Contextual Evaluation of Query Reformulations in a Search Session by User Simulation

Jiepu Jiang[1], Daqing He[1], Shuguang Han[1], Zhen Yue[1], Chaoqun Ni[2]
[1] School of Information Sciences, University of Pittsburgh
[2] School of Library and Information Science, Indiana University Bloomington

jiepu.jiang@gmail.com, dah44@pitt.edu, shh69@pitt.edu, zhy18@pitt.edu, chni@indiana.edu

ABSTRACT

We propose a method to dynamically estimate the utility of documents in a search session by modeling users' browsing behaviors and novelty. The method can be applied to evaluate query reformulations in the context of a search session.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software – *performance evaluation (efficiency and effectiveness).*

General Terms

Measurement, Performance, Experimentation, Human Factors.

Keywords

Evaluation; interactive search; query reformulation; query suggestion; search session.

1. INTRODUCTION

Formulating effective queries is essential to find relevant information. However, users may issue ineffective queries for many reasons, such as typo and using words that are over-generalized or over-specified. Besides, it is also a search strategy to issue multiple queries for search [1], for which each query deals with a facet or sub-topic of the problem. In a word, users usually need to reformulate queries several times in a search session.

Two different techniques can be applied to support systems in a search session. One is query suggestion [2-3], which suggests users with useful queries, so that the users can take the suggested queries for search rather than reformulating their own queries. The other one is to optimize the search results of query reformulations based on search histories, as studied in the TREC session track [4]. A popular method is to use the users' previous search queries and click-through data as relevance feedback for current searches [5]. As examined in [6], the new terms in users' query reformulations are very likely to come from users' search histories, e.g. previous search queries and contents of clicked and judged documents.

For studies of both techniques [2-5], a common problem needs to be solved is the evaluation of query reformulations in the context of a search session. Current studies usually evaluate query suggestions by user experiments [3] or using users' reformulations from search logs as ground truth [6]. But the former is expensive and difficult to be reused, and the latter requires search logs that are mostly inaccessible to the public. Other studies [7-8] adopt a system-oriented evaluation approach, in which TREC-style ad hoc search datasets and evaluation metrics (e.g. P@k and nDCG@k) are adopted without considering the contexts of search sessions. Such method is cheap and reusable, but as will be discussed in this paper, it is difficult to be justified and cannot cope with users' query reformulation behaviors.

We propose a method to evaluate individual query reformulation's search performance in the context of a search session. We believe the novelty factor should be considered in search session, i.e. a relevant document may be useless after it has been viewed by the user. We model users' browsing patterns and novelty in a session, so that the usefulness of a relevant document to the user will be discounted by both the probability that it has been viewed in previous searches and the user's needs for novelty. Our method can be applied to ad hoc search datasets or static session datasets [4] for evaluation of query suggestion algorithms [2-3, 6-8] and session search algorithms [4-5].

2. EVALUATION METHODS

2.1 User Model

To simplify the problem, we assume an idealized setting for search session: a search session is a process that involves one or many rounds of searches dealing with the same information need, for which we assume a set of documents can be judged as relevant to the whole session and the information need. This setting has been adopted in many related studies [4, 7-10]. In the following discussion, we use $\{q^{(1)}, \ldots, q^{(n)}\}$ for a session, or $q^{(1\ldots n)}$ for short; $q^{(i)}$ refers to the i^{th} query in the session.

A query reformulation can refer to any query except the first one of a session. Our purpose is to evaluate a query reformulation $q^{(n)}$'s results ($n \geq 2$) in the context that the user searched for $n - 1$ queries prior to $q^{(n)}$. We assume that a user's cognitive state will be updated in a search session when the user browses and examines search results, so that a relevant result may practically be useless to the user after it has been viewed many times.

Our evaluation method assumes the following user model:

(1) A relevant document d has the utility $rel(d)$ for the user at the beginning of the search session. In a static dataset, $rel(d)$ is the relevance score of d judged by the user.

(2) d may be retrieved as the results for one or many queries. After each time the user viewed d, d's utility has the probability β to be discounted to 0, and the probability $1 - \beta$ to be kept the same.

(3) The user will browse search results by the sequence they are ranked by the system. The user will always view the first result of a query. After viewing a result, the user has the probability p to continue viewing the next result, and the probability $1 - p$ to reformulate the next query or to leave the session.

Here (2) models the users' needs for novelty in a search session. We refer to the parameter β as the user's browsing novelty: a greater β value indicates a higher degree of browsing novelty and it is less likely that the user needs to view a result twice.

(3) models users' browsing behaviors in a search session. We adopt the browsing model in rank-biased precision (RBP) [11]. A

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10...$15.00.

similar browsing model has been adopted in the path-based search session evaluation methods [9]. However, our model differs from [9] in that we do not consider the cases that users may leave the search session before $q^{(n)}$ (modeled by p_{reform} in [9]). Instead, we consider a session $q^{(1\ldots n)}$ in the dataset as an existing fact which is caused by the user's decisions for reformulating from $q^{(1)}$ to $q^{(n-1)}$.

2.2 Relevance Discounting

According to the user model proposed in section 2.1, given a context $q^{(1\ldots n-1)}$, we can define interactive relevance of a document d in the context (*irel*) as the expected utility of the document after the user searched for $\{q^{(1)}, q^{(2)}, \ldots, q^{(n-1)}\}$, as calculated in Eq(1): *rel*(d) is the relevance score of d for the topic; β is user's browsing novelty, as defined in 2.1; $P_{\text{view}}(d|R^{(i)})$ is the probability that d has been viewed by the user in $R^{(i)}$, the search results of $q^{(i)}$. When n = 1 ($q^{(n)}$ is the first query in the session), *irel* is reduced to *rel*.

$$irel(d \mid q^{(1\ldots n-1)}) = rel(d) \cdot \left(1 - \prod_{i=1}^{n-1} \left(1 - \beta \cdot P_{view}(d \mid R^{(i)})\right)\right) \quad (1)$$

The parameter β seems naïve and is just simply neutralizing the two extreme cases (e.g. either completely discount or not). But the browsing model in $P_{\text{view}}(d|R^{(i)})$ can lead to natural discounting of relevance in a search session. According to the browsing model in section 2.1, the probability that the user has viewed a result d in result $R^{(i)}$ is $p^{\text{rank}(d,\, i)\, -\, 1}$, as in Eq(2): rank($d$, i) is the rank position of d in $R^{(i)}$. When d is not in $R^{(i)}$, we simply assign the probability $P_{\text{view}}(d|R^{(i)})$ as 0.

$$P_{view}(d \mid R^{(i)}) = \begin{cases} p^{rank(d,i)-1} & d \in R^{(i)} \\ 0 & d \notin R^{(i)} \end{cases} \quad (2)$$

irel will discount a document's utility to a greater extent if: 1) the document is retrieved by a great number of previous search queries; 2) the document is ranked higher in the results of previous queries; 3) a larger value of p or β is assigned. Let $\{d_1, d_2, \ldots, d_{10}\}$ be a result list of 10 documents, each of which has relevance score 1. Figure 1 shows the *irel* scores for the 10 documents after the user viewed the result list $\{d_1, d_2, \ldots, d_{10}\}$ once.

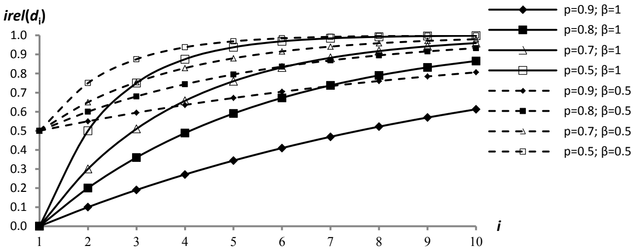

Figure 1. Discounting effects of irel with different parameters.

To evaluate a query reformulation $q^{(n)}$ in the context $q^{(1\ldots n-1)}$, we can first calculate *irel* based on the context $q^{(1\ldots n-1)}$, and then calculate ad hoc search evaluation metrics using *irel* scores rather than *rel*. We put "*i*" in front of the name of an ad hoc search evaluation metric if it is calculated using *irel*, e.g. inDCG@10 means to calculate nDCG@10 using *irel*.

Our method differs from the path-based search session evaluation methods [9] and nsDCG [10] from two perspectives. First, the *irel* method evaluates individual query's search performance in the context of a search session, while [9] and [10] both evaluate a whole search session's search performance. Second, we explicitly considered users' browsing novelty in a search session, but [9] and [10] either assumes no novelty effect exists or ignores duplicate documents in evaluation, which are difficult to be explained.

2.3 Parameters

Here we simply set the two parameters β and p intuitively. However, both parameters may further be modeled by user factors in interactive search:

Browsing style and effort: some users may quickly scan results while some others may carefully examine one by one. Users of different styles may have different browsing persistence (p) and also different chances of missing relevant information in a document, which can be used to model the parameter β. When browsing style and effort are being considered, β and p may be related to each other.

Background knowledge and familiarity with the topic: a user's background knowledge and familiarity with related topics may influence whether, after viewing a result, the user can understand the information, which can be used to model β. Sometimes a user may gradually get familiar with the problem during a search session. Thus, β may change in different stages of the search session.

We also refer to the two parameters β and p as indicators for **search interactiveness**, which is the extent to which the interactive search problem is influenced by users' interactions and differs from an ad hoc search problem. In our study, a higher p or β value will cause a greater difference between *rel* and *irel*, and a greater difference between the interactive search problem and an ad hoc search one.

3. DATASETS

In order to examine users' query reformulation behaviors, we use TREC session track dataset in 2011, which includes 76 search sessions, 280 queries, and 204 query reformulation pairs from real users. We do not use the dataset in 2010 because it is not created using real user experiments. TREC session track used Clueweb09 dataset. We retrieve results for each query using Indri (by query likelihood model) on Clueweb09B dataset. Documents with waterloo spam rank scores less than 70 are removed.

In order to simulate different query reformulation methods and the results of queries on different search systems, we use the TREC8 query track dataset. In TREC 8 query track dataset [13], 6 groups built 9 different retrieval systems, and created 21 different sets of queries for the same 50 topics. For each set of the queries, results on each of the 9 systems are provided.

We compare inDCG with nDCG and nsDCG. nsDCG is implemented using the formula and parameter settings in [9]. We do not remove duplicate results in calculating nsDCG.

4. RESULTS

4.1 Users' Query Reformulation Behavior

Although we properly modeled and explained *irel* and the changes of relevance in a search session, without golden standards, one may still argue the validity of *irel* and the evaluation metrics using *irel*. In our study, although we cannot fully prove the validity of *irel*, we did find out some proof against using ad hoc search metrics for evaluating query reformulations.

In TREC 2011 session track dataset, we extract 204 pairs of users' reformulations ($q_{n-1} \rightarrow q_n$). Table 2 shows the changes of ad hoc search performance from q_{n-1} to q_n and the similarities between the pairs of queries' top ranked results. First, we find, in general, users' query reformulations will not cause significant change on ad hoc search performance. On average, the absolute values of changes in P@10, P@20, nDCG@10, and nDCG@20 do not exceed over 0.03, with certain degree of variance (standard deviation ranging from 0.2 to 0.3 on average); the changes are not significant by neither a paired t-test nor a Wilcoxon test (the reported p values are for the paired t-test).

Results in Table 2 indicate it is unlikely that the users are reformulating for the purpose of improving queries' ad hoc search performance (if we assume the users can in general reformulate effectively), which can be a proof against the use of ad hoc search evaluation metrics in interactive search sessions.

We further examine the similarity between q_{n-1} and q_n's rankings of results and the sets of documents returned in top position. We find on average the users' reformulations tend to retrieve results that are very different from those from previous queries, with average jaccard similarity of results in top 10 and top 20 positions about only 0.35. In general, rankings of results by q_{n-1} and q_n are not correlated. These results indicate users may reformulate queries in order to find novel results that are different from those of previous queries.

Table 2. Improvements of ad hoc search performance and similarity of results for users' query reformulations.

H$_0$: $f_{adhoc}(q_{n-1}) = f_{adhoc}(q_n)$		**Changes of f_{adhoc} by q_{n-1}→q_n**		
		mean	**SD**	***p* value**
f_{adhoc} (average over topics)	P@10	0.026	0.275	0.171
	P@20	0.022	0.212	0.143
	nDCG@10	0.021	0.241	0.209
	nDCG@20	0.019	0.204	0.180
Similarity of q_{n-1} and q_n in top ranked results				
Jaccard Similarity (average over topics)	Top 10 results	0.357	0.377	-
	Top 20 results	0.354	0.360	-
Spearman's ρ (average over topics)	Top 10 results	0.103	0.609	-
	Top 20 results	0.145	0.577	-

4.2 Stability

4.2.1 Error Rate Revisited

Error rate is an indicator for the stability of evaluation metrics. A popular method of calculating error rate [14] aims at studying the following problem: if statistically significant differences between two systems have been observed on one topic set, will we observe conflicting results on another topic set? This type of error rate is enough to indicate the stability of our metrics for evaluating session search algorithms, as studied in [4-5]. However, for query suggestions, we need to also consider the effects of search systems, because it relies on the specific search systems to generate results for the query suggestions. A superior query suggestion method may be ineffective if we switch to other search systems.

We use TREC 8 query track data for our study. We iteratively use one query set (A) from the 21 sets as "original queries", and two other sets (B and C) as two types of reformulations. Thus, we will have $21 \times \binom{2}{20} = 3990$ pairs of "systems" for comparison and draw 3990 conclusions on which one is more effective. For each pair of query reformulations for comparison, results can be calculated on different topics and different systems. We can calculate two types of error rate:

(1) **Within-system error rate**. The problem to be studied is: in a system, if we observe statistically significant difference between two algorithms from one set of topics, whether conflicting results will be observed from another set of topics. Random partitions of topics are generated, i.e. we split 50 topics into two partitions of size n and $50 - n$. n ranges from 5 to 25. For n value, we randomly generate 100 different partitions for our study. For each of the randomly generated size n partition (referred to as "*decision partition*" in following discussions), we use a paired t-test with $p < 0.05$ to draw conclusions on whether "A→B" is better or worse than "A→C", and check whether conflicting results are observed on the size $50 - n$ partition. Error rate is calculated as the rate that a conflicting conclusion is observed.

(2) **Cross-system error rate**. The problem to be studied is: on a set of topics, if we observe statistically significant difference between two query suggestion algorithms in one retrieval system, whether we will observe conflicting results from another system on the same set of topics. We also randomly generate subsets of the 50 topics from size 5 to 25 (100 random subsets for each n value). For each random partition of topics, we iteratively examine for each pair of the 9 systems on whether significant results are observed from one system, but conflicting results are observed from another. Error rate is calculated as the number of times a conflicting result is observed divided by the number of times a significant result is observed on the decision partition.

Within-system error rate studies the traditional problems in studying IR evaluation metrics, e.g. whether are conclusions from experiments on a limited number of topics generalizable? In comparison, cross-system error rate studies a unique problem for query suggestions, e.g. whether a good query suggestion for one retrieval system is still effective when we switch to another system.

4.2.2 Within-System Error Rate

Figure 3 shows the within-system error rates for nDCG@10, nsDCG@10, and inDCG@10 with different values of p and β. To compare with previous studies of error rates [14], we also estimate a trend line for error rates. However, instead of using an exponential function as suggested in [14], we find power functions may better fit with the trends. In general, we find exponential function has limited fitness with the observed values, with R^2 ranging from 0.4 to 0.6, while R^2 for power functions are around 0.7 to 0.8.

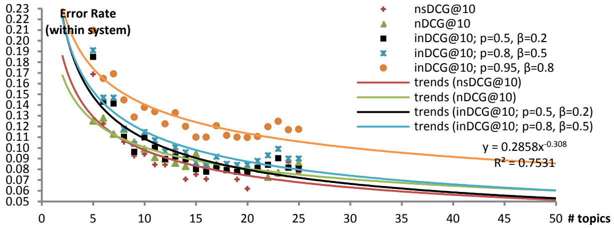

Figure 3. Within-system error rate of inDCG, nDCG, and nsDCG.

In general, error rate will drop with increased number of topics in the decision partition. This observation is similar to previous studies in evaluation metrics for IR systems. When a limited number of topics are examined, e.g. n = 5 or 10, we will easily come to wrong conclusions, no matter by ad hoc search metrics such as nDCG@10 or by session search metrics.

For inDCG@10 with different parameters, we find the more we discount relevance by contexts (the higher p and β values and the higher interactiveness), the less stable the inDCG metrics are. To keep the clarity of figure 3, we only show error rates of inDCG for 3 different parameter settings. The general trends we observed are very consistent (with p from 0.5 to 0.95, β from 0.2 to 1.0): error rate of inDCG will increase if either p or β increases. For example, inDCG@10 with $p = 0.95$ and $\beta = 0.8$ will consistently have about 0.02 higher error rate than inDCG@10 with $p = 0.8$ and $\beta = 0.5$. The trends of parameters are also consistent with the lowest error rate observed for nsDCG. Because we do not remove duplicate documents, nsDCG do not consider novelty and will not discount relevance of articles by contexts, which is comparable to the setting of inDCG with $\beta = 0$. Results indicate the error rates of inDCG are comparable to those of nDCG and nsDCG; only when the search problem is highly interactive (with high values of p and β), inDCG will have an observable higher error rate than nDCG and nsDCG.

We also find the higher values of p and β, the less similar and correlated between query suggestions' ad hoc search performance (by nDCG) and the search performance in a search session (by inDCG). We applied the query suggestion method in [8] to TREC robust 04 dataset using Clueweb09B anchor texts as alternatives of query logs for query reformulation. For each topic, a list of query suggestions (including both query term addition and query term substitution patterns) are generated. We select up to 100 top ranked queries by nDCG@10 and inDCG@10. Figure 4 shows the correlation of queries' rankings by nDCG and inDCG, and the overlap between the two ranked lists of queries. Results in Figure 4 indicate the higher degree of search interactiveness (greater *p* or *β*), the less similarity between queries' ad hoc search performance and the search performance in session. When the problem is highly interactive (e.g. $p = 0.95$ and $\beta = 0.8$), queries' performance by nDCG and inDCG are very different (spearman's ρ is only 0.130 and jaccard similarity is 0.42).

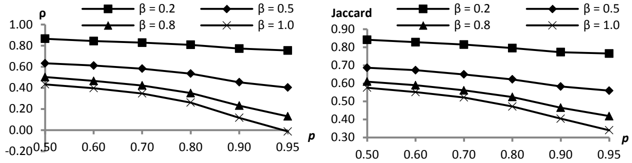

Figure 4. Similarity and correlation between ranked lists of queries by inDCG@10 and nDCG@10.

We further increase the cutoff value *k* for nDCG, inDCG, and nsDCG. In general, when evaluating with a higher cutoff *k* value, we will observe a slightly lower error rate and higher similarity and correlation between queries' ad hoc search and session search performance.

4.2.3 *Cross-System Error Rate*

We further study the cross-system error rates of metrics in evaluation of query suggestions. Figure 5 shows results of cross system error rates for inDCG, which indicates the comparative performance of queries are surprisingly consistent cross different search systems. Although previous studies indicate different IR systems have strong bias to certain types of topics, and practically even "the best system is normally above average for most of the topics, and best for maybe 5%-10% of the topics" [13], our results in Figure 5 indicate the superior performance of one query over others in a retrieval system is very stable when we switch to other retrieval systems. Even when only 5 topics are randomly sampled, two query sets will perform very similarly cross different systems on the sampled 5 topics.

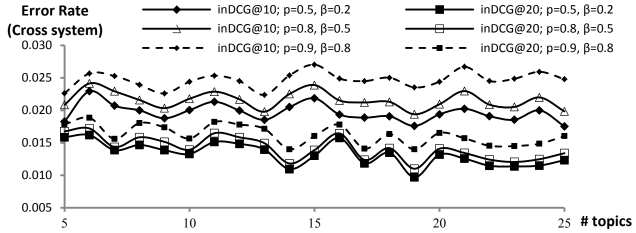

Figure 5. Cross-system error rates of inDCG.

Similar to the trends in Figure 3, we also notice the more we discount relevance in *irel* (with higher *p* and *β*), the more likely we come to a wrong conclusion cross different systems. But in general only less than 3% error rates are observed, which indicate, once we find a good way of generating queries, it can be applied effectively to most of other retrieval systems. This also suggests results reported for query suggestion algorithms using one retrieval system are very likely to be generalized to other retrieval systems.

5. CONCLUSIONS

In this paper, we propose a method to evaluate query reformulations in the context of a search session, which can be used as economic alternatives of user studies and query logs to evaluate query suggestion algorithms [2, 8] or session search system [4, 5]. We find users tend to reformulate queries that can retrieve search results very different from those of previous queries, but users do not reformulate to enhance the ad hoc search performance, which indicates ad hoc search metrics should not be adopted to evaluate query reformulations in a search session. The proposed evaluation methods are stable compared with existing metrics, and have the advantages of simulating users' browsing behaviors and novelty. We find the higher the search interactiveness, the less stable the evaluation metrics are. Besides, queries' search performance are very stable over different retrieval systems, which suggests query suggestion can be widely adopted as a general technique mostly independent of the differences of search systems.

6. ACKNOWLEDGMENTS

This work was supported in parts by National Science Foundation grant IIS-1052773 and III-COR 0704628.

7. REFERENCES

[1] M. J. Bates. 1989. The design of browsing and berrypicking techniques for the online search interface. Online Information Review, 13(5), pp.407-424.

[2] R. Baeza-Yates, C. Hurtado and M. Mendoza. 2005. Query Recommendation Using Query Logs in Search Engines. In EDBT 2004 Workshops, 395-397.

[3] D. Kelly, K. Gyllstrom, and E. W. Bailey. 2009. A comparison of query and term suggestion features for interactive searching. In SIGIR '09, 371-378.

[4] E. Kanoulas, P. Clough, B. Carterette, and M. Sanderson. Session track at TREC 2010. In SIGIR 2010 Workshop on Simulation of Interaction: Automated Evaluation of Interactive IR, 2010.

[5] Z. Yue, J. Jiang, S. Han, D. He. 2012. Where Do the Query Terms Come from? An Analysis of Query Reformulation in Collaborative Web Search. In CIKM '12.

[6] J. Jiang, D. He, S. Han, J. Wu. 2011. Pitt at TREC 2011 session track. In TREC 2011.

[7] L. Bing, W. Lam, and T. Wong. 2011. Using query log and social tagging to refine queries based on latent topics. In CIKM '11, 583-592.

[8] X. Wang and C. Zhai. 2008. Mining term association patterns from search logs for effective query reformulation. In CIKM '08, 479-488. V. Dang and B. W. Croft. 2010. Query reformulation using anchor text. In WSDM '10, 41-50.

[9] E. Kanoulas, B. Carterette, P. D. Clough, and M. Sanderson. 2011. Evaluation over multi-query sessions. In SIGIR '11.

[10] K. Järvelin, S. L. Price, L. Delcambre and M. L. Nielsen. 2009. Discounted Cumulated Gain Based Evaluation of Multiple-Query IR Sessions. In ECIR '08, 4-15.

[11] A. Moffat and J. Zobel. 2008. Rank-Biased Precision for Measurement of Retrieval Effectiveness. In TOIS 27(2).

[12] C. Clarke, M. Kolla, G. Cormack et al. 2008. Novelty and diversity in information retrieval evaluation. In SIGIR '08, 659–666.

[13] C. Buckley and J. Walz. 1999. The TREC-8 Query Track. In TREC 8.

[14] M. Sanderson, J. Zobel. 2005. Information Retrieval System Evaluation: Effort, Sensitivity, and Reliability. In SIGIR '05.