Modeling Rich Interactions in Session Search — Georgetown University at TREC 2014 Session Track

Jiyun Luo, Xuchu Dong and Hui Yang
Department of Computer Science, Georgetown University
37th and O Street NW, Washington DC, USA
{jl1749,xd47}@georgetown.edu, huiyang@cs.georgetown.edu

Abstract

This year we participate in the TREC Session Track Task 1. We adopt the Query Change Model (QCM), weighted QCM, re-ranking, clustering, and error analysis in our approaches. The QCM retrieval model is employed to combine all queries in a session. QCM allows documents that are relevant to any query in a session to appear in the final retrieval list. Weighted QCM combines queries unevenly based on a prediction of query quality. It is based on the following intuition: if a query does not bring any document that leads to a SAT-Click from the user, it suggests that this query is poorly formed. Our re-ranking module is based on implicit feedback from the user; in this case the SAT-Clicked documents. The module boosts a document's ranking position if it has been SAT-Clicked in the session or in other sessions that share similar search topics. We apply K-means clustering algorithm to detect which sessions share similar search topics. Each unique term is one dimension of the vector and is weighted by its idf. We also apply session error analysis in RL3. From the query log, we first identify sessions with similar topics by clustering, then we use SAT-Clicks from most sessions to re-rank the documents for the sessions that the algorithm predicts as poorly issued sessions, i.e. more difficult session due to ill-form queries. Combining above approaches, we achieve a 20.9% nDCG@10 increment and a 13.0% P@10 increment from RL1 to RL2, and with utilization of the whole log data, we achieve a 4% nDCG@10 increment and a 0.5% P@10 increment from RL2 to RL3.

1. Introduction

Session search involves multiple search iterations triggered by query reformulations to accomplish a complex search task. In our groups' 2013 work [1], we model this interactive process of session search as a MDP process. In our 2014 work [3][4], we model it as a POMDP process. TREC 2014 Session track Task 1 intends to test whether we can utilize user interactions with a search engine in a session to improve search accuracy. The task data includes log data of 1021 sessions. The log data of each session records a sequence of queries $q_1,q_2, \dots ,q_{n-1},q_n$ triggered by users, where q_n is the current query in the session. The log also contains retrieved ranking lists for each past query, q_1 to q_{n-1}. Finally the log data collects user-clicked documents/snippets and the dwell time that users spend on each clicked document. There are three subtasks, RL1, RL2 and RL3. RL1 ignores all information in the session log and only relies on the current query to retrieve results. RL2 uses only information from current session to retrieve. RL3 uses any information in the session log to retrieve.

We apply different technologies in each sub tasks. In RL1, we directly feed the last query of a session to Lemur Search Engine. The retrieval algorithm is set as Language Modeling with Dirichlet smoothing. The smoothing parameter mu is set as 5000. In RL2, we adopt QCM algorithm [1] where we combine all queries in a session to formulate effective structured queries. Each search term is assigned with a weight, which is calculated based on whether the term occurs in previous SAT-Clicked documents and whether the term is newly added or removed from previous query. Further more we decrease a previous query's weight if the query didn't bring any document, which leads to a SAT-Click from user. Finally we boost a document's ranking score if it has been SAT-Clicked in the session. In RL3, we also apply QCM and decrease query weight if no SAT-Click documents are retrieved by it. And we boost a document's ranking score if it is SAT-Clicked in sessions that belongs to the same or similar topic. We identify similar topics by clustering sessions based on query similarity using K-means clustering algorithm. Another tactic we applied in RL3 is to replacing bad session's retrieval results with good session's results whose search topic is similar. We evaluate session's performance based on user's click numbers.

We organize this paper as follow. We discuss each technical approach in detail from Section 2 to Section 7. In Section 8, we present our submissions and the evaluation results. In Section 9, we conclude our work.

2. Ad-hoc Retrieval Model (Ad-hoc)

Our RL1 approach directly uses the current query of each session as search terms. The retrieval algorithm is Language Modeling with Dirichlet smoothing [2]. The document d's relevance score towards a search term t is calculated by formula:

$$P(t|q) = \frac{tf(t,d) + \mu P(t|C)}{length(d) + \mu}$$

where length(d) is document d's length. P(t|C) is the probability that term t appears in corpus C. μ is the Dirichlet smoothing parameter and is set to 5000 in our experiment.

3. The Query Change Retrieval Model (QCM)

In session, users modify queries to better express their information needs. In Query Change Retrieval Model (QCM) [1], query changes are considered as relevance feedback to adjust query term weights. First, it defines $\Delta q_i = q_i - q_{i-1}$ as the query change between two adjacent queries, q_{i-1} and q_i. Then Δq_i is divided into three parts: the added terms ($+\Delta q_i$) the removed terms ($-\Delta q_i$) and the theme terms (q_{theme}).

Table 1 A Query Change Example (TREC 2014 Session 52)

Session	Queries	Query Change	Q_{theme}
Session52	q_1 = hydropower efficiency q_2 = hydropower environment q_3 = hydropower damage	$+\Delta q_2$ = environment $-\Delta q_2$ = efficiency $+\Delta q_3$ = damage $-\Delta q_3$ = environment	hydropower

Table 1 presents an example of query changes in TREC 2014 Session track. In query q_i, query term weights are adjusted based on four types of strategies, W_{Theme}, $W_{Add,In}$, $W_{Add,Out}$ and W_{Remove} [1]. The relevance score between query q_i and a document d becomes:

$$Score(q_i\,, d) = \log \mathrm{P}(q_i|d) + \alpha W_{Theme} - \beta W_{Add,In} + \varepsilon W_{Add,Out} - \delta W_{Remove}$$

Parameters α, β, ε and δ are the linear weighting coefficients for each type of strategies. They are set as α=2.2, β=1.8, ε=0.07 and δ=0.4 in our submission. The QCM model combines all queries in a session using formula:

$$Score_{qcm}(q_{1..n}, d) = \sum_{i=1}^{n} \gamma^{n-i}\, Score(q_i, d)$$

where γ is the discount factor for the prior queries in the session. In TREC Session track's setting, evaluation is based on the whole session. The prior queries are equally important as current query, hence we set γ as 1 in our experiment.

4 Weighted QCM

QCM allows documents that are relevant to any query in a session to appear in the final retrieval list. When set parameter γ = 1, we combine all queries in a session evenly. However we argue that queries shouldn't be evenly combined. Here we define two concepts, **Strong SAT-Clicked** document and **Weak SAT-Clicked** document. Strong SAT-Clicked document means a retrieved document that has been clicked by a user and he/she dwelled more than 30 seconds on this document. Weak SAT-Clicked document is also a clicked document but with dwell time more than 10 seconds and less than 30 seconds.

We assume that dwell time on a clicked document indicates how relevant that document is. If a query doesn't bring any document that leads to a SAT-Click from the user, it indicates that this query is poor formed. Hence these queries' weight should be decrease. Weighted QCM combine queries based on query quality. Poor formed queries' weight is decreased by a factor $\omega \in (0, 1)$. Its score function is:

$$Score_{wqcm}(q_{1..n}, d) = \sum_{q_i \in Q_{good}} Score_{qcm}(q_i, d) + \omega \sum_{q_j \in Q_{bad}} Score_{qcm}(q_j, d)$$

Q_{good} is the query set in which every query brings at least one SAT-Click from users. While Q_{bad} is the query set in which every query brings zero SAT-Click from users. The current query is an exception. It brings zero SAT-Click because it has no retrieval results yet, however it belongs to Q_{good}.

5 User-Click Model

Since SAT-Click indicates a document's relevance, we boost a document's ranking score, if it is SAT-Clicked by users.

5.1 Session Level User-Click Model

In this approach, we only use information in the current session. We boost a document's ranking score if it has been SAT-Clicked in the current session. The score function is:

$$Score_{session-click}(q_{1..n}, d) = Score_{qcm}(q_{1..n}, d) + Score_{session-boost}(q_{1..n}, d)$$

$$Score_{session-boost}(q_{1..n}, d) = \frac{\psi|StrongSATClicks_d| + \theta|WeakSATClicks_d|}{\sum_{d_i \in session}(\psi|StrongSATClicks_{d_i}| + \theta|WeakSATClicks_{d_i}|)}$$

Where $|StrongSATClicks_d|$ is the number of times that document d is strongly SAT-Clicked in the current session. $|WeakSATClicks_d|$ is the number of times that d is weakly SAT-Clicked. The boosting score is normalized by the total number of SAT-Clicks in the session. We experimentally set ψ=2 and θ=1.

5.2 Topic Level User-Click Model

This approach is similar to the Session Level User-Click Model. The difference is that instead of only using the information in the current session, we utilize information in all sessions that share similar search topics. We cluster sessions based on their search topics. The cluster algorithm is described in detail in Section 6. We boost a document's ranking score if it has been SAT-Clicked in sessions that share similar search topics with the current session. The score function is:

$$Score_{cluster-click}(q_{1..n}, d) = Score_{qcm}(q_{1..n}, d) + Score_{cluster-boost}(q_{1..n}, d)$$

$$Score_{cluster-boost}(q_{1..n}, d) = \frac{\psi|StrongSATClicks_d| + \theta|WeakSATClicks_d|}{\sum_{d_i \in Cluster}(\psi|StrongSATClicks_{d_i}| + \theta|WeakSATClicks_{d_i}|)}$$

Where Cluster is a set of sessions that share similar search topics with the current session. $|StrongSATClicks_d|$ is the number of times that document d is strongly SAT-Clicked in the Cluster. $|WeakSATClicks_d|$ is the number of times that d is weakly SAT-Clicked. The boosting score is normalized by the total number of SAT-Clicks in the Cluster. We also set ψ=2 and θ=1.

6 Clustering

We cluster sessions based on search topics by comparing queries' similarity between different sessions.

- First, we combine all queries in one session and convert it into a term vector. Each unique search term is one dimension of the vector.
- Then, we assign terms' idf value as weight to each term dimension.
- Finally, we cluster sessions based on the Euclidean distance of their query vectors.

We use K-means clustering algorithm and set K as 60, which is the number of distinctive topic ids in the log file. This number may not be obtainable in a real search environment. We can train it or choose a relatively large K in such situation. Other clustering algorithms without requirement of predetermination of cluster numbers could be other alternatives, however we didn't explore them in our experiments.

7 Session Performance Prediction and Replacement

We detect a specific schema in sessions that share similar search topics, most of which contain SAT-Clicks, however a few do not. It indicates that for the few sessions, the bad retrieval results may be caused due to ill formed queries rather than difficult search tasks. For these sessions, we replace their retrieval results with good session's results whose search topic is similar.

Table 2 Features Extracted from Session Data Log

Feature	Definition
F_1	The user's intent of session s is to make comparison among two or more items.
F_2	The user did not click any retrieved document in session s.
F_3	$t_{dwell} \leq 5$s.
F_4	# of unique terms in the session $s \geq 20$.
F_5	$t_{dwell\_per\_click} < \frac{t^{(3)}_{dwell\_per\_click}}{2}$
F_6	Session s does not contain the most frequent term in $T(s)$.
F_7	# of unique terms in session $s \leq 6$
F_8	$\#\ of\ SAT\ clicks\ in\ session\ s < \frac{\sum_{s' \in T(s)} \#\ of\ SAT\ clicks\ in\ session\ s'}{\lvert T(s) \rvert}$

In order to identify good sessions from bad sessions automatically, we extract eight features from session click data log. For convenience, we introduce some symbols firstly. For each session s, let us use t_{dwell} to denote the user's total dwell time in the whole session and calculate the average dwell time $t_{dwell\_per\_click}$ as

$$t_{dwell\_per\_click} = \frac{t_{dwell}}{\#\text{ of clicks in session } s}.$$

Then all the average dwell times are sorted in a descending order,

$$t^{(1)}_{dwell\_per\_click},\ t^{(2)}_{dwell\_per\_click},\ t^{(3)}_{dwell\_per\_click},\ \cdots$$

Moreover, we use $T(s)$ to represent the topic cluster including s. Based on the above symbols, all the features can be listed in Table 2.

Here, F_1 is set up to deal with a shortcoming of QCM. According to our experience, when applying QCM to session search, the nDCG scores are often small in case that the user try to compare several items in one session. For example, the user may want to compare different infant developmental milestones depending on culture through posing a query like "culture difference in milestones in 0-12 month olds". This is an example from the 111th session in Session Track 2014. We treat one session as this kind when the queries include terms with patterns like "compare", "differ", "versus", "vs" and "v.s.".

All the eight features are Boolean, i.e. should be TRUE or FALSE. For each feature F_i (i=1,2,…,8), we count the number of sessions satisfying F_i=TRUE. For each session s, an estimation score $score_e(s)$ is calculated as follows:

$$score_e(s) = \sum_{i=1}^{8} \frac{1}{\#\ of\ sessions\ satisfying\ F_i = \text{TRUE}} * I(F_i)$$

where $I(F_i)$ is an indicator function. It returns 1 if session s satisfies feature F_i, otherwise it returns 0. All the sessions are ranked according to their estimation scores. The top 1/3 sessions are regarded as bad sessions and the others are regarded as good sessions.

8 Experiments

8.1. Data preparation

We run our experience on dataset Clueweb12 CatA. It consists of 733,019,372 English web pages, collected between February 10, 2012 and May 10, 2012. Spam documents are filtered out based on their Waterloo Spam scores.

8.2. Submission

Table 3 TREC 2014 Session Track Submissions

	GUS14RUN1	GUS14RUN2	GUS14RUN3
RL1	• Ad-hoc Retrieval Model	• Ad-hoc Retrieval Model	• Ad-hoc Retrieval Model
RL2	• Weighted QCM (ω=0.65) • Session Level User-Click Model	• Weighted QCM (ω=0.8) • Session Level User-Click Model	• Weighted QCM (ω=0.8) • Session Level User-Click Model
RL3	• Weighted QCM (ω=0.65) • Topic Level User-Click Model	• Weighted QCM (ω=0.8) • Topic Level User-Click Model	• Weighted QCM (ω=0.8) • Topic Level User-Click Model using topic ids • Session Performance Prediction and Replacement

Table 4 nDCG@10 and P@10 for top 100 sessions

	GUS14RUN1		GUS14RUN2		GUS14RUN3		Max		Med	
	nDCG@10	P@10	nDCG@10	P@10	nDCG@10	P@10	nDCG@10	P@10	nDCG@10	P@10
RL1	0.2053	0.378	0.2053	0.378	**0.2053**	**0.378**	0.3890	0.629	0.1549	0.348
RL2	0.2458	0.426	0.2482	0.427	**0.2482**	**0.427**	0.4865	0.712	0.1626	0.372
RL3	0.2443	0.423	0.2458	0.424	**0.258**	**0.429**	0.5111	0.744	0.1790	0.404

Table 3 lists our submissions in TREC 2014 Session Track and their configurations. We submit three runs in total: GUS14RUN1, GUS14RUN2 and GUS14RUN3. Each run contains three ranking lists, one for task RL1, one for task RL2 and one for task RL3.

It is worthwhile to point out that in GUS14RUN3 task RL3, we apply Topic Level User-Click differently. Here we did not using clustering to determine sessions that share similar search topics, instead we directly apply topic id in the log file to determine session topic's similarity. By doing this we can evaluate the effectiveness of applying the clustering method in Session Search. Further when we apply Session Performance Prediction and Replacement, we also use topic id to determine session clusters. We don't use clustering to determine session clusters, because clustering is based on comparing query similarity. If the queries are similar, then the retrieval performance should be close too. Hence it is difficult to find a good session to replace bad sessions when sessions are clustered by query similarity.

8.3. Results

Table 4 shows the evaluation results of our submissions. The result shows that by utilizing current session information, we achieve a 20.9% nDCG@10 increment and a 13.0% P@10 increment from RL1 to RL2, and with utilization of the whole log data, we achieve a 4% nDCG@10 increment and a 0.5% P@10 increment from RL2 to RL3. All submissions achieve a significant performance improvement from RL1 to RL2, however only GUS14RUN3 achieves a small improvement from RL2 to RL3. It may be caused by the features of Session track tasks. The search tasks are relatively complex. There are rich interactions in the session to help search engine to infer user intent. However there are few similar sessions can be used to recommend good documents for the current session. GUS14RUN2 RL3 and GUS14RUN3 RL3's performances are close, which suggests that clustering sessions by query similarity is as good as directly using topic ids. GUS14RUN3's RL3 gets highest P@10 scores in 20 sessions out of first 100 sessions. It proves that our approaches are highly effective.

9. Conclusion

We apply a combination of several technologies to TREC 2014 Session track. We achieve a significant performance boost from RL1 to RL2, and a small improvement from RL2 to RL3. The evaluation results suggest that 1) considering previous queries and the current query is suitable for session search task; 2) user SAT-Clicks is useful to estimate query quality and document relevance; 3) clustering sessions by query similarity is effective; 4) in session search, a session itself contains rich interaction information which can be used to improve search accuracy.

10. Acknowledgement

The research is supported by NSF grant CNS-1223825 and DARPA Memex program. This material is based on research sponsored by DARPA under agreement number FA8750-14-2-0226. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

11. References

[1]. Dongyi Guan, Sicong Zhang, and Hui Yang. 2013. Utilizing query change for session search. In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval (SIGIR '13). ACM, New York, NY, USA, 453-462.

[2]. 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.

[3]. Jiyun Luo, Sicong Zhang, Hui Yang. Win-Win Search: Dual-Agent Stochastic Game in Session Search. In Proceedings of the 37th Annual ACM SIGIR Conference (SIGIR 2014). Gold Coast, Australia.

[4] Sicong Zhang, Jiyun Luo, Hui Yang. A POMDP Model for Content-Free Document Re-ranking (short paper). In Proceedings of the 37th Annual ACM SIGIR Conference (SIGIR 2014). Gold Coast, Australia .