Anticipatory Search: Using Context to Initiate Search

Daniel J. Liebling, Paul N. Bennett, Ryen W. White
Microsoft Research
Redmond, WA 98052
{ danl, pauben, ryenw }@microsoft.com

ABSTRACT

Identifying content for which a user may search has a variety of applications, including ranking and recommendation. In this poster, we examine how pre-search context can be used to predict content that the user will seek before they have even specified a search query. We call this *anticipatory search*. Using a log-based approach, we compare different methods for predicting the content to be searched using different attributes of the pre-query context and behavioral signals from previous visitors to the most recent browse URL. Each method covers different cases and shows promise for query-free anticipatory search on the Web.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *search process, selection process.*

Keywords

Anticipatory search; Pre-search context; Web search.

1. INTRODUCTION

Personalization using recent pre-query context (within session) can improve result quality when combined with existing search result ranking methods [8]. Others have explored methods of biasing search via query expansion and re-ranking results for queries identified as good candidates for contextualization [6]. However, these methods rely on a user issuing a query, limiting their scope.

An alternative approach is to use the context to *generate* queries. Y!Q contextual search generated query augmentations from Web page content and had a “query-less” mode [5] but never presented an evaluation of that mode. Implicit Query [1] studied a similar problem but for a desktop environment using e-mails and other user activity to create queries and proactively find related documents. Henzinger et al. [4] focused on automatically retrieving information alongside live (television) news streams by extracting keywords. However, no one has studied the problem of performing query-less search to anticipate user needs in general Web search and presented a principled evaluation of their methods.

In this work, we attempt to anticipate information needs, and directly predict the results for which users will next search and visit given only the current browse URL. Queries submitted by users following Web browsing have been shown to be related to the most recently browsed document [7]. Therefore, for any URL for likely to be followed by a query (estimated via transitions observed in logs), we aim to predict the page that the user will visit from the result page for that query. This can both help the user in removing the need to type a query as well as potentially provide higher utility by issuing a more effective query than the user may formulate to retrieve the same result. Anticipatory search is a more challenging problem than URL recommendation since the goal is to not only provide URLs of interest but to identify those that the user would have tried to find — a much smaller set.

We describe and evaluate three methods: (1) a behaviorally-focused method targeting exact-match context, (2) a method that targets higher recall by combining content with a backoff of behavioral context, and (3) a method that combines content with exact-matching behavior context. In addition, we also propose a unique way of evaluating anticipatory search using queries issued to a Web search engine. We now describe each of the methods.

2. SUGGESTING RELATED CONTENT

The data for the study come from users’ browse behavior logged anonymously through a popular Web browser’s instrumentation over two weeks in November 2011. One week of data is used for aggregation and one week for evaluation. We consider (*browse URL*, *browse title*, *query*, *click URL*) tuples. The *browse URL* is the URL that the user visited prior to initiating a search with the given *query*. The *browse title* is the text which appears in the title bar of the browser. The *click URLs* are all links returned by the search engine for the query that the user clicks. Here, we restrict these to “satisfied” clicks, defined as having a dwell time of ≥ 30 seconds, an effective indicator of satisfaction [3]. Further, we consider only data where the domain of the clicked URL differs from that of the browsed URL. We believe the cross-domain problem is more difficult and more informative.

2.1 Exact–Match Behavioral Context

Method 1 builds a table of (*browse URL*, *click URL*, *Pr*(*click* | *browse*)) tuples using the logs. We use the first week of data to aggregate click URL probabilities conditioned on the browse URL. To help ensure reasonable data quality, we further filter the data to those that appeared for at least five different users, and have conditional probabilities of 0.1 or above. Sorting by probability yields the list of suggested URLs for a given browse URL.

To evaluate the method, we examined the (browse, query, click) tuples in a uniform sample of 100,000 tuples from the test set. For each browsed URL, we score the position of the user’s actual click within our sorted list of recommended URLs. For example, the browse URL *aol.sportingnews.com/ncaa-football* is observed in our test set. From our conditional click data we recommend *sportsillustrated.cnn.com/football/ncaa* as the most likely next URL visited since its conditional click probability is 0.13. Following this URL browse, we observe that the user issued the query *sports illustrated*, clicking on the same URL that Method 1 suggests, yielding a reciprocal rank (RR) of 1.0. RR is averaged over all tuples in the test set to compute the mean RR (MRR).

2.2 Content + Backoff Behavioral Context

Our second method uses the title of the browse URL and combines it with the conditional click probabilities of Method 1. We aggregate these click probabilities at the URL domain level, creating a table of (*browse domain*, *click domain*, *Pr*(*click domain* | *browse domain*)) tuples. The method automatically generates a query for each browse URL observed in the test week. The query text is the title of the page located at the browse URL (case folded and sans punctuation) plus a disjunction of “site:” operators which instruct the search engine to restrict results to Web pages on the domain supplied as an argument to the operator. For example, if we observe the browse URL *www.golfdiscount.com/ping-putters* titled *Putters*, and in the prior week found the distribution of click domains for *golfdiscount.com* to be {(*bizrate.com*, 0.14), (*global-*

SIGIR’12, August 12–16, 2012, Portland, Oregon, USA.
ACM 978-1-4503-1472-5/12/08.

golf.com, 0.14), (*golfgalaxy.com*, 0.14), …} then our constructed query becomes [*putters* (*site:bizrate.com OR site:globalgolf.com OR site:golfgalaxy.com*)]. As in Method 1, we drop low probability domains, omitting the domain of the browse URL, if present. We then issue the constructed query to a search engine and retrieve at most the top 100 relevant results. This yields a (*browse URL*, *result set*) mapping. To evaluate the method, we examine each tuple in the sampled test set and find the rank position (if any) of the actual click URL in the results from our constructed query. Continuing the above example, we observe a user click on *www.golfgalaxy.com/putters/search* at position 2, for RR of 0.5.

2.3 Content + Exact–Match Behavior

The third method combines concepts from the previous two methods. To formulate the search query, we use only the normalized title of the browsed URL. As above, we issue the constructed query and retrieve results. We then combine the conditionally-clicked URLs from Method 1 with the results from this query by using a stable sort on the results using the click probability when available. That is, we prefer the rank from our click probability table to the rank of the search engine. The evaluation method is the same as Method 2. To illustrate, the browse URL *drugs.com/prednisone.html* is titled *Prednisone*. We use this term as our query, which returns *en.wikipedia.org/wiki/Prednisone* in position 6 for a RR of 0.17.

3. EVALUATION

Table 1 compares the MRR and recall relative to the 1,810 cases where at least one method suggested the clicked URL. All differences are significant by a Wilcoxon Signed Rank test (*p*=0.01).

Table 1. Coverage and overall accuracy.

Method	n (%) covered	Relative Recall	MRR
1	509 (28.1%)	0.281	0.222
2	973 (53.8%)	0.537	0.289
3	627 (34.6%)	0.346	0.098

Method 2 has both higher recall and precision than the other two methods. This may be because domain-level aggregation increases coverage, while the title and site: operators scope the results. Method 3 can be applied for almost any URL with a title, but its overall firing rate is therefore also high. Figure 1 shows the number of clicked recommended URLs per positional bin (zero-indexed and five rank positions per bin). As shown, Method 2 tends to return relevant results in the top rank positions, like all methods, tailing off exponentially. Methods 1 and 2 seem especially promising since the clicked URL is in top rank of one of the two nearly 40% of the time that is found by any method (with Method 2 accounting for nearly 60% of those). Though Methods 1 and 2 require observing URLs from a log, we believe aggregating by topic can produce similar results. Table 2 shows that that the overlap between the methods is only around 10-20%, suggesting that combining the methods could lead to marked coverage gains over any of singular method. Navigational queries performed poorly; clicked results were almost exclusively the navigational target whose rank was harmed by these methods.

Table 2. Overlap between successfully predicted browse URLs in each method. The diagonal gives the number of distinct successful browse URLs in each method.

	M1	M2	M3
M1	447	–	–
M2	109	922	–
M3	123	138	516

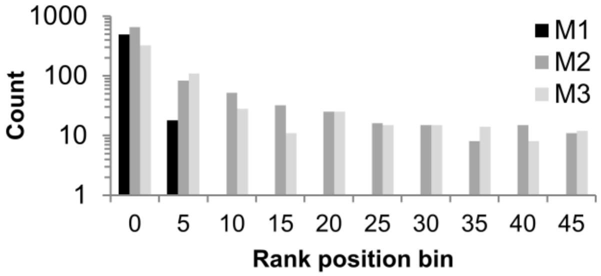

Figure 1**. Count of clicked recommended URLs by position.**

4. CONCLUSIONS

In this work we considered predicting the URL a user clicks from a set of search results. Instead of waiting to receive the user's query to generate the results, we used the pre-search browse URL coupled with existing clickthrough data and/or the browsed page title. We find that although coverage is low, the user's click URL most often appears in the top few results generated by our methods. Since these searches and clicks represent real information needs, one may surmise that the information need was not completely met by the browse URL prior to the search. Web content maintainers could use these URLs to find content that their site is not exposing, or expand their site content.

Search engines could leverage the pre-search context to present relevant URLs without requiring that the user issue a query. In practice, large advertising and search companies already cover much of the browsed Web via contextual ad matching, e.g., [1]. Future work will compare our methods with such models, increase coverage by exploiting method combinations, use other sources of pre-search context such as topic distributions and alternative page features, and examine whether recommendations other than the clicked URLs have utility to users. The goal is a learned selective application of anticipatory search capable of determining when pre-search context is relevant, increasing precision so that invocation happens such that users receive help only when needed.

REFERENCES

[1] Broder, A., Fontoura, M., Josifovski, V., and Riedl, L. (2007). A semantic approach to contextual advertising. *Proc. SIGIR*, 559-566.

[2] Dumais, S.T., Cutrell, E., Sarin, R., and E. Horvitz. (2004). Implicit queries for contextualized search. *Proc. SIGIR*, 594.

[3] Fox, S., Kuldeep, K., Mydland, M., Dumais, S., and White, T. (2005). Evaluating implicit measures to improve Web search. *ACM TOIS*, 23(2): 147-168.

[4] Henzinger, M., Chang, B., Milch, B., and Brin, S. (2005). Query-Free News Search. *World Wide Web: Internet and Web Information Systems*, 8(2):101-126.

[5] Kraft, R., Mahoul, F., and Chang, C.C. (2005). Y!Q: contextual search at the point of inspiration. *Proc. CIKM*, 816-823.

[6] Rahurkar, M. and Cucerzan, S. (2008). Using the current browsing context to improve search relevance. *Proc. CIKM*, 1493-1494.

[7] Rahurkar, M. and Cucerzan, S. (2008). Predicting when browsing context is relevant to search. *Proc. SIGIR*, 841-842.

[8] Teevan, J., Dumais, S.T., and Horvitz, E. (2005). Personalizing search via automated analysis of interests and activities. *Proc. SIGIR*, 449-456.

[9] Xiang, B., Jiang, D., Pei, J., Sun, X., Chen, E., and Li, H. (2010). Context-aware ranking in Web search. *Proc. SIGIR*, 451-458.