Retroactive Answering of Search Queries

Beverly Yang
Google, Inc.
byang@google.com

Glen Jeh
Google, Inc.
glenj@google.com

ABSTRACT

Major search engines currently use the history of a user's actions (e.g., queries, clicks) to personalize search results. In this paper, we present a new personalized service, *query-specific web recommendations* (QSRs), that retroactively answers queries from a user's history as new results arise. The QSR system addresses two important subproblems with applications beyond the system itself: (1) Automatic identification of queries in a user's history that represent standing interests and unfulfilled needs. (2) Effective detection of interesting new results to these queries. We develop a variety of heuristics and algorithms to address these problems, and evaluate them through a study of Google history users. Our results strongly motivate the need for automatic detection of standing interests from a user's history, and identifies the algorithms that are most useful in doing so. Our results also identify the algorithms, some which are counter-intuitive, that are most useful in identifying interesting new results for past queries, allowing us to achieve very high precision over our data set.

Categories and Subject Descriptors

H.3.4 [**Information Systems**]: Information Storage and Retrieval—*User profiles and alert services*

General Terms

Algorithms, Human Factors

Keywords

Personalized search, Recommendations, Automatic identification of user intent

1. INTRODUCTION

Major web search engines (e.g., Google [6], Yahoo [20]) have recently begun offering *search history* services, in which a user's search history – such as what queries she has issued and what search results she has clicked on – are logged and shown back to her upon request. Besides allowing a user to remind herself of past searches, this history can be used to help search engines improve the results of future searches by *personalizing* her search results according to preferences automatically inferred from her history (e.g., [9, 15, 18, 19]).

Current personalization services generally operate at a high-level understanding of the user. For example, references [15, 18] reorder search results based on general preferences inferred from a user's history. However, search history captures specific events and actions taken by a user, so it should also be possible to focus on and address *known, specific* user needs. To this end, we present *query-specific web recommendations* (QSRs), a new personalization service that alerts the user when interesting new results to selected previous queries have appeared.

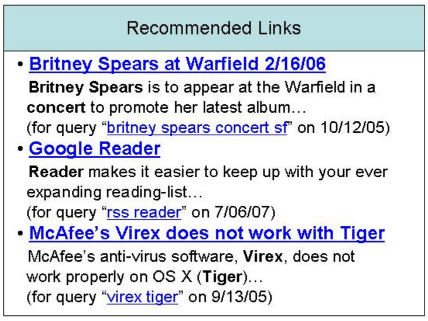
As an example of how QSRs might be useful, consider the query "britney spears concert san francisco." At the time the user issued the query, perhaps no good results existed because Britney was not on tour. However, a few months later when a concert arrives into town, the user could be automatically notified of the new websites advertising this concert. Essentially, the query is treated as a *standing query*, and the user is later alerted of interesting new results to the query that were not shown at the time the query was issued, perhaps because they were not available at that time, or were ranked lower. Since the new results are presented to the user when she is not actively issuing the search, they are effectively *web page recommendations* corresponding to specific past queries.

Obviously, not all queries represent standing interests or unfulfilled needs, so one important problem is how to identify queries that do. Some existing systems, such as Google's Web Alerts [7], allow users to explicitly specify queries for which they would like to be alerted when a new URL in the top-10 search results appears for the query. However, due to inconvenience and other factors, most users do not explictly register such queries: according to a user study conducted over 18 Google Search History users (Section 6.1), out of 154 past queries for which the users expressed a medium to strong interest in seeing further results, *none* of these queries were actually registered as web alerts! One of our major challenges is thus to *automatically* identify queries that represent standing interests.

Moreover, alerting the user of all changes to the search results for the query may cause too many uninteresting results to be shown, due to minor changes in the web or spurious changes in the ranking algorithm. Subjects from the same study indicate that Google's Web Alerts system suffers from these problems. A second challenge is thus to identify those new results that the user would be interested in.

In this paper, we present the QSR system for retroactively recommending interesting results as they arise to a user's past queries. The system gives rise to two important subproblems: (1) automatically detecting when queries represent standing interests, and (2) detecting when new interesting results have come up for these queries. We will present algorithms that address these problems, as well as the results of two user studies that show the effectiveness of our system. We note also that the subproblems studied here have applications beyond our system: for example, au-

WWW 2006, May 23–26, 2006, Edinburgh, Scotland.
ACM 1-59593-323-9/06/0005.



Recommended Links

- **Britney Spears at Warfield 2/16/06**
  Britney Spears is to appear at the Warfield in a **concert** to promote her latest album…
  (for query "britney spears concert sf" on 10/12/05)
- **Google Reader**
  Reader makes it easier to keep up with your ever expanding reading-list…
  (for query "rss reader" on 7/06/07)
- **McAfee's Virex does not work with Tiger**
  McAfee's anti-virus software, **Virex**, does not work properly on OS X (**Tiger**)…
  (for query "virex tiger" on 9/13/05)

Figure 1: Mockup of UI for recommended web pages

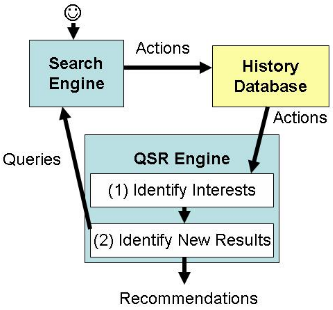


Figure 2: Architecture of QSR system

tomatic identification of standing interests in the form of specific queries can be especially valuable in ads targeting. Our contributions are summarized as follows:

- In Section 2 we describe the interface and architecture of the QSR recommendation system.
- In Section 3 we present our approach to the problem of automatically identifying standing interests from a user's history. We highlight the aspects of information need relevant to standing interests (e.g., prior fulfillment, interest duration), and describe a number of potentially useful *signals*, derived from a user's history, that can be used to identify standing interest.
- In Section 4 we discuss the problem of identifying interesting new web page results. We describe current Web Alert techniques and their potential deficiencies, and define a number of additional signals and techniques that can be used to better determine whether a new result is interesting.
- In Section 6, we present the results of our user study, in which 18 users of the Google Search History service were presented a sample of specific queries from their own history, and were asked to evaluate their level of fulfillment with the results. The purpose of this study was three-fold: (1) to motivate automatic identification of standing interests, (2) to demonstrate that it is possible to automatically detect standing interests from user history, and (3) to measure the accuracy of various signals in determining standing interests. The results of our study are promising, demonstrating clearly that automatic identification of standing interests is both important and possible.

  In the same section, we present the results of a second study, in which users were asked to evaluate the quality of the web page recommendations made over a set of queries from anonymous users – not necessarily their own. The main purpose of this study was to determine which techniques were most useful in determining whether new results are interesting. We find several surprising results – for example, that the rank of the new result is *inversely* related to how interesting it was perceived to be – and present general guidelines for selecting interesting results.

2. SYSTEM DESCRIPTION

The user-facing aspects of the QSR system are quite simple: a user performs queries on the search engine as usual. The search engine tracks the user's history, which is then fed into the QSR system. When the QSR system discovers an interesting new result for a past user query (one which was determined to represent a standing interest), it recommends the web page to her. Recommended web pages may be presented in a number of simple ways. For example, they may be packaged as an RSS feed, and displayed using the user's favorite RSS reader or other compatible interface. Recommendations can also be displayed alongside her search history, or they may even be displayed on the main search page. When a recommendation is displayed, we show both a link to the web page and the query for which the recommendation is made, so that users can recognize the context for the recommendation. A mock-up for fictitious web page recommendations is shown in Figure 1.

Figure 2 shows a high-level overview of the QSR system architecture, which is integrated with that of the search engine. The QSR engine periodically computes recommendations for a user in an offline process consisting of two steps: (1) identifying queries that represent standing interests, and (2) identifying new interesting results. In the first step, QSR will read a user's actions from the history database, and using heuristics described in Section 3, identify the top M queries that most likely represent standing interests. In the second step, QSR will submit each of these M queries to the search engine, compare the first 10 current results with the previous results seen by the user at the time she issued the query, and identify any new results as potential recommendations. QSR will then score each recommendation using heuristics described in Section 4. The top N recommendations according to this score are displayed to the user.

We limit the output of the first step to M queries for efficiency, as the computation of recommendations on each query requires reissuing the query to the search engine. It is possible that not all queries representing standing interests will be considered during one computation. However, given good heuristics, we will at least be able to address the most important queries at any given time. We also limit the output of the second stage to N recommendations, so as not to overwhelm a user with recommendations at any one time. In addition, because it is better to make *no* recommendations than it is to make many poor ones, our focus in both of these steps is on *precision* – selecting only interesting queries and results – rather than *recall.*

In the next two sections, we describe in detail how we approach the two steps in computing recommendations.

3. IDENTIFYING STANDING INTERESTS

The goal of our query-specific recommendation system is to recommend new web pages for users' old queries. However, no matter how good the new result is, a user will not find the recommendation meaningful unless she has a standing interest in that particular query. In this section, we define our notion of a standing interest, and then present a number of potential signals that can be used to automatically identify such interests.

```
html encode java (8 s)
    * RESULTCLICK (91.00 s) -- 2.  http://www.java2html.de/docs/api/de/java2html/util/HTMLTools.html
    * RESULTCLICK (247.00 s) -- 1.  http://www.javapractices.com/Topic96.cjp
    * RESULTCLICK (12.00 s) -- 8.  http://www.trialfiles.com/program-16687.html
    * NEXTPAGE (5.00 s) -- start = 10
         o RESULTCLICK (1019.00 s) -- 12.  http://forum.java.sun.com/thread.jspa?threadID=562942...
         o REFINEMENT (21.00 s) -- html encode java utility
                + RESULTCLICK (32.00 s) -- 7.  http://www.javapractices.com/Topic96.cjp
                     o NEXTPAGE (8.00 s) -- start = 10
                            * NEXTPAGE (30.00 s) -- start = 20
(Total time:  1473.00 s)
```

Table 1: Sample Query Session

3.1 Problem Definition

Different applications may focus on different types of needs and interests. For example, ads targeting may focus on unfulfilled user queries with commercial intentions (travel planning, online purchases, etc.). QSRs are general in that web pages can be meaningfully recommended for many kinds of user queries. For our purposes, we say that a user has a *standing interest* in a query if she *would be interested in seeing new interesting results.* There are a number of reasons a user may or may not have a standing interest in a query. For example:

1) Prior Fulfillment. Has the user already found a satisfactory result (or set of results) for her query?

2) Query Interest Level. What is the user's interest level in the query topic? If the user is very interested in the actress Natalie Portman, then she may be interested in seeing good recommendations for the query "natalie portman" even if she already found satisfactory results at the time of the query.

3) Need/Interest Duration. How *timely* is the information need? A user may be planning a vacation in Hawaii, and is performing many queries on local hotels, attractions, and history. Prior to his trip, he may be very interested in any good information he can get on the topic. After his trip, however, he no longer wishes to see any further results.

Given these intuitions, we would now like to determine the *signals* – properties of the query and associated events – that can help us to automatically identify prior fulfillment, interest level and duration of user needs.

Example. Let us consider the sample query session in Table 1. The user initially submitted the query `html encode java` – presumably to find out how to encode html in a java program. After 8 seconds of browsing the search results, she clicks on the second result presented, and remains viewing that page for 91 seconds. She then returns to the results page and views the first result for 247 seconds. Finally, she views the 8th result for 12 seconds. She then performs a *next page* navigation, meaning that she views the next page of results, starting at position 11. She views the 12th result for a long time – 1019 seconds. However, perhaps because she is still unable to find a satisfactory result, she submits the query refinement `html encode java utility` – she is explicitly looking for an existing java utility that will allow her to encode html. After a single result click for 32 seconds, the user looks at the next page of results ranked 11-20, and immediately looks at the following page of results ranked 21-30. She then ends the query session.

How can we determine whether the user found what she was looking for, and how interested she is in seeing new results? First, it would appear that the user was interested in finding an answer, since she spent a considerable amount of time in the session, viewed a number of pages, and performed a large number of refinements (query refinements, next pages, etc.). Second, we might also guess that the user did not find what she was looking for, since the session ended with her looking at a number of search results pages, but not actually clicking on anything. Finally, it is not as clear what the duration of the user's information need is. However, since this query topic seems to address a work-related need, we might guess that the user needs to find a solution immediately, or in the near future. Thus, from this one example we can see how one might determine information need with signals such as duration of the session, number of actions, ordering of actions, and so on.

Query Sessions. As the above example suggests, rather than focusing on individual queries, which may be related to one another, we consider *query sessions*, which we define as all actions associated with a given initial query. Such actions can include result clicks, spelling corrections, viewing additional pages of results, and query refinements. We define a query to be a *query refinement* of the previous query if both queries contain at least one common term. For the remainder of the paper, we will use the term *refinement* to more broadly refer to spelling corrections, next pages, and query refinements.

Because we evaluate a user's interest in a query session, rather than a specific query, once we have identified an interesting query session, we must determine the actual query to make recommendations for. A session may consist of many query refinements, so which should be used? Should we create a new query consisting of the terms appearing across multiple refinements? For the purposes of our initial prototype and user study, we use the query refinement which is directly followed by the largest number of result clicks. If two or more query refinements are tied, then we choose the refinement for which the total duration of clicks is longest. For example, in the query session shown in Table 1, we will register the query "html encode java" because it has four result clicks, while "html encode java utility" has only one.

Informal feedback from our user study (Section 6) suggests that this approach to query sessions works well in most, but not all, cases. We will continue to investigate alternate definitions of query sessions and query selection in future prototypes.

3.2 Signals

There is a large space of possible signals for identifying query interest. Rather than attempting to create a comprehensive set, here we list the signals which we found to be useful in our system, and briefly describe the intuition behind each one. In Section 6.2.1 we verify our intuitions with the actual results of a user study.

* **Number of terms** – A larger number of terms tends to indicate a more specific need, which in turn might correlate with shorter interest duration and lower likelihood of prior fulfillment.
* **Number of clicks** and **number of refinements** – The more actions a user takes on behalf of a query, the more interested she is likely to be in the query. In addition, a high number of refinements probably implies low likelihood of prior fulfillment.
* **History match** – If a query matches the interests displayed by a user through past queries and clicks, then interest level is probably high. A *history match score* may be generated in a number of ways, such as that described in [18].
* **Navigational** – A *navigational* query is one in which the user is looking for a specific web site, rather than information from a web page [17]. We assume that if the user clicks on only a single result and makes no subsequent refinements, the query is either navigational, or answerable by a single good website. In this case, there is a high likelihood of prior fulfillment and low interest level.
* **Repeated non-navigational** – If a user repeats a query over time, she is likely to be interested in seeing further results. Note, however, we must be careful to eliminate navigational queries which are often repeated, but for which the user does not care to see additional results. Therefore, we only consider a query that has been repeated, and for which the user has clicked on multiple or different clicks the most recent two times the query was submitted.

The signals above are ones that we found to be useful in identifying standing interests (Section 6.2.1). We have also tried a number of additional signals which we found – often to our surprise – not to be useful. Examples include the **session duration** (longer sessions might imply higher interest), the **topic** of the query (leisure-related topics such as sports and travel might be more interesting than work-related topics), the **number of long clicks** (users might quickly click through many results on a query she is not interested in, so the number of *long* clicks – where the user views a page for many seconds – may be a better indicator than the number of any kind of click), and whether the session **ended with a refinement** (this should only happen if the user wanted to see further results). A further discussion of these signals can be found in [21].

It is also important to note that any recommendation system like QSR will have **implicit user feedback** in the form of clicks on recommended links. After our system is launched, we will incorporate a feedback loop to refine and adjust our algorithms based on clickthrough data.

Interest Score. Using the scalar signals described, we would like to define an *interest score* for query sessions that captures the relative standing interest the user has in a session. We define the interest score as: `iscore` $= a \cdot \log(\#\text{ clicks} + \#\text{ refinements}) + b \cdot \log(\#\text{ repetitions}) + c \cdot (\text{history match score})$. We will show (Section 6.2.1) that higher `iscore` values correlate with higher user interest. Note that boolean signals (e.g., repeated non-navigational) are not incorporated into `iscore`, but can be used as filters.

4. DETERMINING INTERESTING RESULTS

Once we have identified queries that represent standing interests, we must address the problem of identifying interesting results to recommend to users as they arise. Recall that new results are detected when the QSR system periodically reissues the query to the search engine. While this problem is precisely that addressed by current web alert services, anecdotal evidence suggests there is room for improvement. For example, in our user study described in Section 5, 2 of the 4 subjects who had ever registered alerts mentioned that after they registered their first alert, they found that the recommendations were not interesting and did not feel compelled to use the system further. Thus it would seem that the acceptance of QSRs and the continued usage of existing web alert services require improved *quality* of recommendations. We say that a recommendation has high quality if it is interesting to the user – it does not necessarily imply that the page itself is good (e.g., high PageRank).

Rank	URL	PR Score	New
1	www.rssreader.com	3.93	No
2	blogspace.com/rss/readers	3.19	No
3	www.feedreader.com	3.23	No
4	**www.google.com/reader**	2.74	Yes
5	www.bradsoft.com	2.80	No
6	www.bloglines.com	2.84	No
7	www.pluck.com	2.63	No
8	sage.mozdev.org	2.56	No
9	www.sharpreader.net	2.61	No

Table 2: Top 10 results for `rss reader`

Example – Web Alerts. To motivate the signals useful in determining the quality of a recommendation, let us consider an example from Web Alerts. On October 16, 2005, an alert for the query "beverly yang," the name of one of the authors, returned the URL `http://someblog.com/journal/images/04/0505/` (domain name anonymized). The alert was generated based solely on the criterion that the result moved into the top 10 results for the query between October 15 and 16, 2005. Although this criterion often identifies interesting new results, in this case the author found the result uninteresting because she has seen the page before and it was not a good page – characteristics that could be determined by considering the user's history and information about the page itself, such as its rank, PageRank score, etc.

Another factor that could be taken into account is whether the appearance of the result in the top 10 is due to there being new information on or about the page, or whether it is due to a spurious change in the rankings. As an example of spurious rank change, for the query "network game adapter," the result `http://cgi.ebay.co.uk/Netgear-W...-QcmdZViewItem` moved into the top 10 on October 12, 2005, dropped out, and moved back in just 12 days later, causing duplicate alerts to be generated.

Example – QSR. Now let us consider a recommendation generated by our system, which received a high evaluation score in the user study described in Section 6.2.2. Consider Table 2, which shows the top 10 results for the query "rss reader," and some associated metadata. In this example, the 4th result, `http://www.google.com/reader`, has been recommended to the user. First, from her history we believe that the user has never seen this result before, at least not as a result to a search. Second, notice that this result is the only new one since the user first submitted the query (column "New") – all other results had been previously returned. Thus, we might hypothesize that this new result is not an effect of random fluctuations in rankings. Third, the rank of the result is fairly high, meaning the page is somehow good relative to other results. Finally, the absolute PageRank and relevance scores of the result (column "PR Score"), assigned by the Google search engine, are also high: although it is difficult to compare absolute scores across queries, we note that the scores for this recommendation are 3 orders of magnitude higher than the web alert example we gave earlier.

Signals. Based on analysis over examples such as the above, we identified a number of characteristics that a good recommendation should have:

1) New to the user – The user should have never seen this URL before. Note that even if the user has never *viewed* the page, she might have still *seen* a link to it as a result for the query.

2) Good Page – The web page should be a "good" result for the query (e.g., good PageRank and TFxIDF relevance).

3) Recently "promoted" – There must be something about the result that caused it to recently become a good result relative to other results from the same query. For example, perhaps the result is new or modified, or it is an old page that has become popular due to external trends, and these changes have been reflected in its rank. If possible, we prefer not to recommend a web page if it contains content similar to results the user has already seen, even if it is an otherwise good result.

Again, there is a large space of signals for the above characteristics of good recommendations. Here we list the signals we found to be useful, and the intuition behind each one. In Section 6.2.2, we will compare our intuitions with results from the user study – some of which are counter-intuitive.

*** History presence** – We store all the URLs shown to a user for her past queries. If a page appears in this history, we should not display it. In fact, because we prefer to err on the side of high precision but low recall, we will not recommend a URL from any *domain* the user has ever seen.

*** Rank** – If a result R is ranked very highly by a search engine, one might conclude that, relative to other results for the query, R is a good page. In addition, if it is also a new result, then the fact that it moved from low to high rank means that it was recently promoted.

*** Popularity and relevance (PR) score** – Results for keyword queries are assigned *relevance* scores based on the relevance of the document to the query – for example, by calculating TFxIDF, anchor text analysis, etc. In addition, major search engines utilize static scores, such as PageRank, that reflect the query-independent *popularity* of the page. The higher the absolute values of these scores, the better a result should be.

*** Above Dropoff** – If the PR scores of a few results are much higher than the scores of all remaining results, these top results might be authoritative with respect to this query. For our purposes, we say that a result R is "above the dropoff" if there is a 30% PR score dropoff between two consecutive results in the top 5, and if R is ranked above this dropoff point.

We found the above signals to be effective in our system (Section 6.2.2); however, we also tried a number of additional signals which were not effective, often to our surprise. For example, we defined the **days elapsed since query submission** signal, hypothesizing that the more days that have elapsed since the query was submitted, the more likely it is for interesting new results to exist. However, we find this signal to have no effect on recommendation quality. We also defined a **sole changed** signal, which is true for a result when it is the only new result in the top 6. We hypothesized that it would identify those new results that are *not* a product of rank fluctuation. However, we found this signal to actually be *negatively* correlated with recommendation quality.

We also defined an **all poor** signal, which is true when all top 10 results for a query have PR scores below a threshold. We hypothesized that if every result for a query has low score, then the query has no good pages to recommend. Our experiments show this signal to be effective in filtering out poor recommendations; however, support for this observation was not high. Further details for all signals can be found in [21].

Quality Score. As with `iscore`, we attempt to define a *quality score* that is correlated with the quality of the recommendation. Initially, we defined this score as follows: `qscore` $= a \cdot (\text{PR score}) + b \cdot (\text{rank})$. Although this definition is simple and intuitive, we found (Section 6.2.2) that it is in fact a *suboptimal* indicator of quality. We thus define an alternate score with superior performance: `qscore*` $= a \cdot (\text{PR score}) + b \cdot (\frac{1}{\text{rank}})$. Discussion of this counter-intuitive result will be given in Section 6.2.2. Again, the boolean signal "above dropoff" is used as a filter, but not incorporated directly into `qscore*`.

5. USER STUDY SETUP

The purpose of our study is to show that our system is effective, and to verify the intuitions behind the signals defined in previous sections. We conducted two human subject studies on users of Google's Search History service. Our first study is a "first-person study" in which history users are asked to evaluate their interest level on a number of their own past queries, as well as the quality of recommendations we made on those queries. Because users know exactly what their intentions are in terms of their own queries, and because these queries were not conducted in an experimental setting, we believe a 1st person study produces the most accurate evaluations. However, because the number of recommendations is necessarily limited, due to our current implementation of the "history presence" signal (as described below), we were not able to gather sufficient first-person data on recommendation quality signals. Thus, we conducted a second study, in which "third-person" evaluators reviewed anonymous query sessions, and assessed the quality of recommendations made on these sessions.

The survey was conducted internally within the Google engineering department. It is thus crucial to note that while our results demonstrate the promise of certain approaches and signals, they are not immediately generalizable until further studies can be conducted over a larger population.

5.1 First-Person Study Design

In our first study, each subject filled out an online survey. The survey displayed a maximum of 30 query sessions from the user's own history (fewer sessions were shown only when the user's history contained fewer than 30 sessions). For each query session, the user was shown a visual representation of the actions, like the example shown in Table 1.

For each query session, next to the visual representation of actions, we ask the first three questions shown in Table 3. Question 1 deals directly with prior fulfillment of the query, while Question 3 deals with duration. We do not explicitly ask for a user's interest level in query topic; instead this is implicit in Question 2, which directly measures the level of standing interest in the query.

For each query session, we also attempted to generate

(1) During this query session, did you find a satisfactory answer to your needs?

Yes	**Somewhat**	**No**	**Can't Remember**
52.4%	21.5%	14.9%	11.2%

(2) Assume that some time after this query session, our system discovered a new, high-quality result for the query/queries in the session. If we were to show you this quality result, how interested would you be in viewing it?

Very	**Somewhat**	**Vaguely**	**Not**
17.8%	22.5%	22.0%	37.7%

(3) How long past the time of this session would you be interested in seeing the new result?

Ongoing	**Month**	**Week**	**Minute/Now**
43.9%	13.9%	30.8%	11.4%

(4) Assume you were interested in seeing more results for the query. above How good would you rate the quality of this result?

(First-person study)

Excellent	**Good**	**Fair/Poor**
25.0%	18.8%	56.3%

(Third-person study)

Excellent	**Good**	**Fair**	**Poor**
18.9%	32.1%	33.3%	15.7 %

(5) How many queries do you currently have registered as web alerts? (not including any you've registered for Google work purposes)

0	**1**	**2**	**>=2**
73.3%	20.0%	6.7%	0%

(6) For the queries you marked as very or somewhat interesting, roughly how many have you registered for web alerts?

0	**1**	**2**	**>=2**
100%	0%	0%	0%

Table 3: Survey Questionnaire and Response Breakdown

query recommendations, based on the current results returned by Google at the time of the survey (recommendations were generated on the fly as subjects accessed the survey online). If a recommendation was found for a query session, we displayed a link to the recommended URL below the query session. For each recommendation, we asked Question 4. Finally, after the survey was conducted, the users were asked Questions 5 and 6. Out of the 18 subjects that completed the survey, 15 responded to these two follow-up questions.

Selecting Query Sessions. Because a user may have thousands of queries in her history, we had to be selective in choosing the sessions to display for the survey. We wanted a good mix of positive and negative responses in terms of standing interest level, but a large fraction of users' past queries are *not* interesting. So first, we eliminated all sessions for special-purpose queries, such as map queries, calculator queries, etc. We also eliminated any query session with a) no events, b) no clicks and only 1 or 2 refinements, and c) non-repeated navigational queries, on the assumption that users would not be interested in seeing recommendations on queries that they spent so little effort on. Simply this heuristic eliminated over 75% of the query sessions in our subject group.

From the remaining pool of query sessions, half the sessions selected for the survey consisted of the highest-ranked sessions with respect to `iscore`, defined in Section 3. The second half consisted of a random selection from the remainder of the sessions. While this selection process prevents us from calculating certain statistics – for example, the fraction of users' queries that represent standing interests – we believe it gives us a more meaningful set of data with which to evaluate signals.

Selecting Recommendations. Given that the space of possible bad web page recommendations is so much larger than the space of good ones, we attempted to only show what we believed to be good recommendations, on the assumption that bad ones would be included as well.

Our method of selecting recommendations is as follows: First, we only attempt to generate recommendations for queries for which we have the history presence signal. At this time, we only have information for this signal on a small subset of all queries, thus it greatly decreases the number of recommendations we can make. Second, we only consider results in the current top 10 results for the query (according to the Google search engine). Third, for any new result that the user has not yet seen (according to the history presence signal), we apply the remaining boolean signals described in Section 4, as well as two additional signals: (1) whether the result appeared in the top 3, and (2) whether the PR scores were above a certain threshold. We require that the result matches at least 2 boolean signals. Finally, out of this pool we select the top recommendations according to `qscore` (defined in Section 4) to be displayed to the user. (We will see later that `qscore` is in fact a suboptimal indicator of quality, though we were not aware of this at the time of the survey).

5.2 Third-Person Study Design

Because we are so selective in making recommendations, we could not gather a significant set of evaluation data from our first-person study. We therefore ran a third-person study in which five human subjects viewed *other* users' anonymized query sessions and associated recommendations, and evaluated the quality of these recommendations. These evaluators were not asked to estimate the original user's interest level in seeing the recommendation; instead they were asked to assume this interest existed. As with the first-person study, we displayed a visual representation of the entire query session, to help the subject understand the intent of the original user. We also asked each subject to view the pages that the original user viewed.

In this study, which focused on recommendation quality, we included two classes of web page recommendations. Half of the recommendations were selected as described in the first-person study. The second half consisted of the highest-ranked new result in the top 10 for a given query. That is, we no longer require that the result matches at least two of our boolean signals, and we disregard its `qscore` value.

The survey appearance was identical to that of the first-person study, except that we did not include the three questions pertaining to the query session itself.

6. RESULTS OF USER STUDY

In this section, we discuss the results of the two user studies described in Section 5. Our goal in this section is to address the following three questions: (1) Is there a need for automatic detection of standing interests? (2) Which signals, if any, are useful in indicating standing interest in a query session? (3) Which signals, if any, are useful in indicating quality of recommendations?

We remind readers that while our results demonstrate strong potential, they are not immediately generalizable due to a number of caveats: the potential bias introduced by our subject population, implementation details that are somewhat specific to the Google search engine, and the filtering of query sessions and recommendations presented to our study subjects. We plan further studies in the future to see how our results generalize across wider user populations and usage scenarios.

6.1 Usage of Web Alerts

One of the crucial differences between our QSR system and existing web alert services is the automatic identification of queries that represent standing interests. However, this feature is irrelevant if users do in fact register the majority of queries in which they have a standing interest.

To assess the level at which web alert systems are used, we asked subjects how many Google web alerts they have ever registered (Question 5), and how many web alerts they have registered on queries in the survey for which they marked as "Very Interested" or "Somewhat Interested" in seeing additional results (Question 6). Of the 18 subjects in our first-person study, 15 responded to these two questions, and the breakdown of responses is shown in Table 3. From this table, we see that *none* of the users registered any of the queries from the survey for which they were very or somewhat interested in seeing additional results! The total number of such queries was 154. In addition, 73% of the subjects have never registered any Google web alert (outside of Google work purposes), and the largest number of alerts registered by any subject was only 2.

While the bias introduced by our subject population may affect these results somewhat, we believe that the results still clearly point to the need for a system such as QSR that automatically identifies standing user interests.

In terms of *why* users do not register web alerts, the main reason (from informal feedback from subjects) is simple laziness: it is too time and thought-consuming to register an alert on every interesting query. In addition, two of the respondents who had registered at least one Google web alert commented that they did not register additional alerts because of the low quality of recommendations observed from the the first alert(s). These comments motivate the need for improved methods in generating web recommendations.

6.2 Effectiveness of Signals

In this section, we discuss the results of our study that demonstrate the effectiveness of the signals and heuristics defined in Sections 3 and 4. In our first-person study, 18 subjects evaluated 382 query sessions total. These subjects also evaluated a total of 16 recommended web pages. In our third-person study, 4 evaluators reviewed and scored a total of 159 recommended web pages over 159 anonymous query sessions (one recommendation per session). The breakdown of the results to both studies are shown in Table 3.

Summary. A summary of our results from this section are as follows:

- Standing interests are strongly indicated by a high **number of clicks** (e.g., > 8 clicks), a high **number of refinements** (e.g., > 3 refinements), and a high **history match score**. We can combine these signals into the interest score **iscore**, to produce an even stronger signal. We identify all query sessions with an **iscore** value above a threshold τ_i as standing interests with good accuracy – for example, we can achieve a precision of 69% and a recall of 28% (where recall is defined as the percentage of query sessions marked as interesting in the study)
  Secondary signals of standing interests include **repeated non-navigational** and **number of query terms**.
- Recommendation quality is strongly indicated by a high **PR score**, and surprisingly, a **low rank**. We can combine these signals into the **qscore*** signal. By selecting all recommendations with a **qscore*** value above a threshold τ_q, we can achieve precision/recall tradeoffs of, for example, 70%/88%, 83%/46% or 100%/12% (where recall is defined as the percentage of recommendations marked as good or excellent in the study).
  Secondary signals of recommendation quality include **above dropoff**.

In the remainder of this section, we discuss the experimental results and figures from which our observations are drawn. Additional results may be found in [21].

6.2.1 Identifying Standing Interests

Our ultimate goal for this portion of the QSR system is to automatically identify queries that represent standing interests. To determine standing interest we asked users how interested they would be in seeing additional, interesting results for a query session (Question 2). The breakdown of responses to this question is shown in Table 3.

In Figures 3 to 5, we show the percentage breakdown for each response for this question along the y-axis, given a value for a signal along the x-axis. For example, consider Figure 3, where the signal along the x-axis is the number of clicks. When there are 0 clicks in the session, the percentage of the sessions that users marked as "Not interested" in seeing new results was 40.5%, and the percentage in which users were "Very interested" in was 14.3%. In each of these graphs, the largest x-value x_{max} represents all data points with an x-value greater than or equal to x_{max}. For example, in Figure 3, the last data point represents all query sessions with at least 14 clicks. We cut off the x-axis in this manner due to low support on the tail end of many of these graphs.

Number of clicks and refinements. Figure 3 shows us the breakdown of interest levels for different values of the click signal. Here we find that, as we expected, a higher number of clicks correlates with a higher likelihood of a standing interest. For example, the probability of a strong interest is a factor of 4 higher at >= 14 clicks (53.6%), compared with 0 clicks (14.3%). When we look at the number of refinements as a signal in Figure 4, we see similar behavior. At 0 refinements, the user is 5 times more likely to *not* be interested than she is to be very interested. However, at >= 12 refinements, the user is twice as likely to be very interested than not. Both of these signals match intuition: the more "effort" a user has put into the query, both in terms of clicks and refinements, the more likely the user is to have a standing interest in the query.

History match. When a query closely matches a user's history (i.e., in the top 10 percentile using our history match score – see [21]), the probability that the user is very interested is 39.1%, which is over 2 times the overall probability of being very interested. Likewise, the probability that a user is *not* interested is just 4.3% – almost an order of magnitude less than the overall probability of being not interested! We conclude that while low history match scores do not necessarily imply interest (or lack thereof), high history match scores are a strong indicator of interest.

Number of terms. We also note that the number of query terms does somewhat affect interest level, but not to the same degree as our other signals. In particular, our sub-

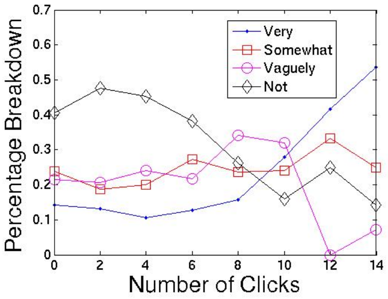


Figure 3: Number of Clicks vs. Standing Interest Level

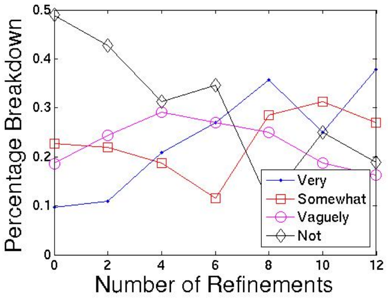


Figure 4: Number of Refinements vs. Standing Interest Level

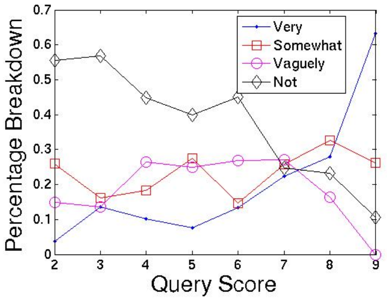


Figure 5: IScore vs. Standing Interest Level

jects were very interested in 25% of the queries with >= 6 terms, but only 6.7% of the queries with 1 term. It would appear that specific needs represented by longer queries imply higher interest levels, though as we show in [21], they also imply more ephemeral interest durations.

Repeated Non-navigational. The support for repeated non-navigational queries is quite low – only 18 queries fall into this category. However, we can observe a good indication of *prior fulfillment*. We find that users are more likely to have found a satisfactory answer (77.8%) if the query was a repeated one, than if the query was not (51.3%). Further investigation over a larger dataset is needed to confirm the quality of this signal.

Interest score. Putting the most effective signals together into a single score, in Section 3 we defined the *interest score* for a query session to be:

$$\texttt{iscore} = a \cdot \log(\#\text{ clicks} + \#\text{ refinements}) + b \cdot \log(\#\text{ repetitions}) + c \cdot (\text{history match score})$$

Figure 5 shows the breakdown of interest levels as `iscore` is varied along the x-axis. Here we see that interest level clearly increases with score. When the score is high (>= 9), the percentage of queries that represent strong interests is over 17 times higher than when the score is 0. Likewise, the probability of being *not* interested is over 5 times lower

Precision and Recall. Our goal is to develop a heuristic that can automatically identify those query sessions in which users have standing interests. For purposes of evaluation, we say that a user has a *standing interest* in a query session if the user marked that they were "Very" or "Somewhat" interested in seeing additional results for that session. We define *precision* as the percentage of query sessions returned by this heuristic that were standing interests. *Recall* is difficult to define, because we do not know how many queries represent standing interests in a user's entire history. Instead, we define recall as the percentage of all standing interests *that appeared in the survey* that were returned by this heuristic.

In our current prototype of QSR, our heuristic is to return all query sessions with an `iscore` value above a threshold τ. By varying τ, we can achieve a number of precision/recall tradeoff points – for example, 90% precision and 11% recall, 69% precision and 28% recall, or 57% precision and 55% recall. Because we are more interested in high precision than high recall (since, as discussed in Section 2, we can only generate recommendations for a limited number of queries), we would select a tradeoff closer to 69%/28%.

To better understand these numbers, we note that in our study, only 382 out of 14057 total query sessions from our subjects' histories were included in the survey. Of these 382, 154 were marked as standing interests. In addition, recall from Section 5 that a portion of the query sessions in the survey were "randomly" chosen (after passing our initial filters), without regard to `iscore` value. Of these sessions, only 28.5% were marked as standing interests. Thus, a strategy of randomly selecting query sessions after applying a few initial filters (e.g., there must be at least one action, it must not be navigational, etc.), yields a precision of just 28.5%.

6.2.2 *Determining Quality of Recommendations*

In both the 1st-person and 3rd-person studies, users evaluated the quality of a number of recommendations. Because of the low number of such evaluations in the 1st-person study, the results shown in this subsection are gathered from our 3rd-person study.

Note from Table 3 that the breakdown of evaluations across the two studies (Question 4) are not identical but reasonably close. Our goal is to recommend any result that received a "Good" or "Excellent" evaluation – we will call these the *desired* results. Using this criteria, 43.8% of the results from the first-person study were desired, as compared to 53.0% of the third-person study. We will show that our method for selecting recommendations in the 1st-person study was not ideal, possibly explaining the discrepancy between the two studies. For the purposes of this exploratory work, we will focus on the data gathered on the 3rd-person study. We hope to gather additional first-person data in future user studies.

Rank. Our first, and initially most surprising, observation is that rank is actually *inversely* correlated with recommendation quality. Figure 6 shows us the percentage of desired recommendations (i.e., with an "Excellent" or "Good" rating from the evaluator) along the y-axis, as we vary the rank along the x-axis. Note that a larger numerical value for rank means that the search engine believed that result to be of *lower quality* than other results. Here we see clearly that as rank deteriorates (i.e., grows larger in value), the percentage of high-quality recommendations increases, from 45-50% for rank above 5, to 73% for ranks 9 and 10.

After further investigation, we discovered that there is an inverse correlation between rank and PR scores. Most recommendations with good rank (e.g., 1 or 2) had low absolute PR score values, while recommendations with poor rank had high PR scores. The explanation for this is as follows: If a new result was able to move all the way to a top ranked position for a given query, then chances are that the query has many (relatively) poor results. Thus, this new result is also likely to be poor in terms of relevance or popularity, even though relative to the old results, it is good.

This observation also implies that our `qscore` value, used to select results to recommend for our 1st-person study, is

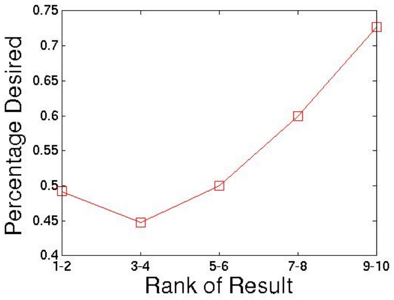


Figure 6: Rank vs. Percentage of Desired Recommendations

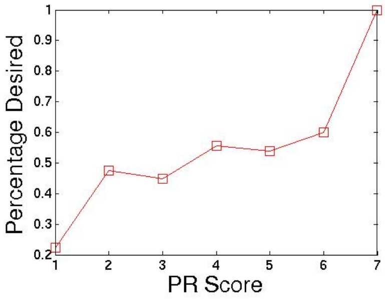


Figure 7: PR Score vs. Percentage of Desired Recommendations

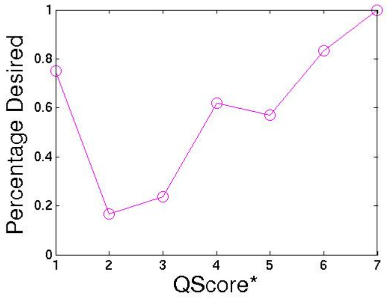


Figure 8: `Qscore*` vs. Percentage of Desired Recommendations

in fact a suboptimal indicator of recommendation quality. This may partially explain why the quality levels indicated in the 1st-person study are lower than those in the 3rd-person study.

Popularity and Relevance Score. The explanation for rank's inverse correlation with quality implies that PR scores should be correlated with quality. In Figure 7 we see that this is the case: only 22% of the recommendations with the lowest score of 1 were considered to have high quality, compared to 100% of the recommendations with a score of 7 or more! Despite this promising evidence, however, we find that for the bulk of the recommendations with scores 2 to 6, the probability of being high quality is ambiguous – just 50%. We would ideally find a signal that is better at differentiating between results.

QScore*. Based on our previous observations, we tried a new signal, `qscore*`, which we define as follows: `qscore*` $= a \cdot \text{PR score} + b \cdot \frac{1}{\text{rank}}$. Any result with a non-positive value for this score is eliminated. The idea behind this score is to emphasize the low quality that occurs when a new result moves to a top rank. In Figure 8 we see the quality breakdown as a function of this new score. From this figure we make two observations: (1) `qscore*` is good at differentiating quality recommendations (the curve has a steep slope), and (2) a strange spike occurs at `qscore*` $= 1$. We would like to conduct further studies to confirm our first observation and to explain the second. Initial investigation suggests that the spike occurs because it accounts for all top-ranked results with *medium* PR scores. In particular, 86% of all recommendations in this data point have a rank of 1. Top-ranked results with low PR scores – those results that cause the inverse correlation between rank and quality – have non-positive values of `qscore*`, and are thus filtered from consideration.

Precision and Recall. We say that a web page recommendation is "desired" if it received an "Excellent" or "Good" rating in our survey. Our goal is to identify all desired recommendations. Our heuristic is to assign each potential web page recommendation a score (such as `qscore*`), and select all pages above a threshold τ. For a given scoring function and threshold, we define *precision* as the percentage of desired web pages out of all pages selected by the heuristic. *Recall* is defined to be the percentage of selected desired pages out of all desired pages considered in our survey dataset.

By varying the threshold τ, we can achieve different precision/recall tradeoffs for a given scoring function. Figure 9 shows the precision-recall tradeoff curves for three different quality scoring functions: (1) `qscore`, our original scoring function, (2) `PR score`, the scores assigned by the search engine that should reflect relevance and popularity, and (3) `qscore*`, our new scoring function. For `qscore*`, we recommend all pages above the threshold τ, *and* all pages with a score of 1 (to accommodate the spike seen in Figure 8). From Figure 9, we see that if we desire a precision above 85%, then we should use `PR score`. In all other cases, `qscore*` provides the best precision/recall tradeoff, often achieving over twice the recall for the same precision when compared to `PR score`. For example, with `qscore*` we can achieve a precision/recall tradeoff of 70%/88%, whereas with `PR score`, the closest comparison is a tradeoff of 68%/33%. Function `qscore` is completely subsumed by the other two functions.

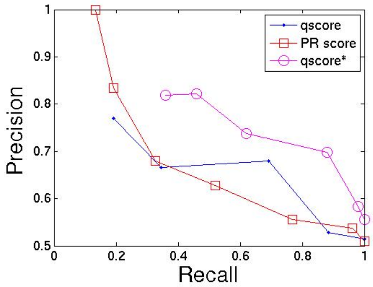


Figure 9: Precision/Recall Tradeoff for Quality Scores

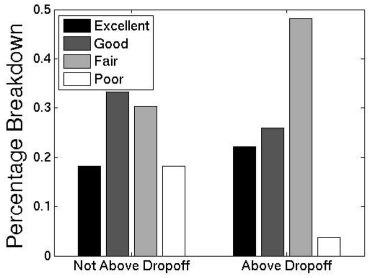


Figure 10: Above Dropoff vs. Recommendation Quality

Again, we emphasize that these results are specific to Google's search engine and are not immediately generalizable to all situations. However, we believe they provide insight into the higher-level principles that govern the tradeoffs seen here.

Above Dropoff. This boolean signal is also a reasonable indicator of recommendation quality. In Figure 10, we see the breakdown of recommendation quality when recommendations passed the “above dropoff” signal (on the right of the figure), and when they do not (on the left of the figure). From this figure, we see that this signal is very good and eliminating “Poor” recommendations: only 3.7% of all recommendations above the dropoff were given a “Poor” rating, compared to 18.2% of all recommendations not above the dropoff. The downside of this signal is that it results in a large percentage of “Fair” recommendations, as opposed to “Good” ones.

7. RELATED WORK

Many existing systems make recommendations based on past or current user behavior – for example, e-commerce sites such as Amazon.com [1] recommend items for users to purchase based on their past purchases, and the behavior of other users with similar history. A large body of work exists on recommendation techniques and systems, most notably collaborative filtering and content-based techniques (e.g., [3, 5, 8, 13, 16]). Many similar techniques developed in data-mining, such as association rules, clustering, co-citation analysis, etc., are also directly applicable to recommendations. Finally, a number of papers have explored personalization of web search based on user history (e.g., [9, 11, 18, 19]). Our approach differs from existing ones in two basic ways. First, our technique of identifying quality URLs does not rely on traditional collaborative filtering or data-mining techniques. We note, however, that these techniques can be used to complement our approach – for example, we can be more likely to recommend a URL if it is viewed often by other users with similar interests. Second, the QSR system will only recommend a URL if it addresses a specific, unfulfilled need from the user's past. In contrast, existing systems tend to simply recommend items that are like ones the user has already seen – an approach that works well in domains such as e-commerce, but that is not the aim of our system.

The idea of explicitly registering standing queries also exists; for example, Google's Web Alerts [7] allows users to specify standing web queries, and will email the user when a new result appears. Along the same vein, a large body of recent research has focused on continuous queries over data streams (e.g., [2, 4, 12, 14]). To the best of the authors' knowledge, however, our work is the first on automatically detecting queries representing specific standing interests, based on users' search history, for the purposes of making web page recommendations. Ours is also the first to provide an in-depth study of selecting new web pages for recommendations.

Related to the subproblem of automatically identifying standing interests, a recent body of research has focused on automatically identifying a user's goal when searching. For example, reference [10] identifies the user's high-level goal for a query (e.g., navigational vs. informational) based on aggregate behavior across many users who submit the same query, and assumes that all users have the same intent for a given query string. Our work is related in that we also try to identify a user's intent; however, we try to predict what the specific user is thinking based on her specific actions for a specific query – in other words, it is much more focused and personalized.

8. CONCLUSION AND FUTURE WORK

Our user studies show that a huge gap exists between users' standing interests and needs, and existing technology intended to address them (e.g., web alerts). In this paper, we present QSR, a new system that retroactively answers search queries representing standing interests. The QSR system addresses two important subproblems with applications beyond the system itself: (1) automatic identification of queries that represent standing interests and unfulfilled needs, and (2) identification of new interesting results. We presented algorithms to address both subproblems, and conducted user studies to evaluate these algorithms. Results show that we can achieve high accuracy in automatically identifying queries that represent standing interests, as well as in identifying relevant recommendations for these interests. While we believe many of our techniques will continue to be effective across a general population, it will be interesting to see how they perform across a wider set of users.

9. REFERENCES

[1] Amazon website. http://www.amazon.com.

[2] S. Babu and J. Widom. Continuous queries over data streams. In *SIGMOD Record*, September 2001.

[3] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence*, 1998.

[4] J. Chen, D. DeWitt, F. Tian, and Y. Wang. Niagaracq: A scalable continuous query system for internet databases. In *Proc. of SIGMOD*, 2000.

[5] D. Goldberg, D. Nichols, B. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. In *Communications of the ACM*, December 1992.

[6] Google website. http://www.google.com.

[7] Google Web Alerts. http://www.google.com/alerts.

[8] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR*, August 1999.

[9] G. Jeh and J. Widom. Scaling personalized web search. In *Proc. of WWW 2003*, May 2003.

[10] U. Lee, Z. Liu, and J. Cho. Automatic identification of user goals in web search. In *Proc. of WWW 2005*, May 2005.

[11] F. Liu, C. Yu, and W. Meng. Personalized web search by mapping user queries to categories. In *Proc. of the Conference on Information and Knowledge Management*, November 2002.

[12] S. Madden, M. Shah, J. Hellerstein, and J. Raman. Continuously adaptive continuous queries over streams. In *Proc. of SIGMOD*, 2002.

[13] P. Melville, R. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommenadtions. In *Proc. of the Conference on Artificial Intelligence*, July 2002.

[14] J. H. Hwanga nd M. Balazinska, A. Rasin, U. Cetintemel, M. Stonebraker, and S. Zdonik. High availability algorithms for distributed stream processing. In *Proc. of the 21st International Conference on Data Engineering*, April 2005.

[15] J. Pitkow, H. Schutze, T. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Edar, and T. Breuel. Personalized search. In *Communications of the ACM, 45(9):50-55*, 2002.

[16] A. Popescul, L. Ungar, D. Pennock, and S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence*, 2001.

[17] D. Rose and D. Levinson. Understanding user goals in web search. In *World Wide Web Conference (WWW)*, 2004.

[18] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proc. of WWW*, 2004.

[19] J. Sun, H. Zeng, H. Liu, Y. Lu, and Z. Chen. Cubesvd: A novel approach to personalized web search. In *Proc. of WWW 2005*, 2005 May.

[20] Yahoo website. http://www.yahoo.com.

[21] B. Yang and G. Jeh. Retroactive answering of search queries. Technical report, 2006. Extended version, available upon request.