BUPT_PRIS at TREC 2012 Session Track

Chuang Zhang, Xiaotian Wang, Songlin Wen, Runze Li
Pattern Recognition and Intelligent System Lab,
Beijing University of Posts and Telecommunications, P.R.China

Abstract

In this paper, we introduce our experiments carried out at TREC 2012 session track. Based on the work of our group in TREC 2011 session track, we propose several methods to improve the retrieval performance by considering the user behavior information over the session, which includes use query expansion based on meta data, query expansion based on click order, optimization based on history ranked lists and so on. The results show that some methods can really improve the search performance and some methods need to be optimized.

1. Introduction

The TREC Session track ran for the third time in this year, and its goal for this year is to test whether systems can improve their performance for a given query by using previous queries and user interactions with the retrieval system (including clicks on ranked results, dwell times, etc.) [1]. Based the sessions, there are four tasks in TREC 2011 session track: run the retrieval system:
RL1: only using the current query.
RL2: using the current query and the set of past queries in the session.
RL3: using the current query, the set of past queries in the session and the ranked lists of URLs
RL4: using the current query, the set of past queries in the session, the ranked lists of URLs, the clicked URLs and the time spent on the clicked documents.
RL1 retrieval effectiveness is viewed as the basic standard. By comparing RL1 with the effectiveness of RL2, RL3, RL4, I can evaluate whether the retrieval system can use previous queries and user interactions to improve the search performance.

2. Experiment setup

In our experiment, we choose Category B comprising 50 million documents as the search dataset.
Indri search is the search engine for the search process in our experiment. Indri search service for ClueWeb09 collection is available on the web. The service enables the user to submit the queries and obtain top documents returned by Indri search engine. Query expansion and term weighting can be applied in Indri search.
Spam Rankings data provided by University of Waterloo include the spam scores of the web pages in ClueWeb09 collection. In our experiment, the web pages with spam score less than 40 are viewed as spam and filtered out from the final search results. We use spam ranking filter to do this

operation.

The anchor log for the Cluweb09 collection has been processed and made available on the web. We use the anchor log for the Category B of 43 million lines. Each line in the log file presents a document in the collection with anchor text of the document.

3. Design

We submitted three runs. The design of the three runs is shown in the table below.

Table1: Design of the three runs with methods

System	Run1	Run2	Run3
RL1	1. Spam ranking filter	1. Spam ranking filter 2. Re-rank by PageRank score	1. Spam ranking filter 2. VSM similarity model
RL2	1. User behavior model 2. Spam ranking filter	1. User behavior model 2. Spam ranking filter 3. Re-rank by PageRank score and indri score	1. User behavior model 2. Spam ranking filter 3. VSM similarity model
RL3	1. Anchor log model 2. Spam ranking filter	1. Optimization Based on History Ranked Lists 2. Spam ranking filter	1. Query expansion based on meta data 2. Spam ranking filter
RL4	1. Anchor log model 2. User behavior model considering the attention time 3. Spam ranking filter	1. Query expansion based on clicked titles and snippets 2. Spam ranking filter	1. Query expansion based on click order 2. Spam ranking filter

Different combinations of the methods are used in each run to test the optimization performance of the methods.

4. Methods

4.1 User behavior model

User behavior model can do the query expansion and term weighting by considering the users' behavior in the session [2]. The detail process is shown as follows.

Assume $q_i = (t_1, t_2, \dots t_n)$ is the i_{th} query in one search session, and t_j is the j_{th} term of q_i.S_i

represents the set of history queries of the i_{th} user behavior.

Thus, $S_1 = \{ q_i \}$, $S_2 = S_1 \cup \{q_2\}$, and $S_{i-1} = S_{i-2} \cup \{q_{i-1}\} = (t_1^{'}, t_2^{'}, \dots, t_m^{'})$. The query expansion and term weighting is realized in the following process.

1. The weight of term t_j is set to $\frac{1}{n}$ for the new query $q_i = (t_1, t_2, \dots t_n)$.And $\sum_{i=1}^{n} \frac{1}{n} = 1$.
2. $(e_1, e_2, \dots e_m)$ is the weight vector of the terms for the history query set $S_{i-1} = (t_1^{'}, t_2^{'}, \dots, t_m^{'})$. And $\sum_{i=1}^{m} e_i = 1$.
3. The query set is expanded as $S_i = S_{i-1} \cup \{q_i\}$ and the normalized expanded term weights :

$$d \sum_{i=1}^{m} e_i + (1-d) \sum_{i=1}^{n} \frac{1}{n} = 1$$

d is the attenuation factor， and d < 0.5.I choose 0.4 as the attenuation factor in my project.

4. Assume there are k terms appearing both in the new query and in the previous query:

$$S_{i-1} \cap q_i = (t_1^{'}, t_2^{'}, \dots t_k^{'}) = (t_1, t_2, \dots t_k) \ \ k \le m \text{ and } k \le n$$

The query set S_i is expanded as : $S_i = S_{i-1} \cup \{q_i\} = (t_1^{'} \dots t_k^{'}, t_{k+1}^{'} \dots t_m^{'}, t_{k+1} \dots t_n)$

Finally, the term weighs $(e_1^{'}, e_2^{'}, \dots, e_m^{'})$ are assigned as the following functions show

$$e_i^{'} = \begin{cases} de_i + (1-d)\frac{1}{n} & i \in [1, k] \\ de_i & i \in [k+1, m] \\ (1-d)\frac{1}{n} & i \in [m+1, n] \end{cases}$$

4.2. Anchor log model

University of Essex developed a method for extracting useful terms and phrases to expand the reformulated query in Session Track 2010 [3]. Based on that, we modified it to adapt to the new requirements.

We retrieve the anchor texts of the documents in ranked lists for past queries. And we extract the top ten terms to expand the query terms. The weight of original query terms is set to 0.7 and the terms from anchor log has the weight value of 0.3.

After the stop word filtering, query is expanded in the following form:

$$\begin{array}{l} \#\text{combine}(\\ 0.7\#\text{combine}(r_c) \\ 0.3\#\text{combine}(e_1 e_2 \dots e_{10})\ ) \end{array}$$

r_c is the current query and e_j is the j_{th} anchor log expansion term. Finally, submit the expanded query to Indri search to get the new search results.

4.3.Optimization based on history ranked lists

In Session Track 2010, University of Lugano proposed a method of generating optimized ranked list of current query by the rank of documents in ranked list of current query and only one past query [4]. Based on it we developed one improved method to do n-1 iterative procedures for n-1 past queries and the current query. Finally we can get the scores of optimized result lists and sorting them in ascending order:

Assume the returned ranked lists for the past queries are RL_1, RL_2, RL_3…RL_n, where RL_n is the ranked list of the last past query. The ranked list we need to calculate for the current query is RL_{n+1}. We can re-rank the documents in ranked list of RL_{n+1} by considering the ranked lists of RL_1,RL_2…$RL_{n.}$ The final ranked list current query is denoted as FinalRL.

If there are some past queries:

There are ranked lists RL1, RL2, RL3…RLn for past query we need to calculate.

For any document in RL1 and RL2 denoted as document i:

If document i of RL2 appears in RL1, score[i] = 1/rl2[i] + 0.2(1/rl2[i] - 1/rl1[i]); If document i of RL2 does not appear in RL1, score[i] = 1/rl2[i]; If document i of RL1 does not appear in RL2, score[i] = -1.

We get the ranked list TEMP1-2 according to the score in descending order.

Then we do the iteration until we get the score of the ranked list for current query. Finally, we can get the ranked list of FinalRL according to the score.

4.4.Query expansion based on meta data

This method uses the meta tags in the documents of ranked lists for past queries to do the query expansion. We collect the terms in "keyword" and "description" data in meta tag. Then we extract the top 10 terms with highest frequency to expand the query.

The query becomes

$$\begin{array}{l}\#\text{combine}(\\ (1-\text{d})\#\text{combine}(\text{r}_c) \\ \text{d}\#\text{combine}(\text{e}_1\text{e}_2 \ldots \text{e}_{10})\ \ )\end{array}$$

r_c is the current query and e_j is the j_{th} meta data expansion term. d is the attenuation factor which is between 0 and 1. We use the data of TREC 2011 session track to test which value assigned to d can achieve the best performance. By using session data and the relevance judgments for TREC 2011 session track, we find that when d = 0.5 the search results achieve the highest relevance score.

4.5.Query expansion based on click order

We think the click order can reflect the attraction of the documents titles to the user. In this model we use the clicked titles to query expansion by considering the click order.

1. The term weight of current query is set to $w_{current}$, so (1- $w_{current}$) is assigned to the expanded terms.

2. Assume that there are n history queries. RL1, RL2...RLn are the ranked lists of the history queries.

Assume the user click m titles, denoted as (RL, click order k, title), such as:

1, 1, title1
1, 2, title2
……
1, m, $title_m$

a) Assign the weight to the current query and the expanded terms :

weight of RL_k = (1-$w_{current}$)*(k/1+2+3+4+…n)

weight_RL1=(1- $w_{current}$) *(1/1+2+3+4+…n) weight_RL2=(1- $w_{current}$)*(2/1+2+3+4+…n)

b) Assign the weight of expanded terms to the terms in clicked titles:

weight_RL1_title$_k$= weight_RL1 * (m+1-k/1+2+3+4+…m), where k is the click order

the weight of title1 in RL1: weight_RL1_title1= weight_RL1 * (m+1-1/1+2+3+4+…m)

the weight of title2in RL1: weight_RL1_title2= weight_RL1 * (m+1-2/1+2+3+4+…m)

By doing this in turn we can get the weight of all the titles.

In our experiment, we set $w_{current}$ = 0.5.

4.6.User behavior model considering the attention time

The attention time of the clicked documents can reflect the usefulness of the information in the document as conceived by the user. Songhua Xu etc[5] proposed the attention time prediction algorithm in 2008. Based on that, we build the model using the dwelling time of the clicked documents to calculate the documents relevance level and re-rank the documents.

1. For the k_{th} Clicked document C_{ik} in session i ,t_{inter} represents the dwelling time interval. t_{offset} represents the time offset. t_{att} denotes the attention time on the document.

$$t_{inter}\,(C_{ik}) = (t_{end}\ - t_{start}\,) * d_c$$

$$t_{offset}\,(C_{ik}) = \frac{2\exp\,(-d*rank\,(C_{ik}))}{1+\exp\,(-d*rank\,(C_{ik}))}$$

$$t_{att}\,(C_{ik}) = \ t_{inter}\,(C_{ik})t_{offset}\,(C_{ik})$$

rank(C_{ik}) is the rank number of the k_{th} Clicked document in session i.

We set the control parameter d_c= 0.1 to make the interval small, and d= 0.2 controlling the drop off.

2. The j_{th} document attention time in prediction is calculated :

$$t_{predict}\ = sim\big(d_{ij}, C_{ik}\big) * t_{att}\,(C_{ik})$$

And re-rank the documents considering the prediction attention time.

4.7.Query expansion based on clicked titles and snippets

This method uses the titles and snippets for the clicked web pages for past queries to do the query expansion. We collect the terms of titles and snippets data in clicked web pages. Then we extract the top 30 terms to expand the query. The terms of current query have weight 0.6, and the terms extracted from titles and snippets have weight 0.4.

5. Results

This year we submit three runs for the four tasks (RL1, RL2, RL3, RL4). Table2 shows the relevance scores of our runs with ndcg@10 and nerr@10. Different from Session 2011, relevance for TREC 2012 session track is defined against the entire topic and not against different subtopics.

Table2: results of three runs

Run	**wildcat1**	**wildcat2**	**wildcat3**

RL1.ndcg@10	**0.2177**	**0.0844**	**0.2068**
RL2.ndcg@10	**0.2130**	**0.1338**	**0.1947**
RL3.ndcg@10	**0.2715**	**0.2121**	**0.2876**
RL4.ndcg@10	**0.2567**	**0.2692**	**0.2608**
RL1.nerr@10	**0.2610**	**0.1156**	**0.2419**
RL2.nerr@10	**0.2546**	**0.1682**	**0.2297**
RL3.nerr@10	**0.3257**	**0.2540**	**0.3231**
RL4.nerr@10	**0.317**	**0.3213**	**0.3144**

By analyzing the results, we have the following findings:

1. For wildcat1 and wildcat2, the relevance score of RL2 is less than that of RL1. We use user behavior model in RL2, which improved the search performance efficiently for 2011 session track, so user behavior model does not work well for the session data of this year. For the session data of this year, many topics contain more than one session. We think this change may be one reason for the bad performance of user behavior model. We may need to do some adjustment to make this method fit the session data of this year better.
2. For all the three runs scores of RL3 and RL4 are higher than those of RL1 and RL2. This indicates that by considering the previous interactions in session the system can improve the search performance.
3. wildcat1.RL3 and wildcat3.RL3, where we use anchor log model and query expansion based on meta data, achieve high scores. This indicates that anchor log data and meta data of the previous ranked lists are useful for system to predict the intention of the users.
4. wildcat2.RL1 and wildcat2.RL2, where we use PageRank score to re-rank the search results, perform badly. This situation may indicate that re-ranking the results by only considering PageRank scores are not appropriate for the tasks of session track. The relevance between the results and the user intention also should be considered.
5. The scores of Query expansion based on clicked titles and snippets in wildcat2.RL4 and Query expansion based on click order wildcat3.RL4 in are lower than the scores of RL3. This indicates that we do not use the click data very efficiently. These two methods can improve search performance, but they need to be optimized in the future work.

References

[1] http://ir.cis.udel.edu/sessions/guidelines.html

[2] Hongtao Chen (2008). Research and application of user search behavior.

[3] M-Dyaa Albakour, Udo Kruschwitz, Jinzhong Niu, Maria Fasli. Autoadapt at the Session track in TREC 2010. University of Essex

[4] Mostafa Keikha,Parvaz Mahdabi, Shima Gerani. University of Lugano at TREC 2010. University of Lugano

[5] Songhua Xu, Yi Zhu, Hao Jiang and Francis C.M. Lau. A User-Oriented Webpage Ranking Algorithm Based on User Attention Time. Zhejiang University, Yale University, The University of Hong Kong