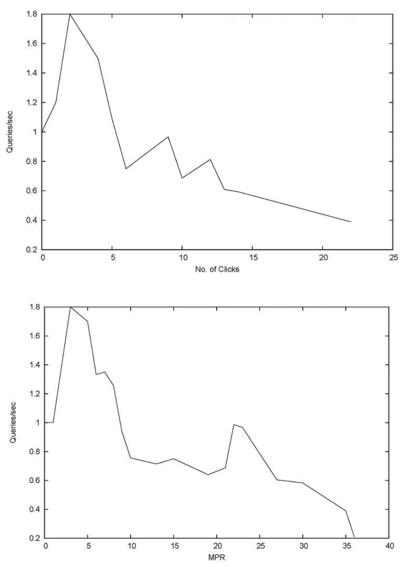

Figure 4: Clicks Vs Queries/sec and MR Vs Queries/sec

put of the graph. It can also be attributed to the low quality of results that the search engine might have returned due to various reasons.

5 Related Work

Although simulation-based methods have been used to test query modification techniques (Harman, 1988) or to detect shifts in the interests of computer users (Mostafa et al., 2003), to our knowledge not much research went into creating relevance feedback for web search based on search simulations.

Searcher simulations were created by White et al (Mostafa et al., 2003; White et al., 2005), for evaluating implicit feedback models. The simulations assume the role of a searcher, browsing the results of a retrieval. It is assumed that the actual relevant and irrelevant documents for a query are given. The system creates simulations of searchers by simulating relevance paths i.e., how the user would traverse results of retrieval. Different strategies were experimented like, the users only view relevant/non-relevant information, i.e., follow relevant paths from only relevant or only non-relevant documents, or they view all relevant or all non-relevant information, i.e., follow all relevance paths from top-ranked relevant documents or top-ranked non-relevant documents etc. Their research tries to model only certain phases of the search process like clicking the results and to some extent the process of looking and identifying the results to click. It also does not consider modeling the nature of the searcher in context and also does not calculate the relevance of a document for a user. The search process is not complete without discussing or characterizing the user that participates in the search and computing the relevance of a document for a user.

In (Agichtein et al.,, 2006), they show that clickthrough data and other implicit data of a user can be used to build user models to effectively personalize the search results. Craswell et al (Craswell et al., 2008) have also done some good work in this area. They try to model the results browsing pattern of the user. (Craswell et al., 2008) brings out the position bias in the user's click-decision making process. It provides some interesting browsing models which can be used in our prognostic search process. We used the cascade model – best performing model – proposed by them to compare the effectiveness of our approach.

In our approach, we address some of these issues to improve the reliability of the simulated feedback and the scalability of the simulations. We first identify certain parameters that are natural to the search process on the whole and are generic to hold well across search engines and users. Wherever applicable we try to characterize these parameters as probabilistic distributions, using large volumes of data from existing search engine clickthrough logs. We then instantiate these parameters by drawing values from these probabilistic distributions. This ensures that the simulated feedback resembles as closely as possible to the real world scenario and thus is of high quality. We can easily run the simulations on large sets of documents to create large amounts of simulated feedback, as there are no interventions of a human to provide any kind of extra information or relevance information on the document set.

6 Evaluation

In this section, we present the evaluation procedure of our approach. We first collected query

Table 1: System Configurations

System	Patience	Clicks
System1	Random	Random
System2	Random	Proposed method
System3	Proposed method	Proposed method

log data of 60 users using a browser plug-in for two months. Our query log data consists user-id, queries and the time at which they are entered, list of search results – rank, title, snippet and url of the result –[4] and the results clicked by the user. We used 70% of this query log data to build profiles of the searchers and the rest of the data is used for evaluation purpose. Using the rest of the query log data, we initiated the prognostic search process giving the queries sequentially in the order given by the user. We compared the simulated clicks with the clicks already generated by the user. We found that the data generated by us is 77% accurate and its recall[5] value is 68%. We measured the accuracy of our system as follows.

$$\text{Accuracy} = \frac{\text{No. of simulated clicks clicked by the user}}{\text{Total no. of results clicked by the user}} \quad (4)$$

We also built two more systems which we considered as the baseline systems. The first system gives a random value for the patience value of the user – random value is used to determine the number of documents to be browsed during the prognostic search process – and random value is given for the user's *Perceived relevance threshold* parameter. The second system generates the patience value of the user according to the process described by us in section 3.2.3 and gives a random value for the *Perceived relevance threshold* value of the user. Systems built by us can be summarized as shown in table 1:

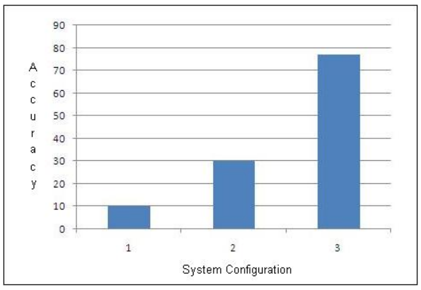

Figure 5: Results comparision

Figure 5 shows a comparision of the accuracies of the three systems. Here, we can see that the baseline 1 which uses random values for *patience* and *generating clicks* is only 10% accurate in generating clickthrough data. However, with the addition of our *generating clicks* approach to the baseline 1, the performance increased by 200%. And the system 3 which uses our proposed models for both *patience* and *generating clicks* generates 77% accurate data which is a 670% improvement over the baseline 1.

We also performed manual evaluation of our system. Since manual evaluation requires a lot of effort, we performed it using 25 judges. We randomly selected 25 users from our query log data and used their data to build profiles. Then we showed the clicks generated by our system to these users. Based on their judgements, we found our system to be 79.5% accurate[6]. Figure 6 shows the accuracy levels of our system according to different judges. We also studied the reason behind the increase in accuracy of our system during human evaluation. We re-examined the clicks generated by the users and found that the users selected the results which they have not selected during their regular search. And the reasons behind these extra clicks are: they have missed examining these results or they have already reached their desired document. Thus it certifies that our system is able to personalize the results and the perceived relevance technique can be used to re-rank the results to personalize them.

As the *cascade* model is the best performing model in (Craswell et al., 2008), we evaluated our system on that model for comparision. We found our system to be 96% accurate. We used the data collected in our clickthrough logs for evaluating

[4] A typical search engine query log does not contain the snippets of the results and the whole list of search results. It only contains the link clicked by the user and the rank of that result.

[5] Recall is the fraction of results clicked for this query and simulated successfully

[6] We took the average of the accuracies of our system for each of these judges/users.

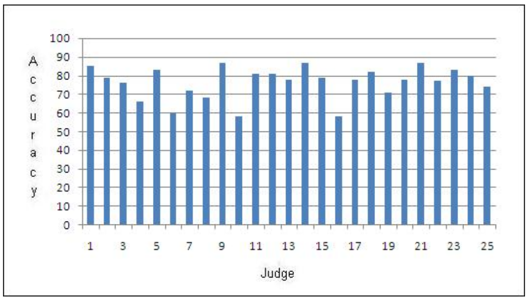

Figure 6: Accuracy based on human judge evaluation

our system using this model.

7 Conclusion and Future work

In this paper, we proposed Simulated Feedback based on insights from clickthrough data and using prognostic search methods to generate feedback. There is a lot of scope for interesting future directions to the current work. It would be an interesting experiment to see the use of the simulated feedback in evaluation of personalized search algorithms. Consider a personalized search algorithm, and use it to learn a user model from existing explicit/implicit feedback data. Learn a user model using the same algorithm from simulated feedback and compare the results. We plan to pursue the same in future.

As an extension to the current work, we aim to improve the web search process especially the query formulation step with insights from a user study. We are working towards incorporating much richer and complex models for query formulation like HMMs etc. Ability of the system to automatically create query reformulations of the original when no clicks are found is another interesting future work. We also plan to dig more information about the user by analysing the query log data. For example, the difference in the time between the clicks and the distance between the clicks can be used to analyze the browsing behaviour of the user. These observations can inturn be used in generation of simulated feedback thus reducing its gap with real world implicit feedback.

References

Mark Claypool, Phong Lee, Makoto Wased and David Brown. 2001. *Implicit interest indicators*. In Intelligent User Interfaces.

Granka L., Joachims J., and Gay G. 2004. *Eyetracking analysis of user behavior in www search*. Conference on Research and Development in Information Retrieval, SIGIR.

Harman D. 1988. *Towards interactive query expansion*. The 11th Annual ACM SIGIR Conference on Research and Development in Information Retrieval, 321-331.

Thorsten Joachims. 2002. *Optimizing search engines using clickthrough data*. Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, 133-142.

Kelly D., and Belkin N.J. 2001. *Reading time, scrolling and interaction: Exploring implicit sources of user preferences for relevance feedback during interactive information retrieval*. In Proceedings of the 24th Annual International Conference on Research and Development in Information Retrieval, SIGIR, 408-409.

Mostafa J., Mukhopadhyay S., and Palakal M. 2003. *Simulation studies of different dimensions of users' interests and their impact on user modelling and information filtering*. Information Retrieval, 199-223.

Filip Radlinski and Thorsten Joachims. 2005. *Evaluating the robustness of learning from implicit feedback*. In ICML Workshop on Learning In Web Search.

Rocchio J.J. 1999. *The SMART Retrieval System Experiments in Automatic Document Processing*. Relevance Feedback in Information Retrieval.

Sugiyama K., Hatano K., and Yoshikawa M. 2004. *Adaptive web search based on user profile constructed without any effort from users*. In Proceedings of WWW, 675-684.

Ryen W. White, Ian Ruthven, Joemon M. Jose and C.J van Rijsbergen. 2005. *Evaluating implicit feedback models using searcher simulations*. ACM Transactions on Information Systems,ACM TOIS, 325-361.

Xuehua Shen, Bin Tane and Bin Tan. 2005. *Implicit user modeling for personalized search*. ACM Transactions on Information Systems.

Feng Qiu and Junghoo Cho. 2006. *Automatic Identification of User interest for personalized search*. In proceedings of WWW.

Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais and Thomas White. 2005. *Evaluating implicit measures to improve web search*. ACM Transactions on Information Systems, 147-168.

Eugene Agichtein, Eric Brill, Susan Dumais and Robert Ragno. 2006. *Learning user interaction models for predicting web search result preferences*. In proceedings of 29th conference on research and development in information retrieval, SIGIR, 3-10.

Thorsten Joachims, Laura Granka and Bing Pan. 2005. *Accurately interpreting clickthrough data as implicit feedback*. In proceedings of 28th conference on research and development in information retrieval, SIGIR.

Thomas K. Landauer, Danielle S. Mc Namara and Simon Dennis. 2007. *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates.

Craswell N., Zoeter O., Taylor M. and Ramsey B. 2008. *An experimental comparision of click position-bias models*. In First ACM International Conference on Web Search and Data Mining WSDM.

Olivier Chapelle and Ya Zhang. 2009. *A Dynamic Bayesian Network Click Model for Web Search Ranking*. In proceedings of International World Wide Web Conference(WWW).

Fan Guo, Chao Liu and Yi-Min Wang. 2009. *Efficient Multipl-Click Models in Web Search*. In Second ACM International Conference on Web Search and Data Mining WSDM.

Harman D. 1992. *Relevance feedback revisited*. In proceedings of 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1-10.

Salton G., and Buckley C. 1990. *Improving retrieval performance by relevance feedback*. Journal of the American Society for Information Science.

Daniel E. Rose, and Danny Levinson. 2004. *Understanding user goals in Web Search*. In proceedings of International World Wide Web Conference(WWW).