UMASS at TREC 2014 Session Track

Jiepu Jiang, James Allan
Center for Intelligent Information Retrieval,
School of Computer Science,
University of Massachusetts Amherst
jpjiang@cs.umass.edu, allan@cs.umass.edu

ABSTRACT
In our participation of the TREC 2014 session track, we adopt a learning to rank approach. The ranking features include ad hoc search as well as user's preference of query terms from implicit feedback information. We model user's preference of query terms by considering: query reformulation, SERP browsing, and reading of clicked webpages. We describe our approach in this report.

Keywords
Session; relevance feedback; implicit feedback.

1. METHODS
Let s be a session with a sequence of user queries $\{q_1, q_2, \dots, q_n\}$ and suppose we know user interaction information from q_1 to q_{n-1}. The task of the session track is to retrieve results for the last query q_n using information of the session s and other sessions. Here we do not use information from other sessions and only focus on the session itself.

We adopt a learning to rank approach. The ranking features include both ad hoc search as well as user's preference of query terms using the following implicit feedback information:

- Query reformulation. Query reformulation involves changes of query terms. Such changes may indicate user's preference of terms. We consider changes of query terms in the lastest query reformulation as well as in the whole session.
- SERP browsing. When browsing result snippets in a SERP, the user may click some snippets while skipping others. The decision to click may be made based on terms in the snippets. We assume that the user prefers terms in clicked snippets over skipped ones.
- Reading result webpages. After clicking a webpage, the user may spend time reading details if it is useful (satisfactory click), or quickly close the webpage if useless. We use result dwelltime as an indicator of whether a result is a satisfactory (SAT) click or a dissatisfactory (DSAT) one. We assume that user prefers terms in SAT clicks over DSAT ones.

We re-rank results of ad hoc search using these features. The rest of the report describes the approach and evaluates results.

2. FEATURES
2.1 Baseline Feature
For a session s and a document d, the baseline ranking feature is the query likelihood score (in log form) between the document and the current query q_n, i.e., $\log P(q_n|d)$. This is our baseline run. We use Dirichlet smoothing when estimating document models. The parameter is trained to optimize nDCG@10 in the TREC 2013 session track data.

Feature	Explaination
QL_currq	$\log P(q_n\|d)$

2.2 Past Queries
Previous studies [5, 6, 12] show that past queries are useful information for improving search performance of the current query.

Here we use two types of features based on past queries. The first one is to use past queries as positive relevance feedback. We estimate a query model $P(w|\theta_q)$ as in Equation (1). Then, we use $\log P(\theta_q|d)$ as ranking features, as in Equation (2). This feature is referred to as FixInt_λ where λ is the parameter in Equation (1). We use λ = 0.2 (FixInt_0.2) and λ = 0.5 (FixInt_0.5) because they can lead to optimized nDCG@10 and instance recall, respectively, in the 2013 dataset [8].

The second approach is to directly use the query likelihood scores of past queries as features. We consider the average, maximum, and minimum scores in log form. The following table shows the features considering past queries.

$$\hat{P}(w|\theta_q) = (1-\lambda)\cdot\frac{c(w,q_n)}{|q_n|} + \lambda\cdot\frac{\sum_{i=1}^{n-1} c(w,q_i)}{\sum_{i=1}^{n-1}|q_i|} \quad (1)$$

$$\log P(\theta_q|d) = \sum_{w\in\theta_q} P(w|\theta_q)\log P(w|d) \quad (2)$$

Feature	Explanation
FixInt_0.20	$\log P(\theta_q\|d)$, $\lambda = 0.2$
FixInt_0.50	$\log P(\theta_q\|d)$, $\lambda = 0.5$
QL_pastq_avg	avg $P(q_i\|d)$, $i < n$
QL_pastq_max	max $P(q_i\|d)$, $i < n$
QL_pastq_min	min $P(q_i\|d)$, $i < n$

2.3 Query Reformulation
Query reformulation involves changes of query terms (e.g. query term addition, removal, and substitution [4, 13]). Guan et al. [2] and Zhang et al. [14] utilized such changes as relevance feedback. Here we also adopt such changes as features.

We first consider the latest query reformulation from q_{n-1} to q_n. We consider the following types of query terms:

- Added terms (w_{add}): terms in q_n that do not appear in q_{n-1}.
- Removed terms (w_{rmv}): terms in q_{n-1} but are removed in q_n.
- Retained terms (w_{retain}): terms in both q_{n-1} and q_n.

For each type of query terms, we use the average log probability of the terms from the document model as features. If a type of query term does not exist, its feature value is set to 0.

Similarly, we consider the following types of terms regarding the differences in q_n comparing to all its previous queries:

- Session-wide new terms ($w_{new\_session}$): terms in q_n that do not appear in any of the previous queries.
- Session-wide recovered terms ($w_{rec\_session}$): terms once removed but added back to q_n.
- Session-wide removed terms ($w_{rmv\_session}$): terms appear in any of the previous queries but are removed in q_n.
- Session-wide retained terms ($w_{retain\_session}$): terms appear in each query.

Table 1. UMASS runs and the ranking Features. Each row is a set of features and each column is a run. "Y" means the run in the column used the feature in the row.

Features / Runs	All RL1	UMASS1		UMASS2		UMASS3		UMASS4	
		RL2	RL3	RL2	RL3	RL2	RL3	RL2	RL3
Current Query	Y	Y	Y	Y	Y	Y	Y	Y	Y
Past Queries		Y	Y						
Last Query Reformulation			Y	Y	Y				
Session-wide Query Reformulation			Y		Y				
Clicked Snippets		Y	Y			Y	Y		
Skipped Snippets			Y				Y		
SAT Clicks, dwell time > 30s		Y	Y					Y	Y
DSAT Clicks, dwell time < 15s			Y						Y

The following table shows ranking features considering query term changes in query reformulation.

Feature	Explaination
add_avg	avg log $P(w_{\text{add}}\|d)$
rmv_avg	avg log $P(w_{\text{rmv}}\|d)$
retain_avg	avg log $P(w_{\text{retain}}\|d)$
session_new_avg	avg log $P(w_{\text{new session}}\|d)$
session_rec_avg	avg log $P(w_{\text{rec session}}\|d)$
session_rmv_avg	avg log $P(w_{\text{rmv session}}\|d)$
session_retain_avg	avg log $P(w_{\text{retain session}}\|d)$

2.4 Clicked and Skipped Snippets

Previous studies [9, 10] show that clicked results are more likely relevant comparing to skipped results. Here we adopt similar heuristics but only focus on the snippets of results shown in each SERP. We assume users click some result snippets but skip some others because the user prefers words in the clicked snippets over the skipped ones. For a list of ranked result snippets $\{n_1, n_2, \ldots, n_m\}$, we also assume the user skipped n_i in SERP browsing if the user did not click n_i and there exists a clicked snippet n_j at a lower rank.

We first consider the top words in clicked snippets. Let N_{clicked} be the set of clicked snippets in the session prior to q_n. We estimate $P(w|N_{\text{clicked}})$ based on the contents of the snippets as in Equation (3), where n_i refers to the content of each snippet. We adopt the top 5, 10, and 15 words by $P(w|N_{\text{clicked}})$ and use the weighted log probability of these words from the document as features (referred to as clicked_snippet_top$k$):

$$\text{clicked\_snippet\_top}k = \sum_{w \in \text{top k words in N}_{\text{clicked}}} P(w \mid N_{clicked}) \cdot \log P(w \mid d)$$

$$\hat{P}(w \mid N_{clicked}) = \frac{1}{|N_{clicked}|} \sum_{n_i \in N_{clicked}} P(w \mid n_i) \quad (3)$$

Further, we model the difference between the clicked and skipped snippets using a mixture model approach. We assume words in the skipped snippets are generated from a linear mixture model of $P(w|N_{\text{clicked}})$ and $P(w|N_{\text{skipped}})$ as Equation (4). Since $P(w|N_{\text{clicked}})$ is known, the rest of the job is to estimate $P(w|N_{\text{skipped}})$. We estimate $P(w|N_{\text{skipped}})$ using an EM-algorithm as follows:

$$w \sim (1-\alpha) \cdot P(w \mid N_{skipped}) + \alpha \cdot P(w \mid N_{clicked}) \quad (4)$$

E-step
$$tf^{(n+1)}(w) = tf(w) \cdot \frac{(1-\alpha) \cdot P^{(n)}(w \mid N_{skipped})}{(1-\alpha) \cdot P^{(n)}(w \mid N_{skipped}) + \alpha \cdot P(w \mid N_{clicked})}$$

M-step
$$P^{(n+1)}(w \mid N_{skipped}) = \frac{tf^{(n+1)}(w)}{\sum_{w_i} tf^{(n+1)}(w_i)}$$

We set the parameter α to 0.5 intuitively. $P(w|N_{\text{skipped}})$ is supposed to model the difference between the skipped snippets and clicked ones. We also adopt the top 5, 10, and 15 words by $P(w|N_{\text{skipped}})$ and use the weighted log probability of these words from the document as features (referred to as skipped_snippet_top$k$). The following table shows features considering clicked and skipped snippets.

Feature	Explaination
clicked_snippet_top5	$\sum_{w \in \text{top k in N}_{\text{clicked}}} P(w \mid N_{clicked}) \cdot \log P(w \mid d)$
clicked_snippet_top10	
clicked_snippet_top15	
skipped_snippet_top5	$\sum_{w \in \text{top k in N}_{skipped}} P(w \mid N_{skipped}) \cdot \log P(w \mid d)$
skipped_snippet_top10	
skipped_snippet_top15	

2.5 SAT and DSAT Clicks

Similar to the approach in Section 2.4, we can estimate $P(w|C_{\text{SAT}})$ and $P(w|C_{\text{DSAT}})$ from the content of SAT and DSAT click results. We use click dwell time as an indicator of whether the the click is satisfactory (longer than 30s) or not (shorter than 15s). The same threshold was adopted in previous studies [1, 3]. Still, we select top words from each model and use the weighted log probability of the words from the document as ranking features.

Feature	Explaination
SAT_click_top5	$\sum_{w \in \text{top k in C}_{\text{SAT}}} P(w \mid C_{SAT}) \cdot \log P(w \mid d)$
SAT_click_top10	
SAT_click_top15	
DSAT_click_top5	$\sum_{w \in \text{top k in C}_{DSAT}} P(w \mid C_{DSAT}) \cdot \log P(w \mid d)$
DSAT_click_top10	
DSAT_click_top15	

3. RUNS

Table 1 shows our runs and features. All runs have the same RL1 results (using query likelihood model on the current query). Each run uses a different combination of implicit feedback features in RL2 and RL3 results.

We first generate ranking candidates using the top 1000 results of query likelihood model on the current query. Then, we re-rank the results using ranking models trained on the TREC 2013 session track data. We use RankLib [11] for ranking and adopt linear regression as the ranking model. We also filter results using the Waterloo spam filter scores (results with lower than 70 scores are removed from the ranking list).

4. EVALUATION

Table 2 shows evaluation results of the submitted runs. Results show that result snippet features can improve the baseline ranking (UMASS3). However, many results are surprising and conflict with previous findings. For example, previous studies found that query term change and clicks can improve ranking [2, 8, 14], but our runs using these two types of features led to worse than baseline performance (UMASS2 and UMASS4). We suspect this is mainly

due to our using linear regression for ranking (we adopt it simply because it is fast). We are currently exploring these unexpected results to better understand their source. Besides, as user behavior can be distinct in different types of tasks [7], it is worth examining the contribution of features in different types of tasks.

Table 2. nDCG@10 of the submitted runs.

Features	Runs	nDCG@10
baseline (QL)	*.RL1	0.1630
+ query reformulation (last query)	UMASS2.RL2	0.1496
+ query reformulation (last + session wide)	UMASS2.RL3	0.1349
+ clicked snippets	UMASS3.RL2	0.1832
+ clicked & skipped snippets	UMASS3.RL3	0.1832
+ SAT clicks	UMASS4.RL2	0.1353
+ SAT & DSAT clicks	UMASS4.RL3	0.1414
+ past query & clicked snippets & SAT clicks	UMASS1.RL2	0.1714
+ all features	UMASS1.RL3	0.1702
TREC median	RL1	0.1549
	RL2	0.1626
	RL3	0.1790

5. REFERENCES

[1] Paul N. Bennett, Ryen W. White, Wei Chu, Susan T. Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. 2012. Modeling the impact of short- and long-term behavior on search personalization. In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval (SIGIR '12), 185-194.

[2] 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), 453-462.

[3] Ahmed Hassan. 2012. A semi-supervised approach to modeling web search satisfaction. In Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval (SIGIR '12), 275-284.

[4] Jeff Huang and Efthimis N. Efthimiadis. 2009. Analyzing and evaluating query reformulation strategies in web search logs. In Proceedings of the 18th ACM conference on Information and knowledge management (CIKM '09), 77-86.

[5] Jiepu Jiang, Daqing He, Shuguang Han. 2012. On Duplicate Results in a Search Session. In Proceedings of the 21st Text REtrieval Conference (TREC 2012).

[6] Jiepu Jiang, Shuguang Han, Jia Wu, Daqing He. 2011. Pitt at TREC 2011 session track. In Proceedings of the 20th Text REtrieval Conference (TREC 2011).

[7] Jiepu Jiang, Daqing He, and James Allan. 2014. Searching, browsing, and clicking in a search session: changes in user behavior by task and over time. In Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval (SIGIR '14), 607-616.

[8] Jiepu Jiang, Daqing He. 2013. Pitt at TREC 2013: Different Effects of Click-through and Past Queries on Whole-session Search Performance. In Proceedings of the 22nd Text REtrieval Conference (TREC 2013).

[9] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately interpreting clickthrough data as implicit feedback. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '05), 154-161.

[10] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in Web search. ACM Transactions on Information Systems (TOIS), 25(2).

[11] RankLib. http://sourceforge.net/p/lemur/wiki/RankLib/.

[12] Xuehua Shen, Bin Tan, and ChengXiang Zhai. 2005. Context-sensitive information retrieval using implicit feedback. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '05), 43-50.

[13] Xuanhui Wang and ChengXiang Zhai. 2008. Mining term association patterns from search logs for effective query reformulation. In Proceedings of the 17th ACM conference on Information and knowledge management (CIKM '08), 479-488.

[14] Sicong Zhang, Dongyi Guan, and Hui Yang. 2013. Query change as relevance feedback in session search. In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval (SIGIR '13), 821-824.