Query Change as Relevance Feedback in Session Search

Sicong Zhang, Dongyi Guan, Hui Yang
Department of Computer Science
Georgetown University
37th and O Street, NW, Washington, DC 20057 USA

{sz303, dg372}@georgetown.edu, huiyang@cs.georgetown.edu

ABSTRACT

Session search retrieves documents for an entire session. During a session, users often change queries to explore and investigate the information needs. In this paper, we propose to use query change as a new form of relevance feedback for better session search. Evaluation conducted over the TREC 2012 Session Track shows that query change is a highly effective form of feedback as compared with existing relevance feedback methods. The proposed method outperforms the state-of-the-art relevance feedback methods for the TREC 2012 Session Track by a significant improvement of >25%.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information Search and Retrieval*

Keywords

Relevance Feedback; Session Search; Query Change

1. INTRODUCTION

Session search retrieves documents for an entire session of queries. [3, 4]. It allows the user to constantly modify queries in order to find relevant documents. Session search involves many interactions between the search engine and the user. The challenge for session search is how to make use of these interactions and the user feedback to effectively improve search accuracy. In TREC (Text REtrieval Conference) 2012 Session tracks [6], the users (NIST assessors) clicked retrieved documents and interacted with a search engine to produce the queries and sessions. For each intermediate query, a retrieved document set containing the top 10 retrieval results ranked in decreasing relevance for the query are kept. The clicked data contains the documents clicked by users, their clicking orders, and dwell time. Figure 1 illustrates the interactions among the user and the search engine.

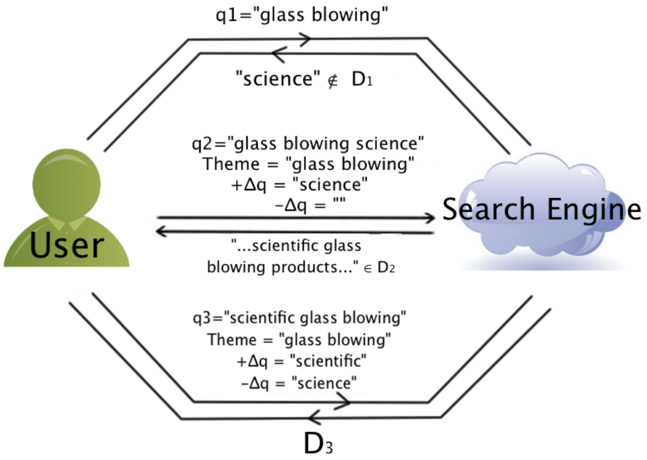

Figure 1: Session search. (TREC 2012 session 85)

Relevance feedback is a popular IR technique. By expanding queries with terms from relevant feedback documents, relevance feedback is able to generate better queries and uses them for better retrieval accuracy. Commonly used relevant feedback schemes include Rocchio with real user feedback [5], pseudo relevance feedback [1],and implicit relevant feedback [7]. Real user feedback is obtained from human assessors that indicate the relevance of a document retrieved for a query. Pseudo relevance feedback, also known as blind relevance feedback, that assumes that the top retrieved documents are relevant, and makes query expansion based on these pseudo relevant documents. Implicit relevance feedback is the form of feedback that is inferred from user behaviors, such as user clicks, clicking order, and dwell time.

In this paper, we propose to use query change as a new form of relevance feedback in session search. Our method utilizes editing changes between two adjacent queries, and the relationship between query change and retrieved documents for the earlier query to enhance session search. Our experiments demonstrate that the proposed approach outperforms other relevance feedback methods.

2. DEFINING QUERY CHANGE

We represent a *search session* $\mathcal{S}$ as a series of queries $\{q_1, ..., q_i, ..., q_n\}$. For an individual query q_i, we can write it as a combination of the common part and the changes between it and its previous query: $q_i = (q_i \cap q_{i-1}) + \Delta q_i$.

We define *query change* Δq_i as the syntactic editing differ-

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.
Copyright 2013 ACM 978-1-4503-2034-4/13/07 ...$15.00.

ences between two adjacent queries q_{i-1} and q_i. Considering the directions of editing, query change Δq_i can be further decomposed into two parts: *positive* Δq and *negative* Δq. They are written as "$+\Delta q$" and "$-\Delta q$" respectively. The *positive* Δq are new terms that the user adds to the previous query; that is, they appear in q_i, but did not appear in q_{i-1}. The *negative* Δq are terms that the user deletes from the previous query; that is, they appeared in $qi-1$, but not appear in q_i.

We thus decompose an adjacent query pair into:

$$+\Delta q_i = q_i \setminus q_{i-1}$$

$$-\Delta q_i = q_{i-1} \setminus q_i$$

$$q_{theme} = q_i \setminus (+\Delta q_i) = q_{i-1} \setminus (-\Delta q_i)$$

where $+\Delta q_i$ and $-\Delta q_i$ represent added terms and removed terms respectively, q_{theme} is the theme terms, and the notation of $\setminus$ represents set-theoretic difference. Table 1 demonstrates a few example TREC 2012 Session queries and their query changes.

The theme terms (q_{theme}) appear in both q_{i-1} and q_i. Generally it implies a strong preference for those terms from the user. For example, in Table 1 session 32, q_1 = "bollywood legislation", q_2 = "bollywood law". q_{theme} = "bollywood".

The added terms ($+\Delta q$) may indicate a specification or drifting between q_{i-1} and q_i. In session 32, ($+\Delta q_2$) = "law".

The removed terms ($-\Delta q$) may indicate a generalization or a drifting. In session 32, ($-\Delta q_2$) = "legislation".

Table 1: Examples of TREC 2012 Session queries.

	Queries	Query Change
Session 32	query 1 = bollywood legislation query 2 = bollywood law	$+\Delta q_2 = law$ $-\Delta q_2 = legislation$
Session 85	query 1 = glass blowing query 2 = glass blowing science query 3 = scientific glass blowing	$+\Delta q_2 = science$ $-\Delta q_2 = \Phi$ $+\Delta q_3 = scientific$ $-\Delta q_3 = science$

3. UTILIZING QUERY CHANGE

Besides queries, a TREC session also contains retrieved document sets $\mathcal{D}$ (set of D_i) for each query q_i, and clicked information $\mathcal{C}$ (set of C_i) for each query q_i.

Based on observation of session search and user intension, we propose an important assumption that the previous search result D_{i-1} influences the current query change Δq_i:

$$\Delta q_i \leftarrow D_{i-1}.$$

In fact, this influence can be in quite a complex way. Figure 1 shows session 85 as an example, illustrating how the previous retrieved documents D_{i-1} influence the query changes.

Based on our definition of query change, we utilize different cases of query change in the calculation of relevance score between the current query q_i and a document d.

Suppose $P(t|d)$ is the original term weight for the retrieve model in our utilization, we increase and decrease term weights on top of it. In the following formulas, $P(t|d)$ is calculated by the multinomial query generation language model with Dirichlet smoothing [9] while $P(t|d)$ is calculated based on Maximum-Likelihood Estimation (MLE):

$$P(t|d) = \frac{TF(t,d) + \mu P(t|C)}{Length(d) + \mu},$$

where d is the document under evaluation, $Length(d)$ is the length of the document, $TF(t,d)$ is the term frequency of t in document d, $P(t|C)$ calculates the probability that t appears in corpus C based on MLE. μ is set to 5000 in experiments.

We adjust the term weights for the three types of query changes as the following:

- Theme terms are the repeated common parts nearly appearing in the entire session. It implies their importance to the session and to the user. We therefore propose to increase their term weights. It is worth noting that theme terms are common terms within a session which show a similar effect of stop words. However, they may not be common terms in the entire corpus. We propose to use a measure that is similar to inverse document frequency (*idf*) to capture this characteristic. We employ the negation of the number of occurrences of t in D_{i-1}, $1 - P(t|D_{i-1})$. The weight increase for a theme term $t \in q_{theme}$ is formulated as follows:

$$W_{Theme} = \sum_{t \in q_{theme}} [1 - P(t|D_{i-1})] \log P(t|d) \quad (1)$$

- For the added terms that occurred in the previous search result D_{i-1}, which are terms $t \in +\Delta q$ and $t \in D_{i-1}$, we deduct their term weights. This is because the term appear both in documents for previous query and in the current query, it will bring back repeated information from the previous query to the current query in some degree. In addition, $t \in +\Delta q$ shows these added terms are not theme terms. Therefore, it has a high probability to deviate from the recent focus of the current query. We thus deduct more weights to reduce redundant information. The weight deduction is proportional to t's term frequency in D_{i-1}.

$$-W_{Add,In} = -\sum_{\substack{t \in +\Delta q \\ t \in D_{i-1}}} P(t|D_{i-1}) \log P(t|d) \quad (2)$$

- For the added terms that did not occur in the search result of previous query D_{i-1}, which are terms $t \in +\Delta q$ and $t \notin D_{i-1}$, we increase the term weights because they demonstrate the novel interests of the user for the current query q_i. We propose to raise the term weights based on inverse document frequency in order not to increase their weights too much if they are common terms in the corpus.

$$W_{Add,Out} = \sum_{\substack{t \in +\Delta q \\ t \notin D_{i-1}}} idf(t) \log P(t|d) \quad (3)$$

- For the terms that are from the previous query, which are terms $t \in -\Delta q$. No matter $t \in D_{i-1}$ or $t \notin D_{i-1}$, we should deduct their term weights. The reason is the following. If they appeared in D_{i-1}, it means that the user observed them and disliked them. If they did not appear in D_{i-1}, the user still dislikes the terms since they are not included in q_i anyway. Just like terms that in added terms that appeared in previously retrieved documents ($t \in +\Delta q$ and $t \in D_{i-1}$), we deduct the term weight for the removed terms according to the following formula.

$$-W_{Remove} = -\sum_{t \in -\Delta q} P(t|D_{i-1}) \log P(t|d) \quad (4)$$

Table 2: Dataset statistics for TREC 2012 Session Track.

#topic =48 #session =98	#query/session =3.03 #session/topic =2.04	#query = 297 #docs =17,861

Table 3: nDCG@10, MAP, and their improvements over the baseline (%chg) for TREC 2012. A statistical significant improvement on nDCG@10 over the baseline is indicated with a † at p < 0.05.

	nDCG@10	%chg	MAP	%chg
Lemur	0.2622	0.00%	0.1342	0.00%
PRF	0.2718	3.66%	0.1309	-2.46%
RF D_{i-1}	0.2122	-19.07%	0.1137	-15.28%
Implicit Click	0.2668	1.75%	0.1355	0.97%
Implicit SAT	0.2655	1.26%	0.1335	-0.52%
QueryChg CLK	**0.3306**	**26.09%**†	**0.1533**	**14.23%**
QueryChg SAT	**0.3300**	**25.86%**†	**0.1535**	**14.38%**

By considering all cases above, the relevance score between the current query q_i and a document d can be represented as a linear combination of various term weight adjustments:

$$\begin{aligned} Score(q_i, d) &= \log P(q_i|d) + \\ &+ \alpha W_{Theme} - \beta W_{Add,In} + \epsilon W_{Add,Out} - \delta W_{Remove} \end{aligned} \tag{5}$$

where d is the document under evaluation, $\log P(q_i|d)$ is the original query-document relevance scoring function in log form, α, β, ϵ, and δ are coefficients for each type of query changes. Empirically, we set the coefficients as $\alpha = 2.2$, $\beta = 1.8$, $\epsilon = 0.07$, and $\delta = 0.4$.

4. EXPERIMENTS

4.1 Search Accuracy Using the Last Query

We evaluate our algorithm on the TREC 2012 Session Track [6]. According to how much prior information is used, the Track is divided into four phases: RL1 (using only the last query), RL2 (using all queries in the session), RL3 (using all session queries and ranked lists of URLs and the corresponding web pages), RL4 (using all session queries, the ranked lists of URLs and the corresponding web pages, the clicked URLs, and the time that the user spent on the corresponding web pages). Table 2 shows the statistics about the TREC 2012 Session Track.

The corpus used in our evaluation is ClueWeb09 CatB.[1] CatB consists of 50 million English pages from the Web collected during two months in 2009. We removed documents whose Waterloo's "GroupX" spam raining score [2] are less than 70.

We compare the following algorithms in this paper:

- *Baseline (Lemur without relevance feedback)* Using the original Lemur system (language modeling + Dirichlet smoothing) to retrieve for the last query q_n.
- *PRF (Pseudo Relevance Feedback).* We utilize pseudo relevance feedback algorithm that developed in Lemur. We use the top 20 documents as pseudo relevant documents. The retrieval is for the last query q_n.
- *RF D_{i-1}.* Rocchio using the previously retrieved top documents proved by TREC. This method uses q_n, q_{n-1}, and D_{n-1}.
- *Implicit Click.* Implicit relevance feedback based on clicked documents of the previous search query q_{i-1}. This method uses $q_n, q_{n-1}, D_{n-1}, C_{n-1}$.
- *Implicit SAT.* Implicit relevance feedback based on SAT [8] clicked documents (the documents that the user clicked and stayed on for at least 30 seconds) from the previous query. This method uses q_n, q_{n-1}, D_{n-1}, C_{n-1}.
- *QueryChg CLK.* (Our algorithm) Relevance feedback using query change based on Eq. 5. This method uses $q_n, q_{n-1}, D_{n-1}, C_{n-1}$. D_{i-1} include the clicked documents and all snippets for the previous query.
- *QueryChg SAT.* (Our algorithm) Relevance feedback using query change based on Eq. 5. This method uses $q_n, q_{n-1}, D_{n-1}, C_{n-1}$. D_{i-1} are SAT clicks and all snippets for the previous query.

Table 4: nDCG@10, MAP, and their improvements over the baseline (%chg) for TREC 2012, after uniform aggregation. A statistical significant improvement on nDCG@10 over the baseline is indicated with a † at p < 0.05.

	nDCG@10	%chg	MAP	%chg
Lemur	0.3227	0.00%	0.1558	0.00%
PRF	0.2986	-7.46%	0.1413	-9.31%
RF D_{i-1}	0.2446	-24.20%	0.1281	-17.78%
Implicit Click	0.2916	-9.64%	0.1449	-7.00%
Implicit SAT	0.2889	-10.47%	0.1467	-5.84%
QueryChg CLK	**0.3258**	**0.96%**	0.1532	-1.67%
QueryChg SAT	**0.3350**	**3.81%**	0.1534	-1.54%

Table 3 shows the search accuracy for these seven runs. We employ the official TREC Session evaluation metrics, nDCG@10 and mean average precision (MAP), for measuring search accuracy. We can see that the proposed methods (QueryChg CLK, QueryChg SAT) improve the baseline by 26.09% and 25.86% respectively in nDCG@10. The improvements are statistically significant (one sided t-test, p =0.05). They also outperforms all other relevance feedback runs. Among other runs, PRF and implicit relevance feedback both improve over the baseline. RFD_{i-1}, however, decreases nDCG@10 by 19.07% than the baseline. This decrease is expected. RFD_{i-1} makes query expansion based on D_{i-1}, which increases the weights of old terms in D_{i-1}. An ideal relevance feedback model, however, should assign a lower weight to these terms since they are no longer novel or no longer satisfying the current information need.

4.2 Search Accuracy Using All Queries

There are multiple queries in sessions search. Prior research has demonstrated that using all queries can effectively improve search accuracy for session search over just using the last query [6]. This technique is called query aggregation. We evaluate our algorithm with query aggregation in this section.

Let $Score_{session}(q_n, d)$ denote the overall relevance score for a document d to the entire session, the aggregated session relevance score can be written as: $Score_{session}(q_n, d) = \sum_{i=1}^{n} \lambda_i \cdot Score(q_i, d)$, where n is the number of queries in a session, $Score(q_i, d)$ is the relevance score between d and q_i, and λ_i is the query weight for q_i. In this paper, we employ the *uniform query aggregation* by setting all queries are equally weighted ($\lambda_i = 1$) for all systems under evaluation.

Table 4 shows the search accuracy with uniform aggregation over all queries. Comparing Table 4 with Table 3, we

[1] http://lemurproject.org/clueweb09/

Table 5: nDCG@10 for different classes of sessions in TREC 2012 Session Track.

	Intellectual	%chg	Specific	%chg	Amorphous	%chg	Factual	%chg
Lemur	0.2740	0.00%	0.2529	0.00%	0.2741	0.00%	0.2557	0.00%
PRF	0.2814	2.70%	0.2721	7.60%	0.2713	-1.02%	0.2664	4.20%
RF D_{i-1}	0.2009	-26.65%	0.1995	-21.12%	0.2285	-16.63%	0.2185	-14.54%
Implicit Click	0.2742	0.10%	0.2508	-0.83%	0.2873	4.81%	0.2627	2.74%
Implicit SAT	0.2749	0.34%	0.2555	1.03%	0.2783	1.55%	0.2603	1.81%
QueryChg Click	0.3746	36.73%	0.3041	20.23%	0.3646	33.03%	0.3062	19.77%
QueryChg SAT	0.3759	37.20%	0.3062	21.08%	0.3604	31.48%	0.3045	19.09%

Table 6: nDCG@10 for different classes of sessions in TREC 2012 Session Track. Uniform Aggregation

	Intellectual	%chg	Specific	%chg	Amorphous	%chg	Factual	%chg
Lemur	0.3656	0.00%	0.2983	0.00%	0.3539	0.00%	0.2989	0.00%
PRF	0.3634	-0.60%	0.2654	-11.05%	0.3412	-3.58%	0.2626	-12.12%
RF D_{i-1}	0.2703	-26.08%	0.2233	-25.15%	0.2719	-23.18%	0.2304	-22.92%
Implicit Click	0.3235	-11.51%	0.2743	-8.07%	0.3138	-11.33%	0.2739	-8.36%
Implicit SAT	0.3235	-11.53%	0.2767	-7.24%	0.3045	-13.96%	0.2697	-9.75%
QueryChg Click	0.3575	-2.22%	0.3089	3.55%	0.3474	-1.84%	0.3082	3.11%
QueryChg SAT	0.3818	4.43%	0.3125	4.76%	0.3637	2.76%	0.3089	3.37%

observe that all systems improve their search accuracy when using query aggregation. The proposed QueryChg SAT run achieves an nDCG@10 of 0.3350, which is a 3.81% improvement over Lemur after uniform query aggregation, and a 27.76% improvement over Lemur without query aggregation. The Lemur run after query aggregation performs well (nDCG@10=0.32) as compared with without query aggregation (nDCG@10=0.26 in Table 3). However, the proposed query change runs (QueryChg Click, QueryChg SAT) do not benefit much from query aggregation. This may be because that uniform aggregation equally weights each query, which assumes query independence among the queries in a session; whereas the query change relevance feedback runs assume that previous query and current query are dependent. The difference in the assumptions between query change relevance feedback model and the uniform aggregation may be the reason that the former does not benefit much from the latter. Other aggregation methods may be able to improve the situation.

4.3 Results On Different Session Types

TREC 2012 sessions were created by considering two different dimensions: product type and goal quality. For product type, a session can be classified as searching for either factual or intellectual target. For search goal, a session can be classified as either specific or amorphous.

Both Table 5 and Table 6 show that the proposed method demonstrate difference effects on different session types. It achieves more improvement on Intellectual sessions (37.20%) and Amorphous sessions (31.48%) than on Factual sessions (19.09%) and Specific sessions (21.08%). This suggests that for more exploratory-style sessions, i.e., more difficult sessions, such as Intellectual and Amorphous sessions, our method is able to generate more performance gain. We believe that our method effectively captures query changes and well represents the dynamics in a search session.

5. CONCLUSION

Based on the idea that query change is an important form of feedback, this paper presents a novel relevance feedback model by utilizing query change. Experiments show that our approach is highly effective and outperforms other feedback models for the TREC 2012 Session Track. Moreover, the proposed relevance feedback method demonstrates different effects over sessions with different types of search targets and goals. It achieves more improvement on the more difficult sessions, such as Intellectual and Amorphous sessions, over the baseline system which does not use relevance feedback. We believe that our method better captures the exploratory nature of a search session by treating query changes as effective user feedback.

6. ACKNOWLEDGMENTS

This research was supported by NSF grant CNS-1223825. Any opinions, findings, conclusions, or recommendations expressed in this paper are of the authors, and do not necessarily reflect those of the sponsor.

7. REFERENCES

[1] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *SIGIR '08*, pages 243–250. ACM.

[2] G. V. Cormack, M. D. Smucker, and C. L. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *Inf. Retr.*, 14(5), Oct. 2011.

[3] D. Guan, H. Yang, and N. Goharian. Effective structured query formulation for session search. In *TREC '12*.

[4] J. Jiang, D. He, and S. Han. Pitt at trec 2012 session track. In *TREC '12*.

[5] T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *ICML '97*.

[6] E. Kanoulas, B. Carterette, M. Hall, P. Clough, and M. Sanderson. Overview of the trec 2012 session track. In *TREC'12*.

[7] Y. Song and L.-w. He. Optimal rare query suggestion with implicit user feedback. In *WWW '10*.

[8] D. Sontag, K. Collins-Thompson, P. N. Bennett, R. W. White, S. Dumais, and B. Billerbeck. Probabilistic models for personalizing web search. In *WSDM '12*.

[9] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, Apr. 2004.