Improving Web Page Retrieval using Search Context from Clicked Domain Names

Rongmei Li
School of Electrical, Mathematics, and Computer Science
University of Twente
P.O.Box 217, 7500 AE, Enschede, the Netherlands
lir@cs.utwente.nl

***Abstract*—Search context is a crucial factor that helps to understand a user's information need in ad-hoc Web page retrieval. A query log of a search engine contains rich information on issued queries and their corresponding clicked Web pages. The clicked data implies its relevance to the query and can be used to define the topical context. However, the log is usually not completely available due to privacy concerns. In this paper, we derive clicked pages from clicked domains and use the surrounding query context to enhance retrieval performance. One strategy is to promote clicked pages directly in the initial retrieval result. Another strategy is to expand the original query using selected terms from the clicked pages. Our experimental results on the TREC GOV2 data and a query log of a major search engine show that both strategies can boost retrieval performance compared to the standard language model and pseudo relevance feedback (PRF) model. Their good performance on early precision allows us to apply PRF further for even more accurate result that is comparable to the performance of true relevance feedback.**

I. Introduction

Search context is a crucial factor that helps to understand a user's information need in ad-hoc Web page retrieval. In a clear context, the semantic meaning of a search query is disambiguated so that a search engine can focus its retrieval to the user's interest. A query log of search engines contains rich information on users' search history, such as [2]: 1) query terms; 2) retrieved documents; 3) clicked documents; 4) document ranks; 5) date and time of the search action; 6) an anonymous identifier for each session. The sequence of activities provides prior knowledge on search context and on language usage of users for a specific domain. This is very meaningful in the setting of ad-hoc Web retrieval, where a search query is often very short and the user and his interesting Web pages may use different vocabularies for the same topic.

Commercial search engines have been using large query logs to improve their search service as collecting such data is relatively easier for them compared to the research community. However, a detailed query log contains sensitive personal information that can be used to identify a particular user by tracing his search activities. Due to this risk, more and more users are reluctant to allow search engines to record their detailed search activities. In response to the user's privacy concerns, search engines anonymize user IDs and degrade some search information, for instance, shorten the clicked URLs to their domain names. But with the degraded data, can search engines still provide good service?

In this paper, we work on a degraded query log, specifically, the case where the click-through data contains only the domain portion of the clicked URL and the user IDs are anonymized. We address two main research questions:

- Can we extract the common topical context from such data for a query?
- Can we use this knowledge to improve retrieval performance of Web pages effectively?

We assume that every click is implicitly relevant to its corresponding query in the log and treat this as our relevance evidence. We adopt three strategies to integrate query and click-through information with the language modeling (LM) framework. One is to promote the clicked Web pages from the initial retrieval result in re-ranking. Another is to extract informative terms from clicked/implicitly relevant Web pages to expand the original query. The relevant pages are restored from the clicked domain names at three levels, namely, the site's home page, the server name, and the domain name. Web pages are then ranked by the cross-entropy score between smoothed documents and the improved query model. In the third, we use the PRF technique to re-rank the previous results.

The rest of the paper is organized as follows: In section II, we review current works using a query log to improve Web retrieval and similar techniques exploiting the topical context. We present language models in section III. Our strategies in the LM framework are presented in section IV. Section V discusses our experiments. Finally we draw our conclusion in section VI.

II. Related Work

Query log analysis has recently been focused on studying user search behavior for improving retrieval result. Such works in literature can be roughly categorized as query-centered and user-centered modeling. Our work is similar to the query-centered ranking that aims at presenting a clear explanation of the query intent for each information request. The main approaches are characterized by vector space models [3], machine learning [1], [12], and LM [14]–[16].

In the LM framework, Shen et al. used query log information as follows: 1) combine query results such that a document could be favored if it has been ranked high by all previous queries; and 2) combine query models in such a way that the context-based query model was the average of all past query

models (see [15]); 3) smooth the original query model with contextual information modeled from the previously issued queries and the clicked document summaries (see [14]). The model 3) was adapted by a decaying weight during interactive search. Another similar work [16] tried to interpolate the query model with history models linearly. The history model was derived from long-term history including past queries, all retrieved documents, and clicked documents. Their experimental results [16] showed that the best improvement was reached when the clicked documents alone were used.

To handle query ambiguity, query expansion (QE) has been broadly adopted in traditional information retrieval (IR). One of its focuses is the expansion terms. True relevance feedback (TRF) [13] extracts those terms from relevant documents explicitly judged by users. Alternatively PRF utilizes the top-ranked documents of the initial result. Recently the query log approach provides the third source, the top-ranked and implicitly judged documents. In a log-based work [3], expansion terms were extracted from the clicked documents according to probabilistic correlations between query terms and those document terms. The experimental result demonstrated that it outperformed PRF. Similarly, we adopt QE and PRF for the log-based approach in this work. Different from them, we apply the Expectation-Maximization (EM) algorithm to select terms from clicked pages besides the maximum likelihood (ML) estimate. The clicked pages are derived from the clicked domain names. Moreover, we integrate the additional query information with the LM framework.

A query log can help re-ranking the retrieval result directly. In [19], the query context was constructed from log queries containing the original query terms and/or their neighboring queries in the same session. The ranking scores were recomputed for top-ranked documents by merging this information with the original rank. Another work [9] refined the query by log queries containing or being associated with the original. The ranking results of the original and N query refinements were merged as follows: 1) combine the top-k rank of all results; 2) insert each result of N query refinements to the original rank at every k-th position; 3) remove the documents in the original rank if they were clicked less frequently. In a machine learning approach [1], a new rank learned from user interaction features was interpolated with the corresponding original rank of each document.

To our best knowledge, there is still no benchmark test data for evaluating log-based approaches. Works in literature had their own test base as follows: 1) use their own system to obtain retrieval results, relevance judgments, and query log [16]; 2) use the available log and hand-made relevance [1], [3], [9], [19]; 3) use standard TREC data collection to recreate query log [14], [15]. For the last, the standard TREC judgments can be used. In this work we propose a different evaluation method where the standard TREC GOV2 data and a query log of a major search engine are used. We map the log queries to the GOV2 queries so that the query-related information in that log can be transferred to TREC data. Thus, we are able to use the available judgments to test our models.

III. Language Models

The "language model" is a probabilistic model for language generation developed for automatic speech recognition systems in the early 1980s. Introduced by Ponte and Croft [11] for IR in 1998, it showed good empirical performance [18]. Motivated by its relative simplicity, effectiveness, and flexibility, we adopt the LM approach in this work.

A serious problem in the LM approach is that a relevant document will not be retrieved if one query term is not found in that document. The fundamental solution is the smoothing technique that assigns a non-zero probability to that missing term. In the IR setting, the Jelinek-Mercer (JM) smoothing [17] is commonly used. The JM-based model is to linearly interpolate the document model with a general background model, using a coefficient λ to control the influence of each. This mixture model can be formulated as follows:

$$P(t|D) = \lambda P_{ml}(t|D) + (1-\lambda)P_{ml}(t|C) \quad (1)$$

where $P(t|D)$ is the probability of generating a query term t from a given document D. The document model $P_{ml}(t|D)$ and collection model $P_{ml}(t|C)$ can be computed by the ML estimate that is the fraction of term frequency in that document D or collection C.

Empirically the JM model performs worse for title queries but better for long verbose queries [17]. It explains common and non-informative terms well in a query. In this work, we use this method to smooth our document model.

A. Parsimonious Models

The standard language model tends to estimate the probability for every term in a document including general terms that often occur in the whole collection. However, these terms are less discriminative and contribute less for distinguishing a relevant document from others. To eliminate these terms from models, Hiemstra et al. [5] introduced the so-called parsimonious language model. This model uses the EM algorithm to estimate the term probabilities in a document. The algorithm is realized by the following two steps:

$$\text{E-step}: e_t = tf(t,D) \cdot \frac{\alpha P(t|D)}{\alpha P(t|D) + (1-\alpha)P_{ml}(t|C)}$$

$$\text{M-step}: P(t|D) = \frac{e_t}{\sum_t e_t}, \text{i.e. normalize the model}$$

where $tf(t,D)$ is the frequency of a term t in a document D, $P(t|D)$ is the ML of term t in D at the first run and the result of the M-step for the rest of runs, and α is the weight factor.

At the E-step, the expectation score e_t is computed for all terms in a document. The general terms have smaller expectation scores as they have relatively higher probabilities $P_{ml}(t|C)$ in the background model compared to their probabilities $P(t|D)$ in a document. At the M-step, the expectation score is normalized and compared to a given pruning factor. Some general terms do not pass the pruning test as their normalized expectation score is low. This selection process continues till the maximized term probability does not change

significantly anymore. This learning process requires no information from a user query or relevance judgment.

B. Cross-entropy Score

Similarly as the document model, a query model can be computed by the ML estimate. The ranking score of a Web page is measured by the difference between the query model and the smoothed document model. Inspired by the relevance model [5], [7], [8], we use cross-entropy score to quantify the information gain between two models. The score is high when two models differ from each other. Otherwise, it is low. This divergence computation is not symmetric. The cross-entropy score is formulated as follows:

$$score(D) = \sum_{i=1}^{l} [P_{ml}(t_i|Q) \cdot \log(P(t_i|D))] \quad (2)$$

where l is the length of the whole language vocabulary, $P_{ml}(t_i|Q)$ is the ML query model, and $P(t_i|D)$ is the smoothed document model.

IV. Query Log Modeling

A query log containing information on search activities of a group of users provides us collective knowledge on search context for similar queries, particularly the semantic meaning of the query. We thus understand the search intent of users better in case of ad-hoc Web retrieval. For instance, given the query "murals" we can find the corresponding clicked domains in our query log. Although these domains are noisy, they still imply the relevance of their contents to this query. A user study [6] on interpreting click-through data proves that the implicit feedback generated from clicks shows reasonable agreement with the explicit judgments on Web pages.

A. Domain Names to Web Pages

To derive the relevance information from a degraded query log, where the click-throughs are the domain names (e.g. "bcn.net"), we need to reconstruct the original URLs that bring forth Web pages. We carry out the restoration process of "domain names" to "Web pages" at three different levels as follows:

- **domain level**: identify Web pages with the same domain name of the clicks. For instance, the clicked "http://www.cancer.gov" can be mapped to "http://seer.cancer.gov/..." as they share the same domain name of "cancer.gov".
- **server level**: identify Web pages having the same phrase between "http://" and the next "/" in their URL as the clicks. For instance, the clicked "http://www.cancer.gov" can be mapped to "http://www.cancer.gov/...".
- **URL level**: identify Web pages having the same URL of the clicks only (e.g. the page of "http://www.cancer.gov").

B. Our Strategies

We hypothesize that Web pages clicked previously by users are still relevant to the same query when it is issued again in the near future. Using this as an independent evidence, we formulate our first strategy as follows:

Strategy 1: Promote the clicked Web pages to the top of the ranking list next time when the same query is given again.

This strategy is realized as follows: 1) obtain the top-k pages from an initial result; 2) add a value to the ranking score of the clicked Web pages. As a result, the original document rank is preserved while the most interesting ones are promoted to the top.

Strategy 1 takes URLs as the sole relevance information and does not address the following problems: 1) the content of Web pages may change; 2) the original click-throughs and restored pages may be irrelevant or have poor quality in content. For more accurate topical context, one should consider their contents, specially, when the aggregated information at the same domain can be relatively stable. Following this idea, we formulate our second strategy as follows:

Strategy 2: Expand the original query model by K selected terms from the clicked pages and rank pages by new cross-entropy score. The improved query model is given as follows:

$$P(t|Q) = \beta P_{ml}(t|Q) + (1-\beta)P(t|L) \quad (3)$$

where β is the weight factor and $P(t|L)$ is the model of clicked pages.

The model of clicked pages is computed as follows: 1) for each query, combine all restored Web pages; 2) compute the parsimonious model using the EM or ML estimate; 3) select top K terms with highest probability for QE.

As the model of clicked pages carries more contextual and semantic information, we expect the expanded query model explains the user's information needs better. Consequently, more relevant Web pages can be biased to the top of the ranking list. With such consideration, we apply PRF further as follows:

Strategy 3: Expand the original query model by N selected terms from the top 10 Web pages of the result of strategy 2 and rank pages by their cross-entropy score. The expanded query model is given by:

$$P(t|Q) = \beta P_{ml}(t|Q) + (1-\beta)P(t|F) \quad (4)$$

where $P(t|F)$ is the feedback model.

The top 10 pages are an empirical choice. The parsimonious and ML models of these 10 pages are estimated similarly as in strategy 2.

V. Experiments

We elaborate our hypothesis and strategies in section IV. In this section, we verify our ideas by experiments. We use the Indri search engine as our tool for standard language models and our own C++ code for parsimonious (or ML) models. We use **standard TREC performance metrics** to evaluate retrieval results including Mean Average Precision (MAP), Binary PREFerence (BPREF), Precision at 10 (P@10) or 20 (P@20), and Reciprocal Rank of top Relevant document (MRR). For determining **statistical significance** of MAP differences between baselines and strategy runs, we do a two-tailed paired T-test at the 0.5 level and report on significant improvements by a bullet (•). The best results of each strategy are bold-faced in tables (Table I to Table III).

A. Query Log Statistics

Our query log was collected from 01 March, 2006 to 31 May, 2006 by a major search engine. It contains 10,154,742 unique queries and 19,442,629 click-throughs with only the domain names of the clicked pages. We do not use the user IDs in this work (for more information, see [10]).

B. Test Data

Our test collection is the TREC GOV2 data. It contains 25,205,179 Web pages with 666 terms on average. It is indexed with the "Porter" stemmer and the standard Lemur stopword list. The collection has 150 topics in which only the title field is used. We assume the GOV2 title queries are the same as those found in our query log. We then use corresponding judgments for evaluation. In total, 29 such queries are identified.

C. Domain Names to Web Pages

As explained in the subsection IV-A, we need to restore the URLs of clicked pages from the clicked domains. In practice, we identify those URLs from the GOV2 collection for strategy 1 as it contains pages with the .GOV domain name only. For strategy 2 and 3, this restriction does not stand. For QE, we assume that pages on the current Web are similar to the clicked and the open directory (DMOZ [4]) provides popular pages. As an experimental result, 243 clicked domains are found in our log for 29 queries and are mapped to 103,911 DMOZ URLs. 11 out of 29 common queries have restored URLs from GOV2 for strategy 1 and from DMOZ for strategy 2.

D. Model Settings

We take $\lambda = 0.9$ for the JM smoothing. In EM-algorithm α is 0.9 and the pruning factor is 0.001 for the parsimonious model. In QE, the weight β is 0.5 for all models. The numbers of expansion terms are 20, 50, or all for strategy 2 and 50 for strategy 3.

E. Experimental Results

We conduct experiments to evaluate our ideas at three different levels. We further categorize the clicked domains into the .GOV domain and any domain corresponding to 11 and 29 common queries and different sources for restoration (see subsection V-C). Our experimental results are presented and discussed in the following sub-sections.

1) Re-ranking (.GOV domain): Strategy 1 is only applicable for 11 queries as explained above. To bring the clicked pages up in the rank, we add a value c to their ranking score. The c value is defined as the maximum value of the absolute scores of all considered documents. Here, we take the top 1,000 documents for re-ranking as a common choice. The retrieval results at three levels are summarized in Table I.

In general, the retrieval is improved based on BPREF and P@10. The main exception is at the server and domain level in MAP. One reason may be that the restoration of the clicked URLs at these levels generates much more URLs than the relevant. Another reason is that unjudged documents in the collection are considered non-relevant in evaluation. When the restored URLs fall into the unjudged category, re-ranking is promoting the non-relevant pages to the top. This operation hurts the performance in MAP. In such a case, we should use the BPREF metric as it only considers the judged documents.

TABLE I
RE-RANKING RESULTS (.GOV DOMAIN)

models/levels	performance metrics				
	MAP	BPREF	P@10	P@20	MRR
baseline (JM)	0.3294	0.4625	0.4455	0.4864	0.5357
url	0.3304	0.4625	0.4636	0.4909	0.6742
server	0.3215	**0.4814**	0.5182	0.4455	**0.8333**
domain	0.3105	0.4768	**0.5364**	**0.5000**	0.7309

TABLE II
QE RESULTS (.GOV DOMAIN)

models/level.terms	performance metrics			
	MAP	P@10	P@20	MRR
baseline (JM)	0.3294	0.4455	0.4864	0.5357
url.ML.50	0.3484	0.5273	0.5045	0.6440
server.ML.50	0.3729	0.6091	0.5818	0.6924
domain.ML.50	0.3714	0.5909	0.5682	0.6318
url.EM.50	0.3509	0.5273	0.5227	0.6440
server.EM.50	**0.3849** •	**0.6273**	**0.6045**	**0.7455**
domain.EM.50	0.3842 •	0.6091	0.5909	0.6848

Another performance factor is the number of documents for re-ranking. The result shows consistent precision increment at 10 but fluctuation at 20 when compared to the baseline. This finding suggests the portions of clicked relevant and non-relevant documents are different for three mapping results. So are the distributions in the ranking list. Therefore, the cut at 1,000 documents is not the optimal choice. However, our preliminary result is still promising for the standard choice.

2) Query Expansion (.GOV domain): Table II shows the results of QE on the restored .GOV pages. Clearly, all expanded JM models outperform the JM baseline and the EM estimate is more effective than the ML estimate. The best improvement is reached at the server level.

3) Pseudo Relevance Feedback (.GOV domain): The experimental results of PRF on the restored .GOV pages are presented in Table III. All runs win over the standard PRF and the result of strategy 2. The best result is achieved at the domain level in MAP and at the server level in P@10.

4) Discussion on any Domain: For the same 11 queries, we extend our experiments of QE and PRF on all restored pages including those with a non-gov domain. All runs show performance gain over baselines. But they do not win over

TABLE III
PRF RESULTS (.GOV DOMAIN)

models/level.terms	performance metrics			
	MAP	P@10	P@20	MRR
baseline (JM-PRF)	0.3470	0.4909	0.5227	0.4819
url.ML.50-50	0.3821	0.5818	0.5818	0.6848
server.ML.50-50	0.4226 •	**0.6818**	**0.6682**	0.6948
domain.ML.50-50	0.4232 •	0.6455	0.6591	0.6948
url.EM.50-50	0.3959	0.5727	0.5773	0.6924
server.EM.50-50	0.4386 •	0.6727	0.6500	0.8409
domain.EM.50-50	**0.4391** •	0.6455	0.6409	**0.8409**

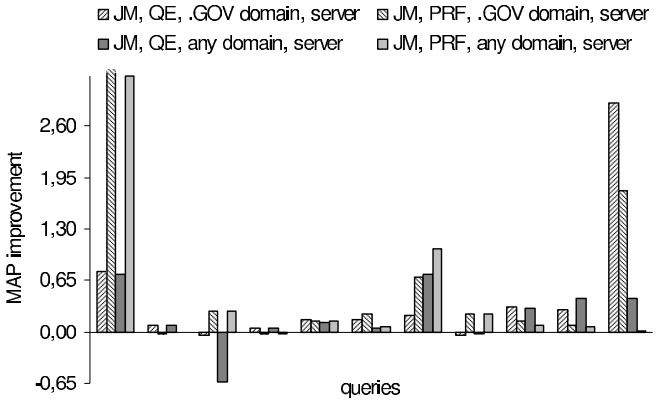

Fig. 1. Per-query MAP improvement against corresponding baselines for strategie 2 and 3 with the EM estimate at the server level (11 common queries)

TABLE IV
TRF RESULTS (.GOV DOMAIN)

models	11 common log and GOV2 queries			
	MAP	P@10	P@20	MRR
baseline (JM-TRF)	0.4734	0.8455	0.7591	1.0000
29 common log and GOV2 queries				
baseline (JM-TRF)	0.3366	0.7034	0.5983	0.9015

their ".gov" counterparts. An exception is that they outperform the runs for the .GOV domain for both estimates at the URL level. Query-wise, PRF shows effectiveness even when QE fails to improve the original result. A detailed comparison per query at the server level is visualized in Figure 1.

We repeat QE and PRF experiments on 29 queries for the restored pages at any domain. All results are better than their baselines. For QE, the best run is at the URL level. For PRF, it is at the server level. The results also show that the accuracy increases with the number of expansion terms.

Our results prove the effectiveness of using the implicit relevance feedback for retrieval. Is their performance comparable to that of explicit TRF? To answer this question, we simulate TRF by using judged documents (TREC qrels) as follows: take the top 10 judged relevant pages from the initial result to construct the parsimonious model and select the top 50 terms from it to expand the original query. Compared this results (see Table IV) with our best results by MAP, the performance differences are **7.25%**, **23.65%** for 11 queries; **10.28%**, **26.47%** for 29 queries in MAP and P@10 respectively. This evidence strongly suggests that our strategies are promising for effective retrieval for such degraded data.

VI. CONCLUSION

This paper aims at improving performance of ad-hoc Web retrieval using the topical context. A detailed query log provides us such source, specifically, the clicked Web pages. In our setting, we have a degraded query log where only the domain name of the clicks is known. In this work, we try to restore the clicked URLs from the clicked domains using the GOV2 collection for pages with the .GOV domain and DMOZ as the collection of popular Web pages. The restoration process is carried out at three levels, namely, the page URL, the server, and the domain levels. We then derive search context from the restored Web pages as they present the common interest of topic for a group of users. We present three strategies to integrate the query and click-through information with the LM framework. One is to bias the clicked pages to the top of the ranking list. Another is to expand the original query by selected terms from the clicked pages. The last applies PRF to re-rank the results of strategy 2. Compared to the state-of-the-art models, namely the standard language model and PRF model, our experiments on the TREC GOV2 collection and a query log of a major search engine demonstrate more accurate retrieval results. The results also suggest they can be comparable to the TRF model.

ACKNOWLEDGMENT

The author is grateful to Djoerd Hiemstra and Peter M.G. Apers for their valuable suggestion and discussion. This work is sponsored by the Netherlands Organization for Scientific Research (NWO) under project number 612-066-513.

REFERENCES

[1] E. Agichtein, E. Brill and S. Dumais. Improving Web Search Ranking by Incorporating User Behavior Information, In *Proceedings of SIGIR*, pp. 9–26, (2006)
[2] C. Castillo, C. Corsi, and D. Donato. Query-log Mining for Detecting Spam, In *Proceedings of AIRWeb*, pp. 17–20, (2008)
[3] H. Cui, J.R. Wen, J.Y. Nie, and W.Y. Ma. Probabilistic Query Expansion Using Query Logs, In *Proceedings of WWW*, pp. 325-332, (2002)
[4] DMOZ. Open Directory Project, *http://dmoz.org/*, (2008)
[5] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious Language Models for Information Retrieval, In *Proceedings of SIGIR*, pp. 178–185, (2004)
[6] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately Interpreting Clickthrough Data as Implicit Feedback, In *Proceedings of SIGIR*, pp. 154–161, (2005)
[7] J. Lafferty and C.X. Zhai. Document Language Models, Query Models, and Risk Minimization for Information Retrieval, In *Proceedings of SIGIR*, pp. 111–119, (2001)
[8] V. Lavrenko and W.B. Croft. Relevance Models in Information Retrieval, *Language Modeling for Information Retrieval*, pp. 11–56, (2003)
[9] J. Parikh and S. Kapur. Unity: Relevance Feedback Using User Query Logs, In *Proceedings of SIGIR*, pp. 689–690, (2006)
[10] G. Pass, A. Chowdhury, and C. Torgeson. A Picture of Search, In *Proceedings of InfoScale*, pp. 1, (2006)
[11] J.M. Ponte and W.B. Croft. A Language Modeling Approach to Information Retrieval, In *Proceedings of SIGIR*, pp. 275–281, (1998)
[12] F. Radlinski and T. Joachims. Query Chains: Learning to Rank from Implicit Feedback, In *Proceedings of KDD*, pp. 239–248, (2005)
[13] J. Rocchio. Relevance Feedback in Information Retrieval, *The SMART Retrieval System*, pp. 313–323, (1971)
[14] X.H. Shen, B. Tan, and C.X. Zhai. Context-sensitive Information Retrieval Using Implicit Feedback, In *Proceedings of SIGIR*, pp. 43–50, (2005)
[15] X.H. Shen and C.X. Zhai. Exploiting Query History for Document Ranking in Iterative Information Retrieval, In *Proceedings of SIGIR*, pp. 377–378, (2003)
[16] B. Tan, X.H. Shen, and C.X. Zhai, Mining Long-term Search History to Improve Search Accracy, In *Proceedings of KDD*, pp. 718–723, (2006)
[17] C.X. Zhai and J. Lafferty. A Study of Smoothing Methods for Language Models Applied to Information Retrieval, *ACM Trans. Inf. Syst.*, vol. 22(2), pp. 179–214, (2004)
[18] C.X. Zhai and J. Lafferty. Model-based Feedback in The Language Modeling Approach to Information Retrieval, In *Proceedings of CIKM*, pp. 403–410, (2001)
[19] Z.M. Zhuang and S. Cucerzan. Re-ranking Search Results Using Query Logs, In *Proceedings of CIKM*, pp. 860–861, (2006)