Search Result Re-ranking by Feedback Control Adjustment for Time-sensitive Query

Ruiqiang Zhang[†] and **Yi Chang**[†] and **Zhaohui Zheng**[†]
Donald Metzler[†] and **Jian-yun Nie**[‡]
[†]Yahoo! Labs, 701 First Avenue, Sunnyvale, CA94089
[‡]University of Montreal, Montreal, Quebec,H3C 3J7, Canada
[†]{ruiqiang,yichang,zhaohui,metzler}@yahoo-inc.com
[‡]nie@iro.umontreal.ca

Abstract

We propose a new method to rank a special category of time-sensitive queries that are year qualified. The method adjusts the retrieval scores of a base ranking function according to time-stamps of web documents so that the freshest documents are ranked higher. Our method, which is based on feedback control theory, uses ranking errors to adjust the search engine behavior. For this purpose, we use a simple but effective method to extract year qualified queries by mining query logs and a time-stamp recognition method that considers titles and urls of web documents. Our method was tested on a commercial search engine. The experiments show that our approach can significantly improve relevance ranking for year qualified queries even if all the existing methods for comparison failed.

1 Introduction

Relevance ranking plays a crucial role in search engines. There are many proposed machine learning based ranking algorithms such as language modeling-based methods (Zhai and Lafferty, 2004), RankSVM (Joachims, 2002), RankBoost (Freund et al., 1998) and GBrank (Zheng et al., 2007). The input to these algorithms is a set of feature vectors extracted from queries and documents. The goal is to find the parameter setting that optimizes some relevance metric given training data. While these machine learning algorithms can improve *average* relevance, they may be ineffctive for certain special cases. Time-sensitive queries are one such special case that machine-learned ranking functions may have a hard time learning, due to the small number of such queries.

Consider the query "sigir" (the name of a conference), which is time sensitive. Table 1 shows two example search result pages for the query, SERP1 and SERP2. The ranking of SERP2 is clearly better than that of SERP1 because the most recent event, "sigir2009", is ranked higher than other years.

query: sigir	
SERP1	url1: http://www.sigir.org
	url2: http://www.sigir2008.org
	url3: http://www.sigir2004.org
	url4: http://www.sigir2009.org
	url5: http://www.sigir2009.org/schedule
SERP2	url1: http://www.sigir.org
	url2: http://www.sigir2009.org
	url3: http://www.sigir2009.org/schedule
	url4: http://www.sigir2008.org
	url5: http://www.sigir2004.org

Table 1: Two contrived search engine result pages

Time is an important dimension of relevance in web search, since users tend to prefer recent documents to old documents. At the time of this writing (February 2009), none of the major commercial search engines ranked the homepage for SIGIR 2009 higher than previous SIGIR homepages for the query "sigir". One possible reason for this is that ranking algorithms are typically based on anchor text features, hyperlink induced features, and click-through rate features. However, these features tend to favor old pages more than recent ones. For example, "sigir2008" has more links and clicks than "sigir2009" because "sigir2008" has existed longer time and therefore has been visited more. It is less likely that newer web pages from "sigir2009" can be ranked higher using features that implicitly favor old pages.

However, the fundamental problem is that current approaches have focused on improving general ranking algorithms. Methods for improving ranking of specific types of query like temporal queries are often overlooked.

Aiming to improve ranking results, some methods of re-ranking search results are proposed, such as the work by (Agichtein et al., 2006) and (Teevan et al., 2005).

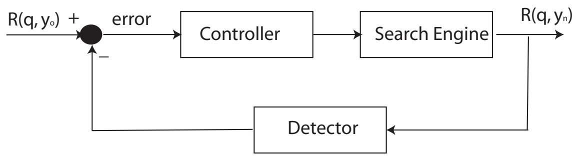


Figure 1: Feedback control for search engine

These work uses user search behavior information or personalization information as features that are integrated into an enhanced ranking model. We propose a novel method of re-ranking search results. This new method is based on feedback control theory, as illustrated in 1.

We make a Detector to monitor search engine (SE) output and compare it with the input, which is the desired search engine ranking. If an error is found, we design the controller that uses the error to adjust the search engine output, such that the search engine output tracks the input. We will detail the algorithm in Section 4.1.

Our method was applied to a special class of time-sensitive query, *year qualified queries* (YQQs). For this category, we found users either attached a year with the query explicitly, like "sigir 2009", or used the query only without a year attached,like "sigir". We call the former explicit YQQs, and the latter implicit YQQs. Using query log analysis, we found these types of queries made up about 10% of the total query volume. We focus exclusively on implicit YQQs by translating the user's implicit intention as the most recent year. Explicit YQQs are less interesting, because the user's temporal intention is clearly specified in the query. Therefore, ranking for these types of queries is relatively straightforward. Throughout the remainder of this paper, we use the "YQQ" to refer to implicit YQQs, unless otherwise stated.

2 Adaptive score adjustment

Our proposed re-ranking model is shown in Eq. 1, as below.

$$
\begin{aligned}
F(q,d) &= \begin{cases} R(q,d) & \texttt{if } q \notin \texttt{YQQ} \\ R(q,d) + Q(q,d) & \texttt{otherwise} \end{cases} \\
Q(q,d) &= \begin{cases} (e(d_o,d_n) + k)e^{\lambda\alpha(q)} & \texttt{if } y(d) = y_n \\ 0 & \texttt{otherwise} \end{cases} \\
e(d_o,d_n) &= R(q,d_o) - R(q,d_n)
\end{aligned} \tag{1}
$$

This work assumes that a base ranking function is used to rank documents with respect to an incoming query. We denote this base ranking function as $R(q,d)$. This ranking function is conditioned on a query q and a document d. It is assumed to model the relevance between q and d.

Our proposed method is flexible for all YQQ queries. Suppose the current base ranking function gives the results as SERP1 of Table 1. To correct the ranking, we propose making an adjustment to $R(q,d)$.

In Eq. 1, $F(q,d)$ is the final ranking function. If the query is not an YQQ, the base ranking function is used. Otherwise, we propose an adjustment function, $Q(q,d)$, to adjust the base ranking function. $Q(q,d)$ is controlled by the ranking error, $e(d_o,d_n)$, signifying the base function ranking error if the newest web page d_n is ranked lower than the oldest web page d_o. $y(d)$ is the year that the event described by d has occurred or will occur. If y_o and y_n indicate the oldest year and the newest year, then $y(d_o) = y_o, y(d_n) = y_n$. $R(q,d_o)$ and $R(q,d_n)$ are the base ranking function scores for the oldest and the newest documents.

k is a small shift value for direction control. When $k < 0$, the newest document is adjusted slightly under the old one. Otherwise, it is adjusted slightly over the old one. Experiments show $k > 0$ gave better results. The value of k is determined in training.

$\alpha(q)$ is the confidence score of a YQQ query, meaning the likelihood of a query to be YQQ. The confidence score is bigger if a query is more likely to be YQQ. More details are given in next section. λ is a weighting parameter for adjusting $\alpha(q)$.

The exp function $e^{\lambda\alpha(q)}$ is a weighting to control boosting value. A higher value, confidence α, a larger boosting value, $Q(q,d)$.

Our method can be understood by feedback control theory, as illustrated in Fig. 1. The ideal input is $R(q,y_o)$ representing the desired ranking score for the newest Web page, $R(q,y_n)$. But the search engine real output is $R(q,y_n)$. Because search engine is a dynamic system, its ranking is changing over time. This results in ranking errors, $e(d_o,d_n) = R(q,d_o) - R(q,d_n)$. The function of "Controller" is to design a function to adjust the search engine ranking so that the error approximates to zero, $e(d_o,d_n) = 0$. For this work, "Controller" is $Q(q,d)$. "Detector" is a document year-stamp recognizer, which will be described more in the next section. "Detector" is used to detect the newest Web pages and their ranking scores. Fig. 1 is an ideal implementation of our methods. We cannot carry out real-time experiments in this work. Therefore, the calculation of ranking errors was made in offline training.

3 YQQ detection and year-stamp recognition

To implement Eq. 1, we need to find YQQ queries and to identify the year-stamp of web documents.

Our YQQ detection method is simple, efficient, and relies only on having access to a query log with frequency information. First, we extracted all explicit YQQs from query log. Then, we removed all the years from explicit YQQs. Thus, implicit YQQs are obtained from explicit YQQs. The implicit YQQs are saved in a dictionary. In online test, we match input queries with each of implicit YQQs in the dictionary. If an exact match is found, we regard the input query as YQQ, and apply Eq. 1 to re-rank search results.

After analyzing samples of the extracted YQQs, we group them into three classes. One is recurring-event query, like "sigir", "us open tennis"; the second is news-worthy query, like "steve ballmer", "china foreign reserves"; And the class not belong to any of the above two, like "christmas", "youtube". We found our proposed methods were the most effective for the first category. In Eq. 1, we can use confidence $\alpha(q)$ to distinguish the three categories and their change of ranking as shown in Eq.1, that is defined as below.

$$\alpha(q) = \frac{\sum_y w(q,y)}{\#(q) + \sum_y w(q,y)} \tag{2}$$

where $w(q,y) = \#(q.y) + \#(y.q)$. $\#(q.y)$ denotes the number of times that the base query q is post-qualified with the year y in the query log. Similarly, $\#(y.q)$ is the number of times that q is pre-qualified with the year y. This weight measures how likely q is to be qualified with y, which forms the basis of our mining and analysis. $\#(q)$ is the counts of independent query, without associating with any other terms.

We also need to know the year-stamp $y(d)$ for each web document so that the ranking score of a document is updated if $y(d) = y_n$ is satisfied. We can do this from a few sources such as title, url, anchar text, and extract date from documents that is possible for many news pages. For example, from url of the web page, "www.sigir2009.org", we detect its year-stamp is 2009.

We have also tried to use some machine generated dates. However, in the end we found such dates are inaccurate and cannot be trusted. For example, discovery time is the time when the document was found by the crawler. But a web document may exist several years before a crawler found it. We show the worse effect of using discovery time in the experiments.

4 Experiments

We will describe the implementation methods and experimental results in this section. Our methods include offline dictionary building and online test. In offline training, our first step is to mine YQQs. A commercial search engine company provided us with six months of query logs. We extracted a list of YQQs using Section 3's method. For each of the YQQs, we run the search engine and output the top N results. For each document, we used the method described in Section 3 to recognize the year-stamp and find the oldest and the newest page. If there are multiple urls with the same yearstamp, we choose the first oldest and the first most recent. Next,we calculated the boosting value according to Eq. 1. Each query has a boosting value. For online test, a user's query is matched with each of the YQQs in the dictionary. If an exact match is found, the boosting value will be added to the base ranking score iff the document has the newest yearstamp.

For evaluating our methods, we randomly extracted 600 YQQs from the dictionary. We extracted the top-5 search results for each of queries using the base ranking function and the proposed ranking function. We asked human editors to judge all the scraped results. We used five judgment grades: Perfect, Excellent, Good, Fair, and Bad. Editors were instructed to consider temporal issues when judging. For example, sigir2004 is given a worse grade than sigir2009. To avoid bias, we advised editors to retain relevance as their primary judgment criteria. Our evaluation metric is relative change in DCG, $\%\Delta_{dcg} = \frac{DCG_{proposed} - DCG_{baseline}}{DCG_{baseline}}$, where DCG is the traditional Discounted Cumulative Gain (Jarvelin and Kekalainen, 2002).

4.1 Effect of the proposed boosting method

Our experimental results are shown in Table 2, where we compared our work with the existing methods. While we cannot apply (Li and Croft, 2003)'s approach directly because first, our search engine is not based on language modeling; second, it is impossible to obtain exact timestamp for web pages as (Li and Croft, 2003) did in the track evaluation. However, we tried to simulate (Li and Croft, 2003)'s approach in web search by using the linear integration method exactly as the same as(Li and Croft, 2003) by adding a time-based function with our base ranking function. For the timestamp, we used discovery time in the time-based function. The parameters (λ, α) have the exact same meaning as in (Li and Croft, 2003) but were tuned according to our base ranking function. With regards to the approach by (Diaz and Jones, 2004), we ranked the web pages in decreasing order of discovery time. Our own approaches were tested under options with and without using adaptation. For no adaption, we let the e of Eq.1 equal to 0, meaning no score difference between the oldest document and the newest document was captured, but a constant value was used. It is equivalent to an open loop in Fig.1. For adaption, we used the ranking errors to adjust the base ranking. In the Table we used multiple ks to show the effect of changing k. Using different k can have a big impact on the performance. The best value we found was $k = 0.3$. In this experiment, we let $\alpha(q) = 0$ so that the result responds to k only.

Our approach is significantly better than the existing methods. Both of the two existing methods produced worse results than the baseline, which shows the ap-

Li & Croft	(λ, α)=(0.2,2.0)	-0.5
	(λ, α)=(0.2,4.0)	-1.2
Diaz & Jones		-4.5^*
No adaptation ($e = 0$, open loop)	k=0.3	1.2
	k=0.4	0.8
Adaptation (closed loop)	k=0.3	6.6^*
	k=0.4	6.2^*

Table 2: $\%\Delta_{dcg}$ of proposed method comparing with existing methods.A sign "$*$" indicates statistical significance (p-value<0.05)

λ	0	0.2	0.4	0.6	0.8	1.0
$\%\Delta_{dcg}$	6.6^*	7.8^*	8.4^*	4.5	2.1	-0.2^*

Table 3: Effect of confidence as changing λ.

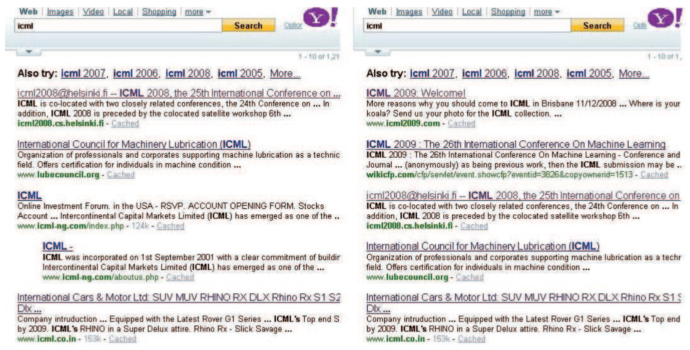
proaches may be inappropriate for Web search. Not surprisingly, using adaption achieved much better results than without using adaption. Thus, these experiments prove the effectiveness of our proposed methods.

Another important parameter in the Eq.1 is the confidence score $\alpha(q)$, which indicates the confidence of query to be YQQ. In Eq. 1, λ is used to adjusting $\alpha(q)$. We observed dcg gain for each different λ. The results are shown in Table 3. The value of λ needs to be tuned for different base ranking functions. A higher λ can hurt performance. In our experiments, the best value of 0.4 gave a 8.4% statistically significant gain in DCG. The $\lambda = 0$ setting means we turn off confidence, which results in lower performance. Thus, using YQQ confidence is effective.

5 Discussions and conclusions

In this paper, we proposed a novel approach to solve YQQ ranking problem, which is a problem that seems to plague most major commercial search engines. Our approach for handling YQQs does not involve any query expansion that adds a year to the query. Instead, keeping the user's query intact, we re-rank search results by adjusting the base ranking function. Our work assumes the intent of YQQs is to find documents about the most recent year. For this reason, we use YQQ confidence to measure the probability of this intent. As our results showed, our proposed method is highly effective. A real example is given in Fig. 2 to show the significant improvement by our method.

Our adaptive methods are not limited to YQQs only. We believe this framework can be applied to any category of queries once a query classification and a score detector have been implemented.


Web | Images | Video | Local | Shopping | more
icml
Search
Optior
1 - 10 of 1,21
Web | Images | Video | Local | Shopping | more
icml
Search
Optic
1 - 10 of 1,

Also try: icml 2007, icml 2006, icml 2008, icml 2005, More...

icml2008@helsinki.fi -- ICML 2008, the 25th International Conference on ...
ICML is co-located with two closely related conferences, the 24th Conference on ... In addition, ICML 2008 is preceded by the colocated satellite workshop 6th ...
icml2008.cs.helsinki.fi - Cached

International Council for Machinery Lubrication (ICML)
Organization of professionals and corporates supporting machine lubrication as a technic field. Offers certification for individuals in machine condition ...
www.lubecouncil.org - Cached

ICML
Online Investment Forum. in the USA - RSVP. ACCOUNT OPENING FORM. Stocks Account ... Intercontinental Capital Markets Limited (ICML) has emerged as one of the ..
www.icml-ng.com/index.php - 124k - Cached

ICML -
ICML was incorporated on 1st September 2001 with a clear commitment of buildir Intercontinental Capital Markets Limited (ICML) has emerged as one of the ...
www.icml-ng.com/aboutus.php - Cached

International Cars & Motor Ltd: SUV MUV RHINO RX DLX Rhino Rx S1 S2 Dlx ...
Company intruduction ... Equipped with the Latest Rover G1 Series ... ICML's Top end S by 2009. ICML's RHINO in a Super Delux attire. Rhino Rx - Slick Savage ...
www.icml.co.in - 153k - Cached

Also try: icml 2007, icml 2006, icml 2008, icml 2005, More...

ICML 2009: Welcome!
More reasons why you should come to ICML in Brisbane 11/12/2008 ... Where is your koala? Send us your photo for the ICML collection. ...
www.icml2009.com - Cached

ICML 2009 : The 26th International Conference On Machine Learning
ICML 2009 : The 26th International Conference On Machine Learning - Conference and Journal ... (anonymously) as being previous work, then the ICML submission may be ..
wikicfp.com/cfp/servlet/event.showcfp?eventid=3826&copyownerid=1513 - Cached

icml2008@helsinki.fi -- ICML 2008, the 25th International Conference on
ICML is co-located with two closely related conferences, the 24th Conference on ... In addition, ICML 2008 is preceded by the colocated satellite workshop 6th ...
icml2008.cs.helsinki.fi - Cached

International Council for Machinery Lubrication (ICML)
Organization of professionals and corporates supporting machine lubrication as a techr field. Offers certification for individuals in machine condition ...
www.lubecouncil.org - Cached

International Cars & Motor Ltd: SUV MUV RHINO RX DLX Rhino Rx S1 S Dlx ...
Company intruduction ... Equipped with the Latest Rover G1 Series ... ICML's Top end by 2009. ICML's RHINO in a Super Delux attire. Rhino Rx - Slick Savage ...
www.icml.co.in - 153k - Cached

Figure 2: Ranking improvement for query ICML by our method: before re-rank(left) and after(right)

References

Eugene Agichtein, Eric Brill, and Susan Dumais. 2006. Improving web search ranking by incorporating user behavior information. In *SIGIR '06*, pages 19–26.

Fernando Diaz and Rosie Jones. 2004. Using temporal profiles of queries for precision prediction. In *Proc. 27th Ann. Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 18–24, New York, NY, USA. ACM.

Yoav Freund, Raj D. Iyer, Robert E. Schapire, and Yoram Singer. 1998. An efficient boosting algorithm for combining preferences. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 170–178.

Kalervo Jarvelin and Jaana Kekalainen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20:2002.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142.

Xiaoyan Li and W. Bruce Croft. 2003. Time-based language models. In *Proc. 12th Intl. Conf. on Information and Knowledge Management*, pages 469–475, New York, NY, USA. ACM.

Jaime Teevan, Susan T. Dumais, and Eric Horvitz. 2005. Personalizing search via automated analysis of interests and activities. In *SIGIR '05*, pages 449–456.

Chengxiang Zhai and John Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214.

Zhaohui Zheng, Keke Chen, Gordon Sun, and Hongyuan Zha. 2007. A regression framework for learning ranking functions using relative relevance judgments. In *SIGIR '07*, pages 287–294.