Web Search Clickstreams

Nils Kammenhuber
Technische Universität München, Germany
hirvi@net.in.tum.de

Julia Luxenburger
Max-Planck Institute of Informatics, Germany
julialux@mpi-inf.mpg.de

Anja Feldmann
Deutsche Telekom Laboratories, Germany
anja.feldmann@telekom.de

Gerhard Weikum
Max-Planck Institute of Informatics, Germany
weikum@mpi-inf.mpg.de

ABSTRACT

Search engines are a vital part of the Web and thus the Internet infrastructure. Therefore understanding the behavior of users searching the Web gives insights into trends, and enables enhancements of future search capabilities. Possible data sources for studying Web search behavior are either server- or client-side logs. Unfortunately, current server-side logs are hard to obtain as they are considered proprietary by the search engine operators. Therefore we in this paper present a methodology for extracting client-side logs from the traffic exchanged between a large user group and the Internet. The added benefit of our methodology is that we do not only extract the search terms, the query sequences, and search results of each individual user but also the full *clickstream*, i.e., the result pages users view and the subsequently visited hyperlinked pages. We propose a finite-state Markov model that captures the user web searching and browsing behavior and allows us to deduce users' prevalent search patterns. To our knowledge, this is the first such detailed client-side analysis of clickstreams.

Categories and Subject Descriptors

I.6 [**Computing Methodologies**]: Simulation and Modeling; C.2.m [**Computer Systems Organization**]: Computer-Communication Networks—*Miscellaneous*; H.1.2 [**Information Systems**]: Models and Principles—*User/Machine Systems*

General Terms

Measurement, Human Factors

Keywords

Web search, clickstream, HTTP traces, Markov model

1. INTRODUCTION

Interactions with search engines make up a tremendous part of users' Web activities. Indeed, search engines are an active research area in itself. Yet, in order to enhance their capabilities, a good understanding of current user behavior, especially the characteristics of their clickstreams[1] is needed.

Besides gaining new insights into user search patterns, query clickstreams can serve as a means for Web search enhancement. In the past, query logs [10] have been used to extend state-of-the-art link analysis on the web graph or to perform query clustering [6] for query expansion. The implicit feedback inferred from such logs can be used as input to machine learning approaches [12] or used in the estimation process of language-model based information retrieval [13].

Unfortunately, the currently available data sets about clickstreams are rather limited. In principle there are two ways of gathering such data, either on the server or on the client side. As server-side data is considered proprietary, current analyses are limited to only a few search-engine-specific data sets, including one gathered in 1998 from Altavista [14], one from the Excite search engine [17] in 1997, and one from Vivisimo [16] in 2004. Furthermore, none of these data sets include the full clickstream which consists of all user accesses to Web pages related to a search query. The client-side data gathering has focused on asking volunteers to surf the net using additional browser plugins, e.g., [18], or enhancing HTTP proxies with extended logging functionality, e.g., [2]. Yet, not all sites currently use proxies or are willing to modify them.

We, in this paper, extract client-side logs from packet level traces of the traffic exchanged between a large research network and the Internet. From the packet level traces we extract all HTTP requests and responses as well as the bodies of all responses from search engines. From this data we reconstruct each of the users' query sessions. This includes that we determine for each search query the position of the search results the user clicked upon (if any). Furthermore we recursively analyze how many links (if any) the user followed from the search result. A prototype system analyzes the traffic at the border of the Munich Scientific Network focusing on the Google search engine [1]. Utilizing this data, we present a characterization of the query sessions as well as a finite-state Markov model that relates the Web search clickstreams to the Web hyperlink structure.

The remainder of this paper is organized as follows: Section 2 reviews related work. Terminology used throughout the paper is defined in Section 3. Our methodology for extracting Web search clickstreams is discussed in Section 4. Section 5 summarizes the characteristics of the query sessions while Section 6 presents the Markov model. We conclude the paper with an outlook in Sec-

IMC'06, October 25–27, 2006, Rio de Janeiro, Brazil.

[1] A clickstream contains all Web page accesses of one user.

tion 8.

2. RELATED STUDIES

Studies of Web search behavior can be categorized along different axes according to whether the data is gathered from search engine logs or on the client side, the time period they cover, as well as the measures applied and research questions pursued. Jansen and Pooch [7] survey and compare studies on traditional information retrieval systems, online public access catalogs, and Web searching up to the year 2000. They find that there is a lack of clarity in the descriptions and that the use of different terminologies by the various studies make the results hard to compare. They propose a framework for such analysis which we use in this study.

More recently, a number of researchers, e.g., [9] have focused on categorizing queries according to user search goals in order to improve search performance. Lee et al. [9] rely on packet level traces. Yet, as the focus of their paper is on automatic identification of user goals in Web search, they do not systematically establish a relationship between the position of the search results and the gathered clickstream, nor do they consider follow-up clicks. Another line of work, e.g., [4] aims at a topical query classification using data from a major commercial search service.

Chau et al. [5] examine which documents of the result pages are viewed by the user; they, however, do not consider which hyperlinks the user follows beyond these. In addition, their study is limited to Web site search. Spink et al. [15] use a Markov model for query reformulation patterns of the Excite search engine. Their model, however, cannot include the user behavior beyond the search engine interface: it neglects which documents reachable via the result pages are visited by the user during a query session. None of these studies take the whole Web query clickstream into account.

3. TERMINOLOGY

To simplify the discussion, we briefly summarize some of the terms that we use in the remainder of the paper. The definitions are in part taken from Jansen et al. [7] and from Spink et al. [17].

Term: any unbroken string of alphanumeric characters entered by a user, e.g., words, abbreviations, numbers, URLs, and logical operators.

Query: a set of one or more search terms as typed by the user (may include advanced search operators). As result of a query the search engine returns a **query result page**.

Query session: a time-contiguous sequence of queries issued by the same user.[2]

Unique query: is unique within a query session.

Repeat query: re-occurrence of the same query in the same session, e.g., when a user retrieves several result pages for one query. We furthermore distinguish between **next result page queries** and **real repeat queries**. The latter indicate multiple requests for the same query result page.

Result links: links contained in the query result page.

Result position: the absolute position of a result link in the query result page.

[2]An alternative definition for a query session is: a sequence of queries by one user which satisfy a single information need. Unfortunately identifying such sessions is only possible by finding semantic demarcations, e.g., by relying on query similarity. But recent work [16] indicates that such demarcations may not exist as a user may work simultaneously on several information needs or may have rapidly switching information needs.

Clicks: `text/html` HTTP requests that are the result of a user clicking on a hyperlink.

Result clicks: result links on which a user clicks.

Clickstream: all `text/html` requests related to a query session.

4. SEARCH CLICKSTREAMS

In order to monitor the **Web search clickstreams** of a set of users we rely on capturing client side logs. More specifically, we suggest to use packet level traces as our main data source. From these traces we extract:

for the search engine under study all HTTP requests and responses including their bodies and the HTML links they contain.
for all Web servers all HTTP request and response headers.

Neither standard browsers nor Web proxies provide us with this kind of data. Thus one would have to either instrument all Web browsers or install a modified Web proxy into the data path [2, 8].

While one could use any tool that can reconstruct HTTP level detail from packet level traces (see, e. g., [8]) we utilize the HTTP analyzer of Bro [11], a network intrusion detection system. A self-written "policy" file for Bro extracts all the data described above from both standard HTTP/1.0 as well as persistent or pipelined HTTP connections. For extracting HTML links from the bodies retrieved from the Web search engine, we use the Perl module `HTML::Parser`.

4.1 Search Queries

To determine which requests are search queries, one has to consider the specifics of the search engine. We, in this paper, focus on queries to Google, but the same principle methodology can be used to extract clickstreams for other search engines. For us a HTTP request is a Google query if the following conditions hold:

- The request is to one of our locally-seen Google server in the subnets `66.249.64.0/18`, `64.233.160.0/19`, `216.239.32.0/19`, `72.14.192.0/18`, or `66.102.0.0/20`.
- The `Host` header contains the string "`.google.`".
- The URI starts with `/search?`, contains a non-empty CGI parameter `q=...`, and does *not* contain the string `client=navclient-auto`.
- The `Content-Type` field of the response contains the string `text/html`.
- The `User-Agent` field does not indicate an automated query, i. e., it does not contain any of these strings: *bot, agent, crawler, wget, lwp, soap, perl, python, spider*.

4.2 Sessions (User Web search clickstreams)

In order to statistically evaluate the search behavior, we need to group the search-related actions of individual users. Thus we identify search requests, and we determine individual *sessions* that contain all search requests and all other `text/html` requests that are directly or indirectly reachable via the result pages of the query requests (*search-induced clicks*).

The grouping of requests into sessions is shown in Fig. 1 and works as follows: For each arriving request, we determine whether it is a search request, using the criteria from Section 4.1. If this is the case, we utilize the fact that Google embeds a cookie called `PREF` in the search request stream. The cookie contains a portion labeled `ID`, which Google apparently uses for tracking individual

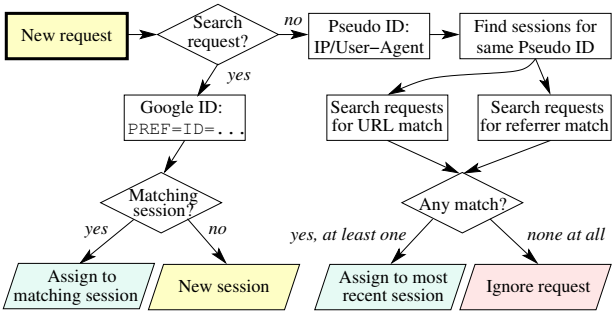

Figure 1: Determining the session for a new request.

users. We search in our pool of currently active sessions for a session that matches the given `ID`, and add the request to the session. If no session matches, we instantiate a new session.

If the current request is not a search request, we determine client IP and HTTP `User-Agent` as a *pseudo-ID* (note that this concept is orthogonal to the Google cookie `ID`s). In all sessions matching this *pseudo-ID*, we look for requests that have either the same URL as the current request, or whose URL matches the HTTP referrer of the current request. We assign the current request to the session with the most recent matching request. If no match can be found, we ignore the current request.

Sessions that have not seen a request for more than five minutes are considered to have ended. They are removed from the pool of active session and will not be re-activated.

4.3 Query terms

Given that we can now group queries into sessions we also want to examine how the queries within each session differ. Therefore we consider the *query terms* within the query based on an understanding of the specifics of the query language provided by the search engine. Google's basic search [1] requires *all* search terms, except stop words[3], to be present in the results, i. e., all terms are implicitly combined by "and". Capitalization plays no role, as all typed letters are automatically converted to lower case. In contrast, term order is decisive. Accordingly, we normalize the queries to lower case but retain the search term order. Negative terms, i.e., terms that should not occur, are preceded by a minus. Phrases are surrounded by quotation marks and are treated as a single term. A plus sign in front of a term tells Google that this is not a stop word, which would be removed otherwise. The tilde sign before a term tells Google to include synonyms and is therefore kept. The `site:` operator restricts the search space to the specified domain. While Google offers several additional advanced features we, in this study, restrict ourself to the ones discussed above, the most common ones.

5. SEARCH CHARACTERISTICS

We now describe the dataset that we use for our analyses of user Web search behavior. The description is followed by an analysis of the user query behavior, in order to reveal some characteristics of the user population that we examine.

5.1 Dataset

For our analysis of the clickstreams we use data collected from the *Münchner Wissenschaftsnetz* (*MWN*; Munich Scientific Network). The MWN provides a 10 Gbit/s singly-homed Internet connection to roughly 50,000 hosts at two major universities along with additional institutes; link utilization is typically only around 200–500 Mbit/s in each direction. Our monitoring machine is connected to the monitoring port of the border switch. Since MWN as a whole imposes too much load on our tracer machine running the Bro HTTP analyzer, we restrict our data gathering to Ludwig-Maximilians-Universität, the larger one of the two universities in Munich (about 44,000 students).

Bro's memory consumption increases slowly over time, as it accumulates state from, e. g., dangling TCP connections, etc. Thus we start a new Bro process every 45 min and let it run for 50 min. The resulting 5 min overlap ensures that HTTP requests that stretch over the 45-minute boundary are not lost, apart from a few long-lasting persistent connections. Duplicate records are removed.

We consider all requests issued from Thursday, August 17th, 17:07 MET until Friday, September 1st, 21:00 MET, excluding a monitor downtime period of about 18 hours. The median packet loss as reported by Bro and `libpcap` during each 50 min interval is 0.0 % (average: 0.15 %; maximum: 7 %). The total number of HTTP requests on TCP port 80 is 125,104,884; the number of transferred HTML objects is 28,026,595, of which only 19,601,616 have HTTP status code 200. Note that these numbers also include "abuses" of the HTTP protocol by non-Web applications.

During this time period, we identified a total of 545,455 Google search queries. There are 275 empty queries where the user clicked the "Search" button before entering a query. Out of the remaining Google queries 414,184 are unique queries, and 130,996 are repeat queries. The repeat queries consist of 105,683 real repeat queries and 25,313 next result page queries. Manual inspection shows that most queries relate to local specifics of the area of Munich. Yet, as expected, the queries also reflect the academic environment. We note that the data is skewed towards academic users which may exhibit a different behavior than the general population.

5.2 Query session characteristics

To understand the characteristics of the sessions within the reconstructed clickstream, we start by presenting statistics similar to those of previous studies [5, 7, 17].

Query length: The maximum number of terms per query we encountered is 199; one user copied a whole text snippet into the query box. The median query length is 1 while the mean is 1.67. This shows an even stronger trend towards very short queries than observed in the course of previous studies, e.g., [14]. 64 % of the queries consist of a single query term, 83 % contain less than three terms, and 99 % consist of less than six terms.

Use of search operators: Of the 414,184 unique queries 11,977, i.e., 3% use phrase search. 832 queries enforce the occurrence of a search term via the plus operator, and 315 queries use the minus operator to exclude search terms. However, similarity search utilizing the tilde operator does not occur at all. Finally, domain queries occur 492 times.

Terms: We identify 249,445 distinct terms in our data, however 124,095 of these are of the form "info:<*url*>" to request further information on specific URLs from Google. We omit these terms in the following statistics (note that this might be an explanation for the large number of one-keyword queries found). The most frequent terms are listed in Table 5.2 in descending order of their frequency. This statistic clearly reveals a bias in our dataset towards local themes (high frequency of terms relating to Munich and Germany), as well as academic subjects. For example, the term "lmu" is short for "Ludwig-Maximilians-Universität", which is the university whose Web traffic is analyzed in this study. Also note the use of stop words ("in", "der") and the set of presumably nav-

[3] very frequent words carrying little information like *a, the, ...*

Term	Freq.	Term	Freq.	Term	Freq.
münchen	9,815	lmu	1,009	die	710
in	2,830	für	969	windows	703
the	1,875	2006	895	wetter	676
of	1,809	online	852	a	662
+	1,724	von	832	lyrics	628
der	1,532	de	825	java	623
und	1,405	to	808	uni	599
download	1,373	bayern	799	berlin	595
muenchen	1,370	linux	783	free	591
and	1,020	wikipedia	761	hotel	582

Table 1: 30 most frequent search terms

igational queries, e. g., "wikipedia". In addition, queries relating to recreational activities are quite frequent such as "hotel", "wetter" which is German for "weather", as well as searches related to companies and products (e.g., both "siemens" and "xp" occur with frequency 224). Yet, in spite of the academic environment, we also find queries on pornographic material, such as "sex" (387) and "porn" (116). Interestingly, the prominence of the term "+" possibly indicates that users have trouble with the correct usage of the "+" operator, and write "+␣the" instead of "+the" to prevent "the" from being neglected as a stopword.

Query sessions: We identify 153,719 query sessions. The median number of queries per session is 2, the mean number is 3.5. There are sessions with more than 100 queries but more than 46 % / 20 % only contain one / two queries. Similarly, the median number of *unique* queries is 1 and the mean number decreases to 2.7. Accordingly, the median number of repeat and real repeat queries is 0 while the mean is 0.85 and 0.69, respectively.

Query refinements: To describe the relationship between two consecutive queries in the same session, we distinguish between the following kinds of query modifications: *repeat* (the same query again), *disjoint* (no overlap in the query terms), *add* (the follow-up query is a superset of the previous one), *delete* (the follow-up query is a subset of the previous query), and *replace* (the follow-up query and the previous one overlap but are not strict subsets). We find that there are 20 % (76,541) repeat, 68 % (265,715) disjoint, 4.75 % (19,661) add, 2.5 % (9,521) delete, and 4.75 % (20,023) replace modifications. This rather coarse-grained categorization leaves space for further investigation, e.g., to examine how often terms are changed into phrases, or how often users re-order terms.

Result pages viewed: The maximum number of viewed result pages is 32. The mean number is 1.06 and the median is 1. If we consider the unique queries, 96.5 % look at only a single result page. Less than 0.1 % consider four or more distinct result pages.

6. A STATE MODEL FOR WEB SEARCH

To better understand the behavior of users that search the Web, we model Web user behavior during a search session as a Markov model. The Markov model relates the hyperlinks between the Web documents (thereby capturing the relationships between the requested documents) with the clickstream (the sequence in which a user requests the documents) and the properties of the documents (position in search result, HTTP status code).

The goal of the model is to help answer numerous questions regarding a user's navigational behavior during Web search, e. g.: Which link result position is likely to contain the answer a user is looking for? Is a user likely to explore the Web site of a top-ranked result click further than a subsequent one? Do protections against "deep links" from search engines affect user search behavior?

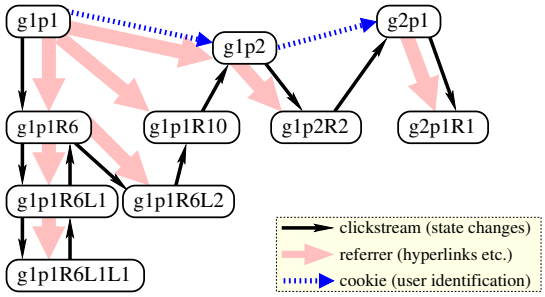

Figure 2: Logical relations (referrer, cookies) and actual click sequence as conducted by the user in a sample search session.

6.1 States and transitions

Each *state* in our Markov model includes important aspects of the users' navigational behavior. It retains the following information:

index of search query in session: g_x **index of result page:** p_y **position of result click:** R_k (if known)

Furthermore, as a user may visit additional pages between result clicks, we capture the tree structure of such requests by keeping an index for the tree depth L_i and the number of sons for each level of the tree L_z. Moreover, a state may have additional attributes for capturing whether the page was reached via a different HTTP status code than "`200 OK`".

Each state captures the logical relationship of the requested page to the query that directly or indirectly made the user access this page. As the user clicks on pages he maneuvers through the state space, and each click on a hyperlink corresponds to a state change. (Note, however, that requests served directly out of the client cache can only be inferred if they do not result in an `If-Modified-Since` request.) In effect, when viewing the clickstream as a set of events over time, the current state represents the user's (presumed) navigational position in the graph of hyperlinked documents accessed during the search session.

Let us consider an example of a search session where the user searches for information on the soccer world championship in 2006. The states in our Markov model are given in parentheses; the entire search is depicted in Fig. 2. At first, the user might submit the query "soccer" to the Google search engine (g_1p_1), and explore the sixth result link on page 1 ($g_1p_1R_6$), e. g., "soccer international root page", in a new browser window. On this Web site he explores, e. g., the link for "German version" ($g_1p_1R_6L_1$), where he follows yet another link ($g_1p_1R_6L_1L_1$). He finds that this link does not contain what he was looking for, thus presses the back button twice, which results in two `If-modified-since` requests ($g_1p_1R_6L_1$, $g_1p_1R_6$), and clicks on the "English version" page instead ($g_1p_1R_6L_2$). Still unsatisfied, he goes back to the initial search result list (g_1p_1) and explores the tenth link: "soccer-sites.com" ($g_1p_1R_{10}$). As this site does not contain the desired information either, he takes a look at the next set of results from Google (g_1p_2), where he clicks on the second result link ($g_1p_2R_2$) "ussoccer.com". This is still not a site about the FIFA World Cup in Germany, so he refines his original search by typing "Soccer world cup" in yet another browser window (g_2p_1). In this case, the first result ($g_2p_1R_1$) points to "The official site for the 2006 FIFA World Cup Germany", which the user clicks on. The thick lines in the background of Fig. 2 show the relationships (i. e., hyperlinks) between the Web pages, whereas the thin black arrows depict the actual user clickstream.

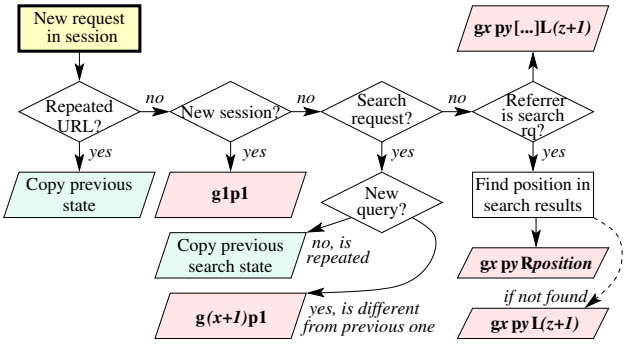

Figure 3: Determining the Markov state for a new request.

We distinguish three types of states:
Query result pages: have the form $g_x p_y$ indicating that this is the y-th result page of the x-th query in the session.
Result clicks: have the form $g_x p_y R_\gamma$, where $g_x p_y$ identifies the query and γ is the result position.
Other clicks: have the form πL_z where π is the state of hyperlinking document Π that originates the click, and z is the index of the clicked document in the vector of clicked hyperlinks originating from Π, ordered in time.

6.2 From HTTP logs to Markov states

For the construction of the Markov model, we focus on `text/html` objects, since each Web page has an HTML document as its skeleton. A brief analysis of HTTP `Content-Types` reveals that the number of transferred objects in other potentially hyperlink-capable formats such as PDF or XML is insignificant.

To capture the Web search clickstreams by means of a Markov model, we apply the following logic: For each new request ν, we first locate the session that it belongs to, using the method described in Section 4. Then we determine the state to be assigned to the request using the mechanism outlined in Fig. 3:

First we examine whether ν requested the same URL as a previous request ρ in the same session. If so, we simply assign ν the same state as ρ. Otherwise, we check whether ν is a search request. If so, we either increment the search index number from $g_x \ldots$ to $g_{(x+1)} \ldots$ (in the case of a new query), or we retain the old search number $g_x \ldots$ if the user repeats her query, e. g., she might request the next Google result page (which requires an adjustment of the page number from $g_x p_y$ to $g_x p_{(y+1)}$), or she might have entered the same search terms again. If the request ν is not a search request, we examine the request ρ that corresponds to the URL where the referrer of ν points to—note that this task is described in Section 4.2. Assume that ρ has state $g_x p_y[\ldots]$, and that previous requests already have been assigned the states $g_x p_y[\ldots]L_1$ to $g_x p_y[\ldots]L_z$. Then we assign ν the state $g_x p_y[\ldots]L_{z+1}$. An exception occurs if ρ is a search request: In this case, we determine the position of the URL for ν in the HTML code pertaining to ρ and thus determine its search rank. Here, the state we assign to ν is $g_x p_y R_{position}$, i. e., it depends on the position of ν in the list, but *not* on the number of child states of ρ.

Note that the user clicking on a single hyperlink may trigger the download of multiple `text/html` documents. For example, the URL that a hyperlink points to may result in a "`302 Found`" redirect (having `text/html` as `Content-Type`), which points to an HTML document consisting of multiple frames, each contained in yet another individual `text/html` object. Thus we assign a request the same state as a previous request from the same session, if they are less than one second apart from each other and are linked via URL / referrer. To keep Fig. 3 simple, this mechanism is omitted in the picture. In the case that we cannot capture the beginning of a session, a "new" session may start with, e. g., $g_1 p_3$ instead of the normal case $g_1 p_1$ shown in Fig. 3. This is due to the fact that we always calculate the page number ($g_\ldots p_y$) from the search request's URL. We find only a small number of such exceptions in our data.

When identifying state transitions as indicated by the temporal evolution of the search clickstream, we find that the time sequence of the clickstream (thin black arrows) is likely to differ from the hyperlink graph as highlighted in Fig. 2. For example, by keeping the query result page in a separate window, the user can call $g_1 p_1 R_{10}$ without having to re-enter his query; thus a re-request of $g_1 p_1$ is not necessary. Note that the same page and therefore the same state can be reached multiple times, e. g., when the user presses the "Back" button after retrieving page $g_1 p_1 R_6 L_1 L_1$.

7. MODEL-BASED ANALYSIS

Our state model does not only investigate the search queries, but also takes into account all subsequent clicks that are direct or indirect consequences of the users clicking on search results (followup clicks). In the following, we demonstrate the broad applicability of our model by highlighting some key findings.

Using the data (see Section 5.1) with our model, we assign a state to 1,488,246 `text/html` requests with HTTP status code 200 and identify 1,336,418 "clicks" (search states). Thus **the share of search operations and their followup clicks amounts to least 6.8 percent** of all transferred HTML documents.

If we analyze the distribution of the number of followup clicks for each individual search request (i. e., the "child states" for each $g_x p_y$), we see a mostly linearly falling slope on a CCDF plot (Fig. 4, circles). In this respect, **search-triggered Web sessions thus do not seem to differ from Web sessions in general** [3]. The same holds for the distribution of total the number of clicks per session (plot not shown).

Similar behavior occurs in the distribution of the number of clicks between a document and the original search request, which we call *click distance*. In addition to this, Fig. 4 reveals that the users behave slightly different during working hours (triangles) than during recreational times (cross-hairs): **During their spare time, users are more likely to engage in "serendipity clicks" leading them away from the page they may have been originally looking for.**

Next, we compare the total number of clicks per session (circles) vs. the distribution of click distances (cross-hairs) in the corresponding time intervals. We observe that both curves overlap almost exactly for almost the entire range, except at the end. This suggests that **most users follow a rather linear approach during searching and browsing; i. e., they normally do not click on the "back" button and follow another link on the previous page,** etc. Only long sessions with many clicks seem to differ significantly in this respect (lower right of plot).

Users are much more likely to re-formulate queries than to view the second result page. By comparing states $g_i p_1$ with $g_i p_2$ and $g_{i+1} p_1$, we observe that a user is about 37 times more likely to enter a new query than to look at the next result page.

If we look at the lozenges in Fig. 5 (bottom), we see that **most users request the first search result**—in fact, more than 60 % of the clicks on any ranked search result (i. e., excluding advertisements, search settings etc.) go to position 1. Note, however, that we witnessed automatic pre-fetching of the first search result in a number of cases.

When we consider the number of follow-up clicks to a search

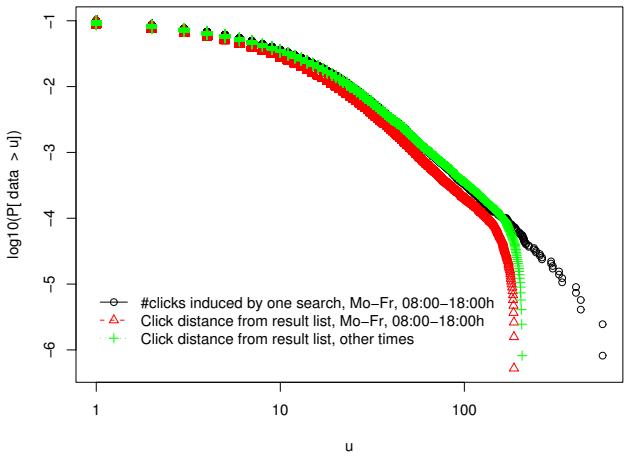

Figure 4: CCDF for number of clicks triggered by one search operation, and CCDFs for click distance between search page and visited document.

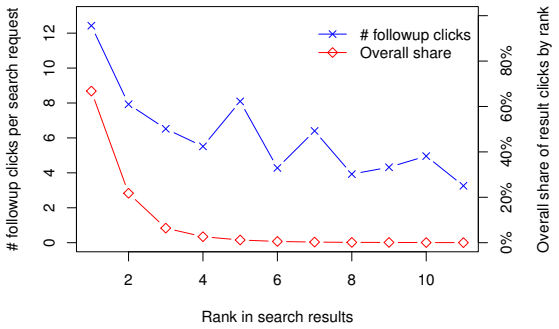

Figure 5: Rank in search result vs. willingness of the user to continue browsing from the result page.

result as a measure for the result quality, we see that top-ranked search results indeed seem to be of higher quality than lower-ranked ones (Fig. 5, top line with crosses). Yet, the presumed result quality difference between high ranks and low ranks is significantly lower than the difference in popularity (lozenges). This suggests that the **page summaries in the result list probably are read by most users** before they click.

Let us finally analyze the impact of HTTP redirects that lead the user away from a search result: If a user enters any state $g_x p_y \{R, L\}_z$ via an HTTP redirect (not issued by the search engine), the average number of clicks that start from this document is only 0.12, as compared to the normal 6.5. This means that a user who clicks on a search result, but is redirected to a different page than the desired one, normally does not spend any time on that page. We conclude that **operators protecting their Website against "deep linking" from search engines repel many potential customers.**

8. SUMMARY AND OUTLOOK

In this paper we analyze Web search clickstreams. Our data is gathered by extracting the HTTP headers from packet-level traffic, as well as the bodies of Web search result pages. We correlate both data sets to extract sequences of subsequently posed queries, and relate each query to its clicked result pages and follow-up clicks. In the future we intend to perform timing-based analyses by considering the time a user spends in each state. Furthermore, we are in the process of gathering more data across an longer time period and / or a different user population to solidify our analysis results.

Based on the data gathered so far we find that most queries consist of only one keyword and make little use of search operators, such as the plus, minus or tilde sign. Moreover, users issue on average four search queries per session, of which most consecutive ones are distinct. Relying on our Markov model that captures the logical relationships of the accessed Web pages, as well as the users' navigational behavior, we gain additional insights on users' Web search behavior. Users are much more likely to re-formulate a query than to look at the second result page. This is consitent with the observation that the top-ranked results are much more attractive to a user, perhaps due to the reluctance to use the scrollbar. Moreover, judging from follow-up click behavior, top-ranked results seem to be of higher quality than lower-ranked ones. "Serendipity browsing" seems to influence user search behavior during recreational time periods. Finally, Web sites that are protected against "deep links" repel many visitors.

Our approach for gathering clickstreams is generic and not limited to Google, our example search engine. In future work, we plan to gather data from multiple search engines for the same user set during the same time period to facilitate comparisons across different search engines. The ultimative goal of this work is to not only gather new insights into users' search patterns, but also to harness the Web search clickstream to improve Web search capabilities.

9. REFERENCES

[1] Google basic search. `http://www.google.com/support/bin/static.py?page=searchguides.html&ctx=basics`.

[2] R. Atterer, M. Wnuk, and A. Schmidt. Knowing the user's every move—user activity tracking for website usability evaluation and implicit interaction. In *WWW*, 2006.

[3] P. Barford. Modeling, Measurement and Performance of World Wide Web Transactions. PhD thesis, Boston University, 2001.

[4] S. M. Beitzel, E. C. Jensen, A. Chowdhury, D. Grossman, and O. Frieder. Hourly analysis of a very large topically categorized web query log. In *ACM SIGIR*, 2004.

[5] M. Chau, X. Fang, and O. R. L. Sheng. Analysis of the query logs of a web site search engine. In *American Society for Information Science and Technology*, 2005.

[6] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma. Query expansion by mining user logs. In *IEEE Trans. Knowl. Data Eng. 15(4)*, 2003.

[7] B. Jansen and U. Pooch. Web user studies: A review and framework for future work. In *American Society of Information Science and Technology*, 2001.

[8] B. Krishnamurthy and J. Rexford. *Web Protocols and Practice*. Addison-Wesley, 2001.

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

[10] J. Luxenburger and G. Weikum. Query-log based authority analysis for web information search. In *WISE*, 2004.

[11] V. Paxson. Bro: A system for detecting network intruders in real-time. In *Computer Networks*, 1999.

[12] F. Radlinski and T. Joachims. Query chains: Learning to rank from implicit feedback. In *KDD*, 2005.

[13] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *ACM SIGIR*, 2005.

[14] C. Silverstein, M. Henzinger, H. Marais, and M. Moricz. Analysis of a very large altavista query log. Technical report, SRC Technical Note 014, 1998.

[15] A. Spink, B. J. Jansen, and H. C. Ozmultu. Use of query reformulation and relevance feedback by excite users. In *Internet Research: Electronic Networking Applications and Policy*, 2000.

[16] A. Spink, S. Koshman, M. Park, C. Field, and B. J. Jansen. Multitasking web search on vivisimo.com. In *ITCC*, 2005.

[17] A. Spink, D. Wolfram, B. Jansen, and T. Saracevic. Searching the web: The public and their queries. In *American Society for Information Science and Technology*, 2001.

[18] H. Weinreich, H. Obendorf, E. Herder, and M. Mayer. Off the beaten tracks: Exploring three aspects of web navigation. In *WWW*, 2006.