# IMPROVING WEB SEARCH BY EXPLOITING SEARCH LOGS

Hongyan Ma
*College of Information, Florida State University, PO Box 3062100, Tallahassee, FL, U.S.A.*

Keywords: Web searching, Relevance feedback, Probabilistic model, Log mining.

Abstract: With the increased use of Web search engines, acute needs evolve for more adaptive and more personalizable Information Retrieval (IR) systems. This study proposes an innovative probabilistic method exploiting search logs to gather useful data about contexts and users to support adaptive retrieval. Real users' search logs from an operational Web search engine, *Infocious,* were processed to obtain past queries and click-through data for adaptive indexing and unified probabilistic retrieval. An empirical experiment of retrieval effectiveness was conducted. The results demonstrate that the log-based probabilistic system yields statistically superior performance over the baseline system.

## 1 INTRODUCTION

In recent years, we have witnessed the explosive growth of information on the World Wide Web; there are billions of static Web pages and perhaps even more information in the hidden Web. Meanwhile, people are relying more and more on the Web for their diverse information needs. Web searching becomes the first step in many information seeking tasks. Hundreds of millions of queries are issued per day. Millions of users directly interact with search engine systems on a daily basis (Burns, 2007).

However, the current search engine systems are still far from optimal (for example, Agichtein, Brill, & Dumais, 2006; Shen et al., 2005a; Nunberg, 2003; Scholer et al., 2004). To be more effective, search engine systems should both be more adaptive (i.e., responsive to continuous changes in the contexts in which they are used), and more personalizable (i.e., to the individual preferences of individual users).

Designers should design systems that autonomously exploit a variety of different sources of data about contexts and users. Recently, there has been a growing interest in exploiting search logs to tailor IR systems adaptively for individual users. For example, search logs have been used to recommend closely associated query terms in query expansion (Anick, 2003; Huang, Chien, & Qyang, 2003), cluster related query terms to facilitate the retrieval process (Beeferman & Berger, 2000), identify users' search contexts (Ozmutlu & Cavdur, 2005), empirically create adaptive index terms for Web documents (Billerbeck et al., 2003; Ding & Zhou, 2007), and re-rank search results (Xue et al., 2002, 2004; Cui and Dekhtyar, 2005; Joachims, 2002; Weires, Schommer, & Kaufmann, 2008).

However, log-based approaches deserve further investigation. It is necessary to examine how an integrated system with different components exploiting search logs performs. Besides, most experiments testing log-based approaches are conducted in a batch mode. Few studies have investigated how real users respond in log-based systems.

This study proposes an innovative probabilistic approach exploiting search logs to improve retrieval performance. A proof-of-concept system, Unified Probabilistic Retrieval (UPIR), is implemented with a coordinator expert at the core, managing dependencies between multiple sub-systems, namely, retrieval agent, indexer agent, and feedback agent, in a principled and efficient way. Real users' queries and click-through data in search logs are exploited through different function modules such as query expansion, adaptive indexing, and ranking. A user-based experiment was conducted to examine system performance.

This paper is structured as follows. Section 2 provides a review of the related literature. Section 3 describes the unified probabilistic approach and the implementation of Unified Probabilistic Information Retrieval (UPIR). Section 4 presents details of the experimental design, including subjects, instruments,

Ma H.
IMPROVING WEB SEARCH BY EXPLOITING SEARCH LOGS.
DOI: 10.5220/0001843802080216
In *Proceedings of the Fifth International Conference on Web Information Systems and Technologies (WEBIST 2009)*, page
ISBN: 978-989-8111-81-4

experimental procedures, and discusses the results. Section 5 concludes this paper by presenting the research findings and future research directions.

## 2 BACKGROUND

In this section, I review major approaches exploiting search logs to improve search engine performance. The focus is on how different kinds of data in search logs are utilized in IR function modules such as query expansion, query clustering, context identification, adaptive indexing, and ranking and what the research results turn out to be. Studies that investigate the characteristics of queries and Web user behaviors or use search logs for evaluation purposes, though useful to gain insight into Web searching, do not apply search logs in information retrieval directly, and hence are not included.

Recent studies have demonstrated that incorporation of query expansion tools into full-scale Web search engines provides users with a useful tool to reformulate their queries. For example, Anick (2003) analyzes log sessions for two groups of users interacting with variants of the AltaVista search engine – a baseline group given no terminological feedback and a feedback group to whom twelve refinement terms are offered along with the search results. It is found that a subset of those users presented with terminological feedback make effective use of it on a continuing basis.

Cui and his colleagues propose a probabilistic method to extract correlations between query terms and document terms by analyzing query logs (Cui et al. 2002, 2003). Rather than analyzing terms in single queries, Huang, Chien, and Oyang (2003) have applied past queries to the problem of suggesting relevant search terms at the session level. Recently, Shen and fellow researchers (2005b) investigated how to infer a user's interest from the user's search context and use the inferred implicit user model for personalized searching.

Beeferman and Berger (2000) propose an innovative query clustering method based on "click-through data." Wen et al. (2001, 2002) describe a density-based clustering method that makes use of user logs to identify the documents the users have selected for a query.

Recently, obtaining contextual information on Web search engine logs has gained more attention among researchers. Shen et al. (2005a) propose several context sensitive retrieval algorithms to combine the preceding queries and clicked document summaries with the current query for better ranking of documents. Ozmutlu & Cavdur (2005) analyze contextual information in search engine query logs to enhance the understanding of Web users' search patterns, more specifically, topic changes within a search session.

Billerbeck et al. (2003) propose an adaptive indexing scheme by automatically selecting terms from past user queries that are associated with documents in the collection. Analyzing query logs at the session level, Zhou et al. (2006) develop a session-based adaptive indexing algorithm to improve the system performance by using Web server logs. Similarly, Ding & Zhou (2007) propose an adaptive indexing scheme using server log analysis to extract terms to build the Web page index. The log-based index is combined with the text-based and anchor-based indexes to provide a more complete view of the page content. Experiments have shown that this could improve the effectiveness of the Web site search significantly.

There are more and more researchers endeavoring to improve ranking by incorporating past search logs. For instance, Xue et al. (2002, 2004) propose a log mining model to improve the performance of site search. Cui and Dekhtyar (2005) propose the LPageRank algorithm for Web site search.

Some researchers build a meta search engine on top of a query-document matcher with re-ranking results based on search log analysis. For example, Joachims (2002) collects implicit measures in place of explicit measures, introducing a technique based entirely on click-through data to learn ranking functions. Hou et al. (2006) propose a framework of Feedback Search Engine (FSE), which not only analyzes the relevance between queries and Web pages but also uses click-through data to evaluate page-to-page relevance and re-generate content relevant search results. Tan et al. (2004) propose a Ranking SVM algorithm in a Co-training Framework (RSCF). Essentially, the RSCF algorithm takes the click-through data containing the items in the search result that have been clicked on by a user as an input, and generates adaptive rankers as an output.

Another use of search logs is to take them as training sets or resources for machine learning in ranking process. Zha et al. (2006) discuss

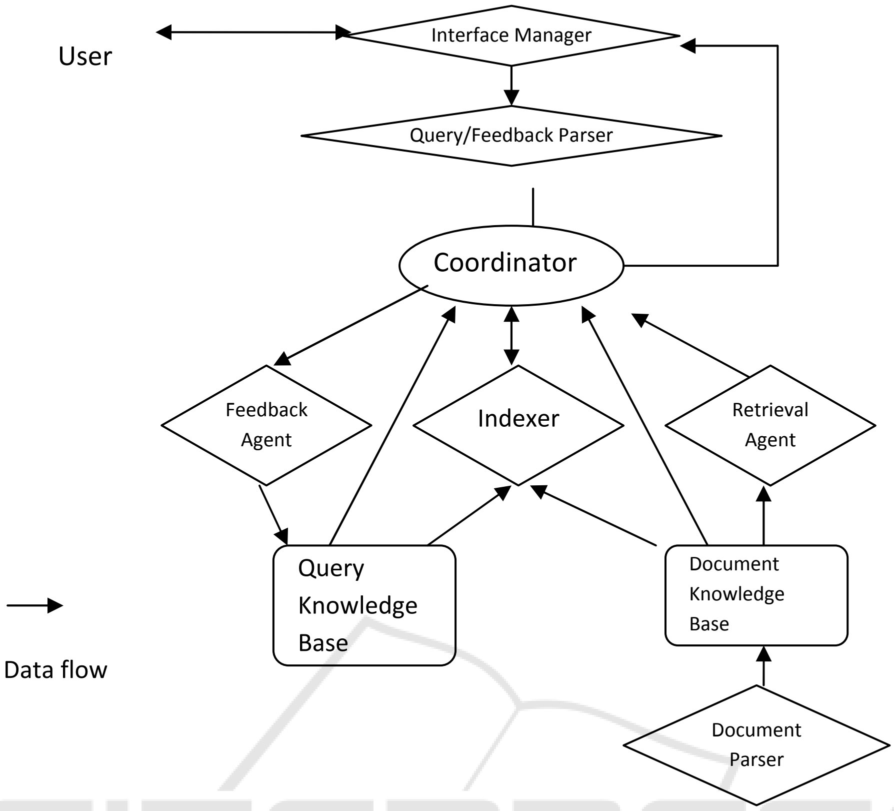

Figure 1: UPIR system architecture.

information retrieval methods that aim at serving a diverse stream of user queries such as those submitted to commercial search engines. Agichtein et al. (2006) show that incorporating user behavior data can significantly improve ordering of top results in real Web search settings.

## 3 PROBABILISTIC METHOD EXPLOITING SEARCH LOGS

This section proposes an integrated probabilistic system, UPIR (Unified Probabilistic Information Retrieval), based on an IR Coordination Model (Ma, 2008), which argues for utilizing cumulative evidence of past query observations and relevance judgments as well as transient relevance judgments from an individual user in the current retrieval iteration. One important assumption is that click-through data in search logs are users' implicit feedback, and hence the clicked documents are relevant to the query. This assumption may appear too strong. However, although the clicking information is not as accurate as explicit relevance judgments in the former case, the user' choice does suggest a certain degree of relevance. In fact, users usually do not make the choice randomly. In addition, we benefit from a large quantity of query logs. Even if some of the document clicks are erroneous, we can expect that most users do click on documents that are, or seem to be, relevant. Empirical studies that examine the reliability of implicit feedback generated from click-through data and query reformulations in web search strongly supports this assumption (Cui et al. 2002; Joachims et al., 2007).

Figure 1 depicts the UPIR system architecture. Each diamond is designed as an object/class. Boxes stand for knowledge bases. Arrows indicate data flow.

- The interface manager is a CGI program interacting with the UPIR system. It connects the users and the system.
- The query/feedback parser receives parameters from the interface, parse and stem query terms, and sends feedback to the coordinator expert.
- The document parser includes a snow-ball stemming algorithm and parses source documents into an inverted file, which is stored in the document knowledge base.

- There are two knowledge bases in the UPIR system, a document knowledge base and a query knowledge base. The document knowledge base keeps indexed Web pages. The query knowledge base stores past queries and implicit feedback from users.
- There are four major functional modules in the UPIR system. The feedback agent works to update the query knowledge base; the indexer agent works as a search agent processing past queries, the retrieval agent processes current queries; the coordinator manages dynamic data flow, initiates and schedules tasks for each agent, handles I/O requests and estimates retrieval status values.

The indexer agent estimates document term weight $W_{Dmi(t)}$ based on a group of past queries containing term ti:

$$\mathrm{W}_{\mathrm{Dmi}}(\mathrm{t}) = \log \frac{p_f(1-q_f)}{q_f(1-p_f)} \qquad (1)$$

$p_f$ = P($D_m$ judged as relevant |$Q_k$ contains term $t_i$), $p_f$ is the probability that document $D_m$ is judged as relevant by a user who submitted a query $Q_k$ containing term $t_i$.

$q_f$ = P($D_m$ judged as relevant |$Q_k$ does not contain term $t_i$), $q_f$ is the probability that document $D_m$ is judged as relevant by a user who submitted a query $Q_k$ without term $t_i$.

*1-$p_f$* = P($D_m$ judged as not relevant |$Q_k$ contains term $t_i$), 1-$p_f$ is the probability that document $D_m$ is judged as not relevant by a user who submitted a query $Q_k$ containing term $t_i$.

*1-$q_f$* = P($D_m$ judged as not relevant | $Q_k$ does not contain term $t_i$), 1-$q_f$ is the probability that document $D_m$ is judged as not relevant by a user who submitted a query $Q_k$ without term $t_i$.

For a particular document $D_m$, with a group of queries $n_q$ containing term $t_i$ which is a subset of query event space $N_q$ with relevance judgments, we have a contingency table as in table 1.

Table 1: Relevance contingency table.

| | **Relevant** | **Non-relevant** | |
|---|---|---|---|
| **$t_i$= 1** | $r_q$ | $n_q$ - $r_q$ | $\boldsymbol{n_q}$ |
| **$t_i$= 0** | $R_q$ -$r_q$ | $N_q$ -$n_q$ -$R_q$ + $r_q$ | $\boldsymbol{N_q}$ **-**$\boldsymbol{n_q}$ |
| | $\boldsymbol{R_q}$ | $\boldsymbol{N_q}$ **-**$\boldsymbol{R_q}$ | $\boldsymbol{N_q}$ |

$N_q$ denotes all the relevance judgments made by users submitting different queries. $R_q$ think $D_m$ is relevant. Of $n_q$ who submitted queries with term $t_i$, $r_q$ have judged a document as relevant. We get:

$p_f$ = $r_q$ / $R_q$
1- $p_f$ = ($R_q$ - $r_q$) / $R_q$
$q_f$ = ($n_q$ - $r_q$) / ($N_q$ - $R_q$)
1- $q_f$ = ($N_q$ - $R_q$ - $n_q$ + $r_q$) )/ ($N_q$ - $R_q$)

Connecting these estimations with formula (1), we get:

$$W_{Dmi}(t) = \log \frac{r_q(N_q - R_q - n_q + r_q)}{(R_q - r_q)(n_q - r_q)} \qquad (2)$$

We must have relevance feedback on the document $D_m$ and query observations with $t_i$. That is, for a document to be indexed with a term, there has to be at least one instance of relevance feedback from a user submitting a query containing term $t_i$ and judging document $D_m$ as relevant. Then we have: $R_q \neq 0, r_q \neq 0$. For estimation reasons, in the case of $N_q$ = $n_q$ (all query observations with relevance feedback in the query knowledge base are those contain query term $t_i$), or $r_q$ = $R_q$ (all relevance judgments on document $D_m$ are made by those queries containing $t_i$), or $n_q$ = $r_q$ (all users submitting query term $t_i$ judge document $D_m$ as relevant), we introduce a parameter *h* to adjust the weights in a similar way as in the BIR model (Robertson & Spärck Jones, 1976). Then we get the formula to estimate document term weights as

$$W_{Dmi}(t) = \log \frac{(r_q + h)(N_q - R_q - n_q + r_q + h)}{(R_q - r_q + h)(n_q - r_q + h)} \qquad (3)$$

The retrieval agent is a typical Binary Independent Retrieval search module. Its main function is to estimate, for each document containing query term $t_i$, the probabilistic BIR weights according to transient feedback regarding this query. The term weights are generated with:

$$\mathrm{W}_{\mathrm{Qki}}(t_i) = \log \frac{(r_d + h)(N_d - R_d - n_d + r_d + h)}{(R_d - r_d + h)(n_d - r_d + h)} \qquad (4)$$

where h=0.5

The coordinator is at the core of the UPIR system, estimating retrieval status values for each document with the formula (5):

$$W(D) = \sum_{t_i -> D_m \cap Q_k} \frac{(K+1)D_{mi}}{KL + D_{mi}} (\alpha W_{Qki}(t_i) + \beta W_{Dmi}(t_i)) \qquad (5)$$

where $W_{Qki}(t_i)$ is query-oriented term weights from the Retrieval Agent, $W_{Dmi}(t_i)$ is document-oriented term weights from the Indexer Agent, $\alpha$ and $\beta$ are parameters assigning credit to different term weights. K is some suitably chosen constant. L is the normalized document length. $D_{mi}$ is frequency of a particular term $t_i$ in a particular document $D_m$.

Formula (5) is derived from a simple, proven formula to weight documentx based on some term weight.

$$W(D) = \sum_{t_i \to D_m \cap Q_k} \frac{(K+1)D_{mi}}{KL + D_{mi}} W(t_i) \qquad (6)$$

This has been examined in many empirical studies in probabilistic retrieval (Spärck Jones, 2000; Robertson & Spärck Jones, 1997).

# 4 RESULTS AND DISCUSSION

## 4.1 Experiment Design

A thinking-aloud experiment (Newell & Simon, 1972) was conducted to compare search performances of the UPIR and a control system, Binary Independence Retrieval (BIR) system with BM-25 weights. A Latin-square within-subject experiment similar to the TREC Interactive Experimental Design was developed to remove the additive effects of searcher, topic, and task sequence.

Real users' queries and click-through data were collected from a Beta search engine, *Infocious* (search.infocious.com) January 31, 2005 - October. 10, 2005. There were 17,228 sample queries, with 2,611 queries with clicked through data. Totally, 2,185 Web documents were associated with 1522 unique queries. The most frequent query was submitted by 82 users. The most frequently accessed Web document was clicked by 26 users.

To solve the "sparsity problem," six topics for simulated tasks were selected from the top 50 queries ranked by the number of click-through data points related to a certain query in the *Infocious* search logs. Selection criteria include temporal effect, prerequisite knowledge, potential distress, and task complexity. That is, the selected topics should be interesting for the users in this study during the time frame when the experiments were conducted and when search logs were collected; the selected topics should not require any specialized or prerequisite knowledge from users; the selected topics should not arouse any potential distress for users; and the selected topics should contain both simple and complex subjects. Six selected topics were then further developed into simulated tasks to promote simulated information needs in users and position searches in a more realistic context (Borlund & Ingwersen, 1997).

The test Collection consists of 13,195 Web pages downloaded by the CPAN WWW::SEARCH module. For each selected topic for user study, the top 2000 Web pages from *Infocious* were downloaded during October 2005, which accounts for 12,000 documents in the database. Besides, 1,195 Web pages that had been clicked through in *Infocious* but not from the six selected topics were also downloaded.

Click-through data are taken as indicators of real users' implicit relevance feedback. Query terms and click-through Web pages were first parsed and stored in a B-tree table. After all the pages in the test collections were indexed, each clicked Web page was updated with its indexing terms by adding new associated terms from the search log.

This study's participants were recruited from undergraduate students at University of California, Los Angeles. Respondents were solicited by posting advertisements around the campus. Individuals were approached, and recruited if agreeable. Twenty-four participants came from various disciplinary areas.

## 4.2 System Performance

System performance was measured by instance recall and instance precision as used in TREC Interactive Evaluation. The searcher was instructed to look for instances of each topic. Two relevance assessors defined the instances from pooled search results from 24 subjects. Instance recall is defined as the proportion of true instances identified during a topic, while instance precision is defined as the number of documents with true instances identified divided by the number of documents saved by the user. It has been proven that instance recall and instance precision are more appropriate for real-user studies than traditional measures such as A-P and R-P (Turpin & Hersh, 2001). Instance recall and instance precision have also been widely used in the TREC Interactive Track since TREC 6, 1997.

### 4.2.1 Mann-Whitney Test

With instance recall and instance precision as performance measures, Mann-Whitney tests are performed to test the hypothesis H0 that there is no statistically significant difference in system performance between the UPIR and BIR systems. The Mann-Whitney test is preferred as it is nonparametric and does not require the assumption that instance recall and instance precision are normally distributed intervals. Besides, experimental results across topics and systems in the Latin-square design are not paired, which makes the Wilcoxon signed rank test not applicable in this analysis.

Table 2: Experiment 2 BIR and UPIR performance for simulated tasks 1-6. Performance measures are instance recall and instance precision. Results that show a significant difference from BIR using a one-tailed Mann-Whitney Test at the 0.05 and 0.10 levels are indicated by ** and *, respectively. The last row in the table shows the percentage of performance improvement. Cells are highlighted when UPIR outperforms BIR significantly.

| | *Instance Recall* | *Instance Precision* |
|---|---|---|
| BIR system | 0.4153 | 0.7143 |
| UPIR system | 0.4772** | 0.7656 |
| % improvement | +14.90% | +7.18% |

Table 2 presents the Mann-Whitney test results and performance improvement percentages in the experiment. The unit of analysis is instance recall and instance precision for one topic in one system and by one user. There are 144 observations in total, and 72 for each system.

The Mann-Whitney test results show that UPIR performs significantly better than BIR at a significance level of $p \leq 0.050$, when instance recall is applied. Instance recall is improved by 14.90%. The null hypothesis H0 can be rejected. However, even though instance precision is improved by 7.18%, the Mann-Whitney tests on instance precision are not statistically significant, and the null hypothesis H0 cannot be rejected.

The possible reason for this is that instance recall and instance precision are based on real users' judgments after each user has personally browsed the short description of each result in the Searcher Worksheet and/or checked the Web documents. It is more likely that the search results are correct and hence the high precision results from both the BIR and UPIR searches. However, since there is a time limit for each task, users do not have much time to check many records. Therefore, each user might choose to check a few records after browsing the short descriptions. This kind of choice is subjective, which leads to different final result sets from each user. Therefore, instance recall for the UPIR and BIR systems by topics could vary at a higher rate than instance precision, thereby showing different statistical significance test results.

The Mann-Whitney tests compare the overall system performance of BIR and UPIR in the experimental settings. In the following sections, I analyze instance recall and instance precision by tasks to see how the two systems perform on different tasks. Since there are only 12 observations for each task by systems, no statistical significance tests are conducted with the small samples. Instead, improvement percentages are presented for system comparison.

### 4.2.2 Instance Recall by Tasks

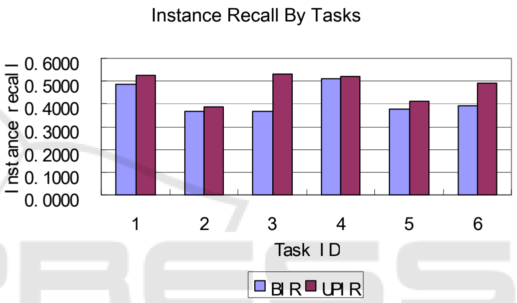

Figure 2: Experiment 2 instance recall by tasks.

Table 3: Experiment 2 instance recall and improvement percentage by tasks.

| *Task* | *Instances* | *BIR* | *UPIR* | *% improvement* |
|---|---|---|---|---|
| 1 | 26 | 0.4873 | 0.5272 | 8.19% |
| 2 | 32 | 0.3653 | 0.3848 | 5.34% |
| 3 | 18 | 0.3646 | 0.5285 | 44.95% |
| 4 | 24 | 0.5089 | 0.5188 | 1.95% |
| 5 | 16 | 0.3762 | 0.4124 | 9.62% |
| 6 | 9 | 0.3895 | 0.4915 | 26.19% |
| Average | 21 | 0.4153 | 0.4772 | 14.90% |

Figure 2 shows that UPIR achieves higher instance recall on all of the six tasks, especially on task 3 (earthquakes that have caused property damage and or loss of life) and task 6 (locations of companies providing private police services). However, UPIR does not demonstrate noticeably better performance than BIR on some tasks. For example, instance recall of UPIR and BIR for task 2 (chess training software and opening theory) and task 4 (houseplant selection) are almost the same.

Table 4: Instance precision and improvement percentage by tasks.

| *Task* | *Instances* | *BIR* | *UPIR* | *% improvement* |
|---|---|---|---|---|
| 1 | 26 | 0.6245 | 0.7273 | 16.46% |
| 2 | 32 | 0.6635 | 0.7184 | 8.27% |
| 3 | 18 | 0.7219 | 0.7285 | 0.91% |
| 4 | 24 | 0.7182 | 0.8162 | 13.65% |
| 5 | 16 | 0.8095 | 0.8133 | 0.47% |
| 6 | 9 | 0.7482 | 0.7896 | 5.53% |
| Average | 21 | 0.7143 | 0.7656 | 7.17% |

These findings are confirmed by the results in Table 3 UPIR improves instance recall over BIR from 1.95% (task 4) to 44.95% (task 3). The average improvement percentage is 14.90%. The number of corrected instances identified from users' judgment pools varies from 9 to 32. Spearman's rank correlation coefficients are calculated for each pair of the number of instances and improvement percentage. In the six cases, no correlations observed are found to be significant at the level of p=0.10. This suggests that UPIR performs better on certain topics than others and such performance differences are not associated with the number of instances.

### 4.2.3 Instance Precision by Tasks

Though the Mann-Whitney test results show that there are no statistically significant differences between instance precision of UPIR and BIR, it is still worthwhile to examine how instance precision changes on different tasks, as it presents another view of UPIR and BIR system performance.

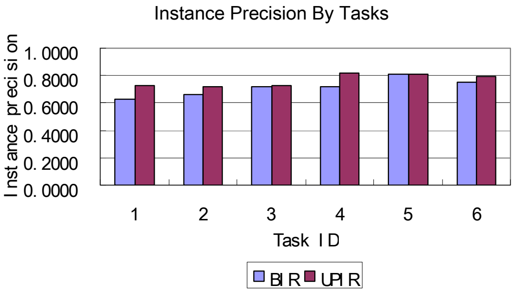

Figure 3: Experiment 2 instance precision by tasks.

Figure 3 shows that UPIR improves instance precision slightly. Table 4 gives more specific data that such improvement ranges from 0.47% (task 5) to 16.46% (task 1). The mean improvement percentage is 7.17%. In general, improvement of instance precision is much smaller than that of instance recall in comparison of mean, minimum, and maximum. It is interesting that the patterns of improvement percentage for instance precision and instance recall are different by tasks. UPIR achieves the largest improvement percentage of instance recall on task 3 (44.95%) and that of instance precision on task 1 (16.46%).

The values of Spearman's rank correlation coefficients are calculated for each pair of the number of instances and improvement percentage of instance precision. At the level of p=0.10, the results of the analysis are non-significant in every case. Similarly, this suggests that UPIR performs better on certain topics than others by the instance precision measure and such performance differences are not associated with the number of instances.

## 5 CONCLUSIONS AND FUTURE WORK

Statistically speaking, log-based UPIR can significantly improve system performance over the baseline system, BIR, in the user-centered Experiment. However, such improvement may vary with different topics, according to the observations on instance recall and instance precision by tasks.

The empirical results further suggest that instance recall is more appropriate to compare system performance than instance precision, since instance recall is of more distinguishing power than instance precision in system comparison. For example, the Mann-Whitney test results show that UPIR performed significantly better than BIR at a significance level of p≤0.050, when instance recall was applied. Instance recall was improved by 14.90%. However, even though instance precision was improved by 7.18%, the Mann-Whitney tests on instance precision were not statistically significant.

This study has taken a first step in implementing a probabilistic method exploiting search logs and validating it empirically. Further studies along this line, such as performance variance on different tasks, will add dimension to the present study and promote successful information retrieval on the Web. With the increasing importance of improving search engine performance, it is imperative that researchers interested in system design as well as user studies take seriously the recommendations discussed above and provide opportunities to improve end-user searching, and search engine effectiveness.

## ACKNOWLEDGEMENTS

Many thanks to Dr. Karen Spärck Jones for helping to shape my original conceptual design of unified probabilistic retrieval. I am grateful to Professor Junghoo Cho and Dr. Alexandros Ntoulas for providing their help and the resources for the experiment. Thanks also to professors Jonathan Furner, Gregory H. Leazer, Christine Borgman, Mark H. Hansen, and Kathleen Burnett for their valuable feedback. This research was supported by Dissertation Fellowship, University of California, Los Angeles, and First Year Assistant Professor Award (FYAP), Florida State University.

## REFERENCES

Agichtein, E., Brill E., & Dumais, S.T. (2006). Improving Web search ranking by incorporating user behavior information. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 19-26). Washington: ACM.

Anick, P. (2003). Using terminological feedback for Web search refinement - a log-based study. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 88-95). New York: ACM.

Beeferman, D., & Berger, A. (2000). Agglomerative clustering of a search engine query log. In *Proceeding of International ACM SIGKDD Conference on Knowledge (ACM SIGKDD"00)* (pp. 407-416). Boston: ACM.

Billerbeck, B., Scholer, F. Williams, H.E, & Zobel, J. (2003). Query expansion using Associated Queries. In Proceedings of *The Twelfth International Conference on Information and Knowledge Management.* (pp. 2-9). New York: ACM.

Borlund, P., & Ingwersen, P. (1997). The development of a method for the evaluation of interactive information retrieval systems. *Journal of Documentation.* 53(3):225-25.

Burns, E. (2007). *U.S. search engine rankings.* Retrieved August 6, 2008, from www.searchenginewatch.com

Cui, Q., & Dekhtyar, A. (2005), On Improving Local Website Search Using Web Server Traffic Logs: A Preliminary Report, In *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management*, (pp.59-66). New York: ACM

Ding, C., & Zhou, J. (2007). Log-based indexing to improve Web site search. In *Proceedings of the 2007 ACM symposium on Applied computing* (pp 829-833). New York:ACM.

Hou, Y., Zhu, H., & He, P. (2006). A framework of feedback search engine motivated by content relevance mining. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence* (pp.749-752). New York: ACM.

Huang, C. Chien, L. & Oyang, Y. (2003). 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.

Joachims, T. (2002). Optimizing search engines using click-through data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 133-142). New York: ACM.

Ma, H. (2008). User-System Coordination in Unified Probabilistic Retrieval: Exploiting Search Logs to Construct Common Ground. Unpublished Doctoral Dissertation. University of California, Los Angels.

Nunberg, G. (2003). As google goes, so goes the nation. *New York Times*, May 18, 2003.

Ozmutlu, H., Cavdur, F. (2005), Application of automatic topic identification on excite web search engine data logs, *Information Processing and Management*, 41,1243-62.

Robertson, S. E. & Spärck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science*, *27*, 129-146.

Robertson, S. E. & Spärck Jones, K. (1997). Simple, proven approaches to text retrieval. Technical Report. University of Cambridge Computer Laboratory.

Scholer, F. Williams, H. & Turpin, A. (2004). Query association surrogates for Web search. *Journal of the American Society for Information Science and Technology, 55(7),* 637-650.

Shen, X., Tan, B., & Zhai, C. (2005a). Context-sensitive information retrieval using implicit feedback, In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieva*l, Salvador, Brazil:ACM

Shen, X., Tan, B., & Zhai, C. (2005b). Implicit user modeling for personalized search. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*. (pp. 824 – 831). New York: ACM.

Spärck Jones, K., & Willett, P. (1997). *Overall introduction*. In K. Spärck Jones and P. Willett (Eds.), *Readings in Information Retrieval* (pp 1 - 7), San

Franciso: Morgan Kaufmann Public.

Spärck Jones, K., Walker, S., & Robertson, S. E. (2000). A probabilistic model of information retrieval: development and comparative experiment. *Information Processing and Management*, *36,* 779-808.

Tan,Q., Chai, X., Ng, W., &Lee D.L. (2004). Applying co-training to clickthrough data for search engine adaptation. In *Proceedings of the 9th International Conference on Database Systems for Advanced Applications (DASFAA)*. New York: ACM.

Turpin, A. H., & Hersh, W. (2001). Why batch and user evaluations do not give the same results. In *Proceedings of the 24th Annual international ACM SIGIR Conference on Research and Development in information Retrieval*. (pp. 225-231). ACM: New York.

Xue,G. Zeng, H., Chen, Z., Ma, W., Zhang, H., & Lu, C. W. (2002). Log Mining to Improve the Performance of Site Search. In *Proceedings of the Third International Conference on Web Information Systems Engineering (Workshops) - (WISEw'02)*. (pp. 238). New York: ACM.

Xue, G. Zeng, H., Cheng,Z. Yu, Y., Ma, W., Xi, W., & Fan, G. (2004) Optimizing Web search using Web click-through data. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management.* (pp.118-126). New York: ACM.

Wen, J. R., Nie. J., & Zhang, H. (2001). Clustering user queries of a search engine. In *Proceedings of the 10th International World Wide Web Conference(WWW"01)*, (pp. 162–168). New York: ACM.

Wen, J., Nie, J., & Zhang, H. (2002) Query clustering using user logs. *ACM Transactions on Information Systems*. *20(1)*. 59-81.

Weires, R., Schommer, C., & Kaufmann S. (2008). SEREBIF - Search Engine Result Enhancement by Implicit Feedback. *4th Intl Conference on Web Information Systems and Technologies (WebIst).* Funchal, Madeira. May 2008.

Zha, H., Zheng, Z., Fu, H., & Sun, G. (2006). Incorporating query difference for learning retrieval functions in World Wide Web search. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management* (pp. 307-316). New York: ACM.

Zhou, J., Ding, C., & Androutsos, D. (2006). Improving Web site search using Web sever logs. In *Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research* (pp.35-48). New York: ACM.