Defining a Session on Web Search Engines

Bernard J. Jansen
College of Information Sciences and Technology, The Pennsylvania State University, 329F IST Building, University Park, PA 16802. E-mail: jjansen@acm.org

Amanda Spink
Faculty of Information Technology, Queensland University of Technology, Gardens Point Campus, 2 George Street, GPO Box 2434, Brisbane QLD 4001, Australia. E-mail: ah.spink@qut.edu.au

Chris Blakely
Infospace, Inc.–Search & Directory, 601 108th Avenue NE, Suite 1200, Bellevue, WA 98004. E-mail: chris.blakely@Infospace.com

Sherry Koshman
School of Information Sciences, University of Pittsburgh, 610 IS Building, 135 N. Bellefield Avenue, Pittsburgh, PA 15260. E-mail: skoshman@sis.pitt.edu

Detecting query reformulations within a session by a Web searcher is an important area of research for designing more helpful searching systems and targeting content to particular users. Methods explored by other researchers include both qualitative (i.e., the use of human judges to manually analyze query patterns on usually small samples) and nondeterministic algorithms, typically using large amounts of training data to predict query modification during sessions. In this article, we explore three alternative methods for detection of session boundaries. All three methods are computationally straightforward and therefore easily implemented for detection of session changes. We examine 2,465,145 interactions from 534,507 users of Dogpile.com on May 6, 2005. We compare session analysis using (a) Internet Protocol address and cookie; (b) Internet Protocol address, cookie, and a temporal limit on intrasession interactions; and (c) Internet Protocol address, cookie, and query reformulation patterns. Overall, our analysis shows that defining sessions by query reformulation along with Internet Protocol address and cookie provides the best measure, resulting in an 82% increase in the count of sessions. Regardless of the method used, the mean session length was fewer than three queries, and the mean session duration was less than 30 min. Searchers most often modified their query by changing query terms (nearly 23% of all query modifications) rather than adding or deleting terms. Implications are that for measuring searching traffic, unique sessions may be a better indicator than the common metric of unique visitors. This research also sheds light on the more complex aspects of Web searching involving query modifications and may lead to advances in searching tools.

Received November 19, 2005; revised April 9, 2006; accepted June 23, 2006

 • Published online 27 February 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/asi.20564

Introduction

One can define a user episode on a Web search engine as a temporal series of interactions among a searcher, a Web system, and the content provided by that system within a specific period. During a Web search episode, the user may take several actions including submitting a query, viewing result pages, clicking on URLs, viewing Web documents, and returning to the Web search engine for query reformulation. However, it is possible that one searching episode will be composed of one or more sessions. We define a session from a contextual viewpoint as a series of interactions by the user toward addressing a single information need.

As with searching sessions on other information retrieval (IR) systems, the goal for the searcher is to locate relevant information that addresses an information need. For evaluation, one can view success or failure at the session level as the critical determinant in the user's perception of the Web search engine's performance. Therefore, the session level is a key paradigm for measuring the performance of Web search engines. Many researchers have analyzed Web searching sessions with the goal of using the information about users' activities to improve the performance of Web search engines. For example, Shneiderman, Byrd, and Croft (1998) presented suggestions for designing Web search engine interfaces that support the searching session strategies of users. In addition,

Hansen and Shriver (2001) examined navigation data using a session-level analysis to cluster search sessions.

Attempts at designing personalized Web systems relying on session-level data have taken a variety of approaches. CiteSeer (Lawrence, Giles, & Bollacker, 1999) utilizes an agent paradigm to recommend computer science and computer-science-related articles based on a user profile. CiteSeer (http://citeseer.ist.psu.edu/) offers a variety of searching assistance based on searcher interactions during the session. Jansen and Pooch (2001) designed a client side application for Web search engines that provided targeted searching assistance based on the user interactions during a session. The researchers noted that there are predictable patterns of when searchers seek and implement assistance from the system (Jansen, 2005, 2006a). These patterns may indicate when the searcher is open to assistance from the system, thereby avoiding task interruptions.

Using transactions logs, Anick (2003) examined the interactive query reformulation support of the AltaVista search engine for searchers. The researcher used a baseline group of AltaVista searchers given no query feedback and a feedback group offered 12 refinement terms along with the search results. There was no significant difference in searching performance between the two groups; however, Belkin et al. (2003) reported that query expansion assistance may be helpful and improve searching performance.

Yet, an obstacle with all of these applications relying on searching data is determining "exactly what is the session" in practical terms. That is, what is the set of interactions by the user that relates to a single information need? With traditional IR or library systems, one user usually could be distinguished from another user based on a logon; however, in the Internet environment, how to determine a session between a searcher and a Web search engines is an open question.

Related Studies

On the Web, the difficulty of how to define a search session is due in part to the stateless nature of the client–server relationship. Most Web search engine servers have used the Internet Protocol (IP) address of the client machine to identify *unique visitors*. With referral sites, Internet service providers (ISP), dynamic IP addressing, and common user terminals, it is not always easy to identify a single user session on a Web search engine. Therefore, a single IP address does not always correspond to a single user; however, this approach is commonly used for marketing purposes and Web site traffic reporting.

In response to the dynamically allocated IP situations, Web search engine researchers have moved to the use of cookies, along with IP addresses, for user identification. The use of cookies minimizes the session identification problem somewhat, but with common-access computers (i.e., computers at libraries, schools, labs, office shops, and manufacturing floors which many people share) along with spyware and cookie management software, one computer may not correspond to one searcher. Additionally, a single searcher may engage a search engine with multiple information needs simultaneously or in rapid succession (Spink, Özmutlu, & Özmutlu, 2002; Spink, Park, Jansen, & Pedersen, 2005) during a single searching episode. To consider these multiple information needs together presents significant problems for recommender systems and personalized online content.

Therefore, some search engines also use a temporal boundary along with cookies to help address this problem. This temporal boundary helps minimize the common user terminal issue and also helps delineate repeat searchers to a Web search engine who have returned, but with a new information need; however, this approach does not address the multiple information needs during a single searching episode issue. These methods (IP address; IP and cookie; and IP, cookie, and temporal boundary) all employ a mechanical definition of a session rather than a conceptual definition that defines a searching session within an information-seeking task.

There has been some research into using the query context to define the session. He, Göker, and Harper (2002) used contextual information from a Reuters transaction log and a version of the Dempster–Shafer theory in an attempt to identify search engine session boundaries. Using transaction log IP codes and query context, the researchers determined that the average Web user session duration was about 12 min. Jansen and Spink (2003) reported a mean session length of about 15 min, but with a sizable percentage of sessions being less than 5 min.

Özmutlu and Çavdur (2005) attempted to duplicate the findings of He et al. (2002), but the researchers reported that there were issues relating to implementation, algorithm parameters, and fitness function. Özmutlu and Çavdur (2005) and Özmutlu, Çavdur, Spink, and Özmutlu (2004, 2005) investigated the use of neural networks to automatically identify topic changes in sessions, reporting high percentages (72–97%) of correct identifications of topic shifts and topic continuations. Özmutlu et al. (2005) reported that neural networks were effective at topic identification, even if the neural network application was trained with data from another search engine transaction log. This line of research involved the use of sophisticated algorithmic approaches or extensive amounts of training data for topic identification. Whether one could obtain comparable results with simpler approaches was not investigated. In addition, these research studies did not contrast the findings of their approaches with other methods of session identification or reformulation classifications.

In contrast, this study examines three methods of session identification representing the major approaches taken to identify Web searching sessions. Each method is relatively straightforward and can be easily implemented for real-time identification of sessions without relying on probabilistic methods. Therefore, the computational costs are low, and the accuracy is high. We investigate the results for these three methods of session identification, and also examine quantitative techniques of identifying query reformulations within sessions. Finally, we compare the results from our dataset to results reported in other research.

DOI: 10.1002/asi

Research Question

Our research question is: What are the differences in results when using methods for identification of Web search engine sessions?

We investigated three methods for session identification: (a) IP address and cookie; (b) IP address, cookie, and a temporal cutoff; and (c) IP address, cookie, and context changes. Although there may be other techniques, these three methods represent the major approaches to session identification. We do not evaluate the sole use of an IP address for session identification, as it is commonly known to be inferior to the use of both IP address and cookie.

Research Design

Web Data

Dogpile.com (http://www.Dogpile.com/) is a meta-search engine owned by Infospace, Inc. When a searcher submits a query, Dogpile.com simultaneously submits the query to multiple other Web search engines, collects the results from each, removes duplicates results, and aggregates the remaining results into a combined ranked listing using a proprietary algorithm. Dogpile.com integrates the results of the four leading Web search indices (i.e., Ask Jeeves, Google, MSN, and Yahoo!) along with other search engines into its search results listing. Meta-search engines provide a unique service by presenting the alternate results provided by the various search engines, which have a low rate of overlap (Spink, Jansen, Blakely, & Koshman, 2006).

Dogpile.com has indexes for searching the *Web*, *Images*, *Audio*, and *Video* content, which searchers can access via tabs off the Dogpile.com interface. Dogpile.com also offers query reformulation assistance with alternate query suggestions listed in an *Are You Looking For?* area of the interface.

Figure 1 shows the Dogpile.com interface with query box, tabbed indexes, and "Are You Looking For?"

Hitwise (2005; http://www.clickz.com/stats/sectors/search_tools/article.p p/3528456) stated that Dogpile.com was the 9th most popular Web search engine in 2005 as measured by number of site visits. ComScore Networks (2003; http://www.comscore.com/press/release.asp?press=325) stated that Dogpile.com had the industry highest visitor-to searcher conversion rate of 83% (i.e., 83% of the visitors to the Dogpile.com site executed a search) in 2003.

Data Collection

We collected the records of searcher–system interactions in a transaction log that represents a portion of the searches executed on Dogpile.com on May 6, 2005. The original

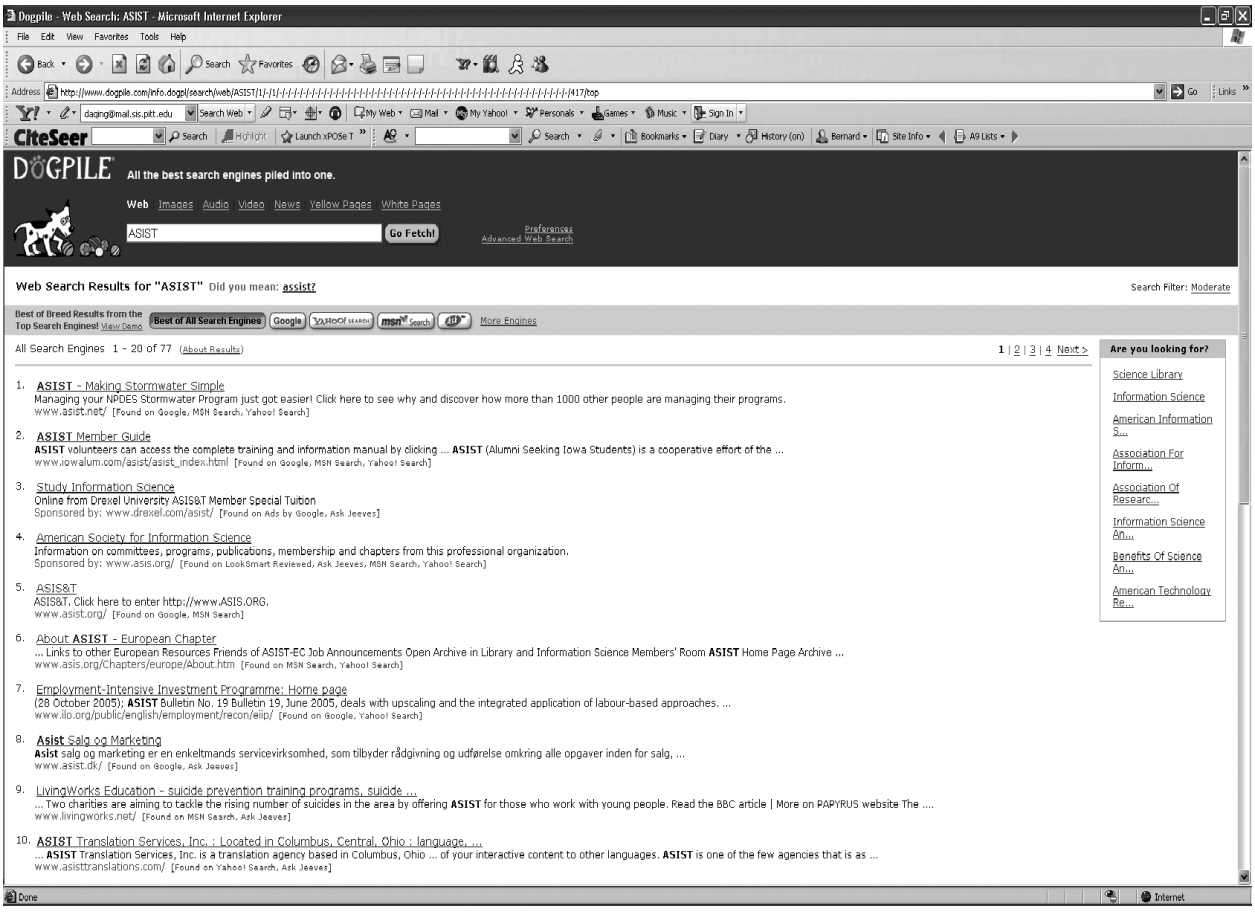

FIG. 1. Dogpile.com meta-search Interface.

DOI: 10.1002/asi

general transaction log contained 4,056,374 records, each containing seven fields:

- *User Identification:* a code to identify a particular computer.
- *Cookie:* an anonymous cookie automatically assigned by the Dogpile.com server to identify unique users on a particular computer.
- *Time of Day:* measured in hours, minutes, and seconds as recorded by the Dogpile.com server on the date of the interaction.
- *Query Terms:* the terms exactly as entered by the given user.
- *Location:* a code representing the geographic location of the user's computer as denoted by the computer's IP address.
- *Source:* the content collection that the user selects to search (e.g., *Web*, *Images*, *Audio*, *News*, or *Video*), with *Web* being the default.
- *Feedback:* a binary code denoting whether the query was generated by the *Are You Looking for?* query reformulation assistance provided by Dogpile.com (see Figure 1).

We imported the original flat ASCII transaction log file of 4,056,374 records into a relational database, and then generated a unique identifier for each record. We used four fields (*Time of Day*, *User Identification*, *Cookie*, and *Query*) to locate the initial query and then recreate the sequential series of actions from a particular user, determined by *User Identification* and *Cookie*. An analysis of the dataset showed that the interactions of Dogpile.com searchers was generally similar to Web searching on other Web search engines (Jansen, Spink, & Koshman, 2007).

Data Preparation

The terminology that we use in this research is similar to that used in other Web transaction log studies (cf. Jansen & Pooch, 2001; Park, Bae, & Lee, 2005) for directed searching on Web search engines.

- *Term:* a series of characters within a query separated by white space or other separator.
- *Query:* string of terms submitted by a searcher in a given instance of interaction with the search engine.
  - *Initial query:* first query submitted in a session by a given user.
  - *Subsequent query:* a query within a session that is not the *initial query*.

At the session level, we deviate from earlier work. In prior studies (cf. Beitzel, Jensen, Chowdhury, Grossman, & Frieder, 2004; Park et al., 2005), researchers generally had defined a *session* as *a series of queries submitted by a user during one episode of interaction between the user and the Web search engine*. Researchers have added certain operational constraints to this definition, including Web pages (Hansen & Shriver, 2001) and temporal cutoffs between query submissions (Silverstein, Henzinger, Marais, & Moricz, 1999). Each of these constraints, or lack of constraints, affects what is a *session*. We investigate the effect of some of these constraints in this article.

How to constrain a session affects other metrics concerning sessions, namely:

- *Session Length:* the number of queries submitted by a searcher during a defined period of interaction with the search engine.
  - How one defines the session boundaries is critically important in determining session length.
- *Sessions Duration:* the period from the submission of the *initial query* through the submission of *final query*.
  - Determining the *initial query* is relatively straightforward. Determining the *final query* again depends on how one defines the session boundaries conditions. For example, if one uses only IP address with no other conditions, then the session duration is the period from the *initial query* until the searcher departed the search engine for the last time (i.e., does not return to the search engine). If one includes other constraints, then there may be multiple sessions by a single searcher within a given episode.
  - As a limitation, unless one has client-side data, search engine logs can measure only the total user time on the search engine, defined as the time spent viewing the first and subsequent results lists and documents, except the final Web document regardless of any other constraints on the session. This final viewing time is not available since the search engine servers record the time stamp. Naturally, the time between visits from the Web document to the server may not have been entirely spent viewing the Web document or interacting with the search.

This view of directed search on the Web certainly ignores browsing for information. There does not currently appear to be a consensus regarding the precise definition of browsing. Bodoff (2006) defined browsing as "actively looking through information (active) or keeping one's eyes open for information (passive), without a particular problem to solve or question to answer (unfocused need)" (p. 70). Bodoff (2006) also provided a nice review of browsing definitions within certain contents and contrasts browsing with directed search, such as that on a Web search engine. Our focus in this research is on directed searching.

Removing Agent Queries

For this research, we are interested in queries submitted by humans, and the transaction log contained queries from both human users and agents. There is no recognized methodology for precisely identifying human from nonhuman submissions in a search engine transaction log. Therefore, researchers usually use a temporal or interaction cutoff (Jansen, Spink, & Pedersen, 2005; Silverstein et al., 1999).

We chose the interaction cutoff approach by removing all sessions with 100 or more queries to be consistent with the approach taken in previous transaction log studies (cf. Jansen & Spink, 2005a; Jansen et al., 2005; Spink & Jansen, 2004). This cutoff is considerably greater than the reported mean number of queries (Jansen, Spink, & Saracevic, 2000) for human Web searchers. This increased the probability that we were not excluding any human searches. This cutoff

DOI: 10.1002/asi

likely introduced some agent sessions; however, we were generally certain that we had included most of the queries submitted primarily by human searchers.

Research has reported that the use of software agents to gather information from search engines has rapidly increased in recent years. For example, in a 2002 report of AltaVista, nearly 46% of submissions were likely from automated programs (Jansen, Mullen, Spink, & Pedersen, 2006). Our process removed a similar percentage of likely agent queries.

Session Analysis Using Multiple Methods

Returning to our research question (i.e., What are the differences in results when using alternative methods for identification of Web search engines sessions?), we investigated defining sessions using three approaches.

Method 1: IP and cookie. For the first approach, we defined the session as the period from the first interaction by the searcher with Dogpile.com through the last interaction as recorded in the transaction log. We used the searcher's IP address and the browser cookie to determine the *initial query* and all *subsequent queries* to establish *session length*. The *session duration* was the period from the time of the *initial query* to the time of the last interaction with the search engine. A change in either IP address or cookie always identified a new session.

The algorithm for Method 1 is shown in Figure 2.

Algorithm: *IP and Cookie Session Identification*
Assumptions:
1. Null queries and page request queries are removed.
2. Transaction log is sorted by IP address, cookie, and time (ascending order by time).

Input: Record R_i with IP address (IP_i) and cookie (K_i), and record R_{i+1} with IP address (IP_{i+1}) and cookie (K_{i+1}).

Variables: S_x = count of sessions

Output: Session Identification, $\mathbf{S_x}$
begin
Move to R_i

Store values for IP_i *and* K_i

$\mathbf{S_x} = 1$

While not end of file
Move to R_{i+1}

If ($IP_i = IP_{i+1}$ *and* $K_i = K_{i+1}$) *then* $\mathbf{S_x}$

Elseif
{
$\mathbf{S_x} = \mathbf{S_x} + 1$
}
(R_{i+1} *now becomes* R_i)

Store values for R_{i+1} *as* IP_i *and* K_i

end loop
end

FIG. 2. Algorithm used to identify sessions for Method 1.

Method 2: IP, cookie, and temporal cutoff. For the second approach to session identification, we again used the searcher's IP address and the browser cookie to determine the *initial query* and *subsequent queries*. In this method, however, we used a 30-min period between interactions as the session boundary. For example, if a searcher submitted two queries within a 30-min period, this searching episode would be counted as one session; however, if a searcher submitted two queries and the interaction period between each query was longer than 30 min, this episode would be counted as two sessions.

We selected the 30-min period based on the industry standard view of a session (e.g., see OneClick.com and Nielsen Netranking). This 30-min norm is most likely based on Catledge and Pitkow's (1995) report that the typical Web session duration was 25.5 min on average, although this session metric included browsing activities. However, other temporal metrics have been used. Silverstein et al. (1999) assigned a temporal cutoff of 5 min between interactions as the maximum session duration. Montgomery and Faloutsos (2001) used a 125-min session period, stating that various temporal cutoffs did not substantially affect results. Additionally, Jansen and Spink (2003) and He et al. (2002) reported that the average search engine session is about 15 min based on IP address alone.

The algorithm for Method 2 is shown in Figure 3.

Method 3: IP, cookie, and content change. For the third session-identification approach, we used a contextual method to identify sessions. We once again used the searcher's IP address and the browser cookie to determine the *initial query* and *subsequent queries*. But instead of using a temporal cutoff, we used changes in the content of the user queries.

For this method, we assigned each query into a mutually exclusive group based on an IP address, cookie, query content, use of the feedback feature, and query length. The classifications are:

- *Assistance:* The current query was generated by the searcher's selection of an *Are You Looking For?* query (see Figure 1).
- *Content Change:* The current query is identical but executed on another content collection.
- *Generalization:* The current query is on the same topic as the searcher's previous query, but the searcher is now seeking more general information.
- *New:* The query is on a new topic.
- *Reformulation:* The current query is on the same topic as the searcher's previous query, and both queries contain common terms.
- *Specialization:* The current query is on the same topic as the searcher's previous query, but the searcher is now seeking more specific information.

DOI: 10.1002/asi

Algorithm: *IP, Cookie, and Time Identification*
Assumptions:
1. Null queries and page request queries are removed.
2. Transaction log is sorted by IP address, cookie, and time (ascending order by time).

Input: Record R_i with IP address (IP_i), cookie (K_i), and time T_i, and record R_{i+1} with IP address (IP_i), cookie (K_i), and time T_i.

Variables:
D = serial time for 30 min
S_x = count of sessions

Output: Search pattern, *SP*

begin
Move to R_i

Store values for IP_i, K_i, *and* T_i

$\mathbf{S_x} = 1$

While not end of file
Move to R_{i+1}

If ($IP_i = IP_{i+1}$ *and* $K_i = K_{i+1}$ *and* $T_{i+1} < T_i + D$)
then $\mathbf{S_x}$

Elseif
{
$\mathbf{S_x} = \mathbf{S_x} + 1$
}
(R_{i+1} *now becomes* R_i)

Store values for R_{i+1} *as* IP_i, K_i *and* T_i
end loop
end

FIG. 3. Algorithm used to identify sessions for Method 2.

The *initial query* (Q_i) from a unique IP address and cookie always identified a new session. In addition, if a *subsequent query* (Q_{i+1}) by a searcher contained no terms in common with the previous query (Q_i), we also deemed this the start of a new session. Naturally, from an information-need perspective, these sessions may be related at some level of abstraction; however, with no terms in common, one also can make the case that the information state of the user changed, either based on the results from the Web search engine or from other sources (Belkin, Oddy, & Brooks, 1982). In addition, from a system perspective, two queries with no terms in common represent different executions to the inverted file index and content collection.

We classified each query using an application that evaluated each record in the database. Building from He et al. (2002), the algorithm for the application is shown in Figure 4.

Results

We now discuss our results, relating to our research question, focusing on both session length and session duration.

Algorithm: **Search** *Pattern Identification*
Assumptions:
1. Null queries and page request queries are removed.
2. Transaction log is sorted by IP address, cookie, and time (ascending order by time).

Input: Record R_i with IP address (IP_i), cookies (K_i), query Q_i, feedback F_i, and query QL_i; and record R_{i+1} with IP address (IP_{i+1}), cookies (K_{i+1}), query Q_{i+1}, feedback F_{i+1}, and query QL_{i+1}.

Variables:
B = {tlt $\in Q_i$ ^ t $\in Q_{i+1}$} // terms in common
C = {tlt $\in Q_i$ ^ t $\notin Q_{i+1}$} // terms that appear in Q_i only
D = {tlt $\notin Q_i$ ^ t $\in Q_{i+1}$} // terms that appear in Q_{i+1} only
E = {1 if $QL_i = QL_{i+1}$} // queries QL_i and QL_{i+1} are the same length; default is 0.
G = {1 if $QL_i > QL_{i+1}$} // query QL_i has more terms than QL_{i+1}; default is 0.
H = {1 if $QL_i < QL_{i+1}$} // query QL_i has less terms than QL_{i+1}; default is 0.

Output: Search pattern, *SP*
begin
Move to R_i
Store values for IP_i, K_i, Q_i, F_i, *and* QL_i
SP = New //*default value for first* R_i *in record set*
While not end of file
Move to R_{i+1}
If ($IP_i \neq IP_{i+1}$ *and* $K_i \neq K_{i+1}$) *then SP* = New
Elseif
{
Calculate values for B, C, D, F, G, and H
If $F_{i+1} = 1$ *then SP* = Assistance
Elseif ($B \neq \emptyset$ ^ $C \neq \emptyset$ ^ $D = \emptyset$ ^ $E = 0$ ^ $G = 1$ ^ $H = 0$)
then SP = Generalization
Elseif ($B \neq \emptyset$ ^ $C \neq \emptyset$ ^ $D \neq \emptyset$ ^ $E = 0$ ^ $G = 1$ ^ $H = 0$)
then SP = Generalization with Reformulation
Elseif ($B \neq \emptyset$ ^ $C = \emptyset$ ^ $D \neq \emptyset$ ^ $E = 0$ ^ $G = 0$ ^ $H = 1$)
then SP = Specialization
Elseif ($B \neq \emptyset$ ^ $C \neq \emptyset$ ^ $D \neq \emptyset$ ^ $E = 0$ ^ $G = 0$ ^ $H = 1$)
then SP = Specialization with Reformulation
Elseif ($B \neq \emptyset$ ^ $C \neq \emptyset$ ^ $D \neq \emptyset$ ^ $E = 1$ ^ $G = 0$ ^ $H = 0$)
then SP = Reformulation
Elseif ($B \neq \emptyset$ ^ $C = \emptyset$ ^ $D = \emptyset$ ^ $E = 1$ ^ $G = 0$ ^ $H = 0$)
then SP = Content Change
Elseif SP = New
}
(R_{i+1} *now becomes* R_i)

Store values for R_{i+1} *as* IP_i, K_i, Q_i ,F_i, *and* QL_i
end loop
Move to R_1
$\mathbf{S_x} = 0$
While not end of file

If SP = New *Then* ($S_x = S_x + 1$)
end loop
end

FIG. 4. Algorithm used to identify sessions for Method 3.

Session Lengths

We begin by examining differences in session lengths, displayed in Table 1.

Method 1 is the approach used to define a session in many Web-searching studies (cf. Spink & Jansen, 2004).

DOI: 10.1002/asi

TABLE 1. Comparing session lengths.

	Method 1: IP and cookie		Method 2: IP, cookie, and 30-min time limit		Method 3: IP, cookie, and query content	
Session length	Occurrences	%	Occurrences	%	Occurrences	%
1	288,231	53.92	533,950	81.15	691,672	71.64
2	88,875	16.63	81,224	12.34	153,056	15.85
3	47,664	8.92	24,840	3.78	58,537	6.06
4	29,345	5.49	9,219	1.40	27,134	2.81
5	19,655	3.68	3,822	0.58	14,168	1.47
6	13,325	2.49	1,755	0.27	7,745	0.80
7	9,549	1.79	944	0.14	4,430	0.46
8	7,169	1.34	622	0.09	2,791	0.29
9	5,497	1.03	442	0.07	1,769	0.18
10	4,130	0.77	331	0.05	1,193	0.12
>10	21,067	3.94	871	0.13	2,944	0.30
	534,507	100.00	658,020	100.00	965,439	100.00

Table 1 shows that more than 79% of the sessions were three or fewer queries, using Method 1. Via Method 1, the mean session length was 2.85 queries, with an *SD* of 4.43. The maximum session length was 99, and the minimum was one query. This finding is similar to other analyses of Web search engine trends. For example, Spink, Jansen, Wolfram, & Saracevic (2002) reported short sessions during Web-searching sessions. Jansen and Spink (2005a), in their analysis of European searching, noted a similar inclination for short sessions as measured by number of queries submitted. However, AltaVista users conducted slightly longer sessions (Jansen et al., 2005). Koshman, Spink, and Jansen (2006) found that 1 in 5 Vivisimo users entered only two queries during their session. Koshman et al. (2006) used IP and cookie on given days to define sessions.

Method 2 has been used by various researchers (cf. Montgomery & Faloutsos, 2000, 2001; Park et al., 2005; Silverstein et al., 1999), although most employed various time limits ranging from 5 to 120 min. Using Method 2, 97% of the sessions were three or fewer queries, which is an 18 percentage-point increase over Method 1. The mean session length was 2.31 queries (15.4% decrease), with an *SD* of 3.18 queries. The maximum session length was 99, and the minimum was one query, which is no change from Method 1.

These results parallel more directly the percentage reported by Silverstein et al. (1999) that 95% of queries were three queries or fewer using a 5-min limit between query submissions. Montgomery and Faloutsos (2001) defined a session as less than 120 min of inactivity between viewings, although they dealt primarily with browsing activity rather than with searching. The researchers report that they tried several cutoff values, but the choice did not substantially alter the findings (Montgomery & Faloutsos, 2000). Catledge and Pitkow (1995) reported that the mean between each user interface event was 9.3 min, and they used session boundaries of 25.5 min between events, although it is unclear where this temporal boundary came from. Catledge and Pitkow also included browsing activities in their session activities.

Using Method 3, we see from Table 1 that 93% of the sessions were three or fewer queries. By way of Method 3, the mean session length was 2.31 queries, with an *SD* of 1.56 queries. The maximum session length was 57, and the minimum was one query. Note that the mean session length was the same as for Method 2, although the *SD* was about half. Generally, it appears that Method 3 provides a more granular definition of the session based on the reduced variance in the number of queries per session. Using 534,507 sessions as the base, Method 2 resulted in a 23% increase in the number of sessions, and Method 3 resulted in an 82% increase in sessions.

We investigated whether these three methods produced significantly different results by performing a chi-square test. The chi-square is a nonparametric test of statistical significance. The chi-square test tells us whether samples are different enough in some characteristic, from which we can generalize that the populations also are different.

A chi-square goodness of fit shows that the three methods are statistically different, $\chi^2(10) = 29.73$, $p < .01$; critical value of $\chi^2 = 23.209$. So, the methods are significantly dissimilar in their classification of sessions by number of queries.

Session Durations

What is the effect of these methods on session duration? Examining session durations, we see in Table 2 that Method 1 shows a large percentage of very short session durations.

The mean session duration was 26 min 32 s, with an *SD* of 1 hr 36 min 25 s. The maximum session was just under 24 hr (23:57:51), and the minimum session was 0 (i.e., the user submitted one query and performed no other search activity on the search engine during the session). This is more than twice that reported by He et al. (2002), who reported a session duration of 12 min.

Using Method 2, the absolute numbers have increased, but the percentages of very short session durations remains relatively stable; however, the mean session duration was

DOI: 10.1002/asi

TABLE 2. Comparing session durations.

	Method 1: IP and cookie		Method 2: IP, cookie, and 30-min time limit		Method 3: IP, cookie, and query content	
Session duration	Occurrences	%	Occurrences	%	Occurrences	%
<1 min	302,653	56.62	372,983	56.68	794,765	82.32
1 to <5 min	83,236	15.57	93,251	14.17	86,358	8.94
5 to <10 min	36,347	6.80	55,956	8.50	28,044	2.90
10 to <15 min	19,806	3.71	36,020	5.47	12,277	1.27
15 to <30 min	27,210	5.09	61,767	9.39	13,752	1.42
30 to <60 min	18,441	3.45	30,790	4.68	12,628	1.31
60 to <120 min	14,236	2.66	6,615	1.01	7,524	0.78
120 to <180 min	8,262	1.55	506	0.08	3,320	0.34
180 to <240 min	5,901	1.10	76	0.01	1,919	0.20
>240 min	18,415	3.45	56	0.01	4,852	0.50
	534,507	100.00	658,020	100.00	965,439	100.00

6 min 36 s, with an *SD* of 16 min 5 s. This is closer to the large number of sessions at approximately 5 min reported by Jansen and Spink (2003). The maximum session was just under 24 hr (23:57:24). As with Method 1, the maximum session length is cause for concern, as it seems highly unlikely that a single searcher would spend 24 hr submitting queries to a search engine. More than likely, these methods are inadvertently combining sessions or the database still contains agent submissions.

Using Method 3, the percentages of very short session durations again remains relatively stable. The mean session duration was 5 min 15 s, with an *SD* of 39 min 22 s. The maximum session duration, as in Method 2, was just under 24 hr (23:41:53).

Comparing the mean session durations, the mean using Method 1 is 333% greater than the mean session duration using Method 2 and 420% greater than the mean session duration using Method 3. This outcome is in contrast to that reported by Montgomery and Faloutsos (2001), where changes in temporal cutoffs for the session boundaries did not substantially alter results.

Accuracy of Classification

We conducted a verification of our classification algorithms (both time-based, Method 2, and query-content-based, Method 3) by manually classifying 2,000 queries. We arrived at five categories of errors, developed a posteriori:

1. *Misspelling:* A word was misspelled or a previously misspelled word causing a change resulting in a misclassification (causes a false *New* or *Reformulation*).
2. *Cookie:* Either cookie not defined or change in cookie, but not a change in user (causes a false *New*).
3. *Special character change:* The original query contained special characters (causes a false *New* or *Reformulation*).
4. *Time gap:* Time gap between queries was too large to be considered a session, but Q_i and Q_{i-1} were still related (causes a false *New*).
5. *Other:* A miscellaneous collection of other reasons (causes a false *New*).

TABLE 3. Misclassifications of queries from 2,000 query samples.

Type of Misclassifications	Occurrences	%
Misspelling	52	47.27
Cookie	23	20.91
Special character change	5	4.55
Time gap	21	19.09
Other	9	8.18
	110	100.00

We see from Table 3 that most of the errors were due to misspellings (i.e., the algorithm counted the word as a new term when in reality the searcher had misspelled a term in the original query and corrected the term in the subsequent query. Most misspellings occurred due to missing spaces between words. However, the sum total of all misclassifications for Method 2 was 1.05% (an accuracy rate of 98.95% for the algorithm). The total of misclassifications for Method 3 was 4.45%, resulting in a 95.55% accuracy rate for the algorithm. This is certainly a reasonable outcome, as there are more factors involved with Method 3 compared to those in Method 2. Therefore, the probability of error increases, and thus the two error rates cannot reasonable be compared to each other. Finally, the accuracy of classifications for both methods is quite high, and Method 3 addresses the contextual aspects that Method 2 does not.

Discussion

We explored three alternative methods for detection of session boundaries using 2,465,145 interactions from 534,507 users of Dogpile.com recorded on May 6, 2005. We compared three methods of session identification: (a) *using IP address and cookie*, (b) *IP address, cookie, and a temporal limit on intrasession interactions*, and (c) *IP address, cookie,*

DOI: 10.1002/asi

TABLE 4. Query reformulation.

Search Patterns	Occurrence	%	Occurrence (excluding *New*)	% (excluding *New*)
New	964,780	63.34	—	—
Reformulation	126,901	8.33	126,901	22.73
Assistance	124,195	8.15	124,195	22.25
Specialization	90,893	5.97	90,893	16.28
Content change	65,949	4.33	65,949	11.81
Specialization with reformulation	55,531	3.65	55,531	9.95
Generalization with reformulation	54,637	3.59	54,637	9.78
Generalization	40,186	2.64	40,186	7.20
	1,523,072	100.00	558,292	100.00

and query reformulation patterns. Our results show that defining sessions by query content (Method 3) provides the best session identification with an extremely high accuracy rate. Comparatively, Method 1 appears to artificially extend both session length and duration. Method 2 appears to artificially shorten session length and duration. By relying on IP address and cookie as a basis, plus content changes between queries, Method 3 provides the best contextual identification of Web sessions within a user episode on a Web search engine.

Method 3, using IP address, cookie, and query-content changes, appears to provide the most detailed method for session identification with both session length and session duration. Since the method does not rely on probability methods, it can be calculated in real time with near total accuracy of new session identification. Using this content approach, Web search systems can develop automated assistance interfaces, such as those reported in Jansen and McNeese (2005), that provide session-level searching assistance to Web engine users.

As an example, Table 4 presents the query modification executed by searchers during their searching episodes.

We see from Table 4 that more than 8% of the query modifications were for *Reformulation*, with another approximately 8% of query modifications resulting from system *Assistance*. If we exclude the *New* queries, *Reformulation* and *Assistance* account for nearly 45% of all query modifications. This finding would seem to indicate that a substantial portion of searchers go through a process of defining their information need by exploring various terms and system feedback to modify the query as an expression of their information need. Another 16% of query modifications are *Specialization*, supporting prior reports that precision is a primary concern for Web searchers (Jansen & Spink, 2005b). With this tighter view of a session, Web search engines can personalize more effectively for searching assistance, content, or online advertising.

The detection of Web searching sessions is a critical area of research for developing more supportive searching systems, especially in the more complex searching environments of exploratory searching and multitasking. The method presented in this research relies on the content of searchers' queries, along with other data collected by the search engine, to identify searching sessions. The method is advantageous for real-time system implementation.

Conclusions and Further Research

For future research, these algorithms may be used as models to facilitate cross-system investigations. An attempt to standardize session detection also would enhance comparative transaction log analyses. We are currently conducting qualitative analysis of Dogpile users' query reformulation that we will compare with the results reported in this article. In addition, several searcher–system interactions can be recorded by the Web search engine server. However, there are other actions such as Back, Forward, Bookmark, and Scrolling, among others, that occur on the client-side computer. The server does not record these actions. We are investigating the development of server–client tools that can monitor the entire set of searcher actions during a session (Jansen, 2006b).

Acknowledgments

We thank Infospace, Inc. for providing the Web search engine transaction log data, without which we could not have conducted this research. We also thank Ms. Danielle Booth for coding the algorithm for Method 3 presented in this article.

References

Anick, P. (2003). Using terminological feedback for Web search refinement—A log-based study. Proceedings of the 26th annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 88–95).

Beitzel, S.M., Jensen, E.C., Chowdhury, A., Grossman, D., & Frieder, O. (2004). Hourly analysis of a very large topically categorized web query log. Proceedings of the 27th annual International Conference on Research and Development in Information Retrieval (pp. 321–328).

Belkin, N., Cool, C., Kelly, D., Lee, H.-J., Muresan, G., Tang, M.-C., et al. (2003). Query length in interactive Information Retrieval. Proceedings of the 26th annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 205–212).

Belkin, N., Oddy, R., & Brooks, H. (1982). ASK for Information Retrieval: Parts 1 & 2. Journal of Documentation, 38(2), 61–71, 145–164.

DOI: 10.1002/asi

Bodoff, D. (2006). Relevance for browsing, relevance for searching. Journal of the American Society of Information Science and Technology, 57(1), 69–86.

Catledge, L.D., & Pitkow, J.E. (1995). Characterizing browsing strategies in the World Wide Web. Computer Network and ISDN Systems, 27(1), 1065–1073.

Hansen, M.H., & Shriver, E. (2001). Using navigation data to improve IR functions in the context of web search. Proceedings of the 10th International Conference on Information and Knowledge Management (pp. 135–142).

He, D., Göker, A., & Harper, D.J. (2002). Combining evidence for automatic Web session identification. Information Processing & Management, 38(5), 727–742.

Jansen, B.J. (2005). Seeking and implementing automated assistance during the search process. Information Processing & Management, 41(4), 909–928.

Jansen, B.J. (2006a). Using temporal patterns of interactions to design effective automated searching assistance systems. Communications of the ACM, 49(4), 72–74.

Jansen, B.J. (2006b). Search log analysis: What is it; what's been done; how to do it. Library and Information Science Research, 28(3), 407–432.

Jansen, B.J., & McNeese, M.D. (2005). Evaluating the effectiveness of and patterns of interactions with automated searching assistance. Journal of the American Society for Information Science and Technology, 56(14), 1480–1503.

Jansen, B.J., Mullen, T., Spink, A., & Pedersen, J. (2006). Automated gathering of Web information: An in-depth examination of agents interacting with search engines. ACM Transactions on Internet Technology, 6, 442–464.

Jansen, B.J., & Pooch, U. (2001). Web user studies: A review and framework for future work. Journal of the American Society for Information Science and Technology, 52(3), 235–246.

Jansen, B.J., & Spink, A. (2003). An analysis of Web information seeking and use: Documents retrieved versus documents viewed. Proceedings of the 4th International Conference on Internet Computing (pp. 65–69).

Jansen, B.J., & Spink, A. (2005a). An analysis of Web searching by European Alltheweb.com users. Information Processing & Management, 41(2), 361–381.

Jansen, B.J., & Spink, A. (2005b). How are we searching the World Wide Web? A comparison of nine search engine transaction logs. Information Processing & Management, 42(1), 248–263.

Jansen, B.J., Spink, A., & Koshman, S. (2007). Web searcher interaction with the Dogpile.com metasearch engine. Journal of the American Society for Information Science and Technology, 58(5), 744–755.

Jansen, B.J., Spink, A., & Pedersen, J. (2005). Trend analysis of AltaVista Web searching. Journal of the American Society for Information Science and Technology, 56(6), 559–570.

Jansen, B.J., Spink, A., & Saracevic, T. (2000). Real life, real users, and real needs: A study and analysis of user queries on the Web. Information Processing & Management, 36(2), 207–227.

Koshman, S., Spink, A., & Jansen, B.J. (2006). Web searching on the Vivisimo search engine. Journal of the American Society for Information Science and Technology, 57(14), 1875–1887.

Lawrence, S., Giles, C.L., & Bollacker, K. (1999). Digital libraries and autonomous citation indexing. IEEE Computer, 32(6), 67–71.

Montgomery, A., & Faloutsos, C. (2000). Trends and patterns of WWW browsing behaviour. Retrieved October 6, 2005, from http://pages.cpsc.ucalgary.ca/~saul/personal/other_pubs/web_trends.pdf

Montgomery, A., & Faloutsos, C. (2001). Identifying web browsing trends and patterns. IEEE Computer, 34(7), 94–95.

Özmutlu, H.C., & Çavdur, F. (2005). Application of automatic topic identification on Excite Web search engine data logs. Information Processing & Management, 41(5), 1243–1262.

Özmutlu, H.C., & Çavdur, F., Spink, A., & Özmutlu, S. (2004). Neural network applications for automatic new topic identification on Excite Web search engine data logs. Proceedings of the Association for the American Society for Information Science and Technology (ASIST 2004) (pp. 1–10).

Özmutlu, H.C., Çavdur, F., Spink, A., & Özmutlu, S. (2005). Cross validation of neural network applications for automatic new topic identification. Proceedings of the Association for the American Society of Information Science and Technology (ASIST 2005) (pp. 1–10).

Park, S., Bae, H., & Lee, J. (2005). End user searching: A Web log analysis of NAVER, a Korean Web search engine. Library & Information Science Research, 27(2), 203–221.

Shneiderman, B., Byrd, D., & Croft, W.B. (1998). Sorting out searching: A user-interface framework for text searches. Communications of the ACM, 41(4), 95–98.

Silverstein, C., Henzinger, M., Marais, H., & Moricz, M. (1999). Analysis of a very large Web search engine query log. SIGIR Forum, 33(1), 6–12.

Spink, A., & Jansen, B.J. (2004). Web search: Public searching of the Web. New York: Kluwer.

Spink, A., Jansen, B.J., Blakely, C., & Koshman, S. (2006). A study of results overlap and uniqueness among major Web search engines. Information Processing & Management, 42, 1379–1391.

Spink, A., Jansen, B.J., Wolfram, D., & Saracevic, T. (2002). From E-sex to E-commerce: Web Search Changes. IEEE Computer, 35(3), 107–111.

Spink, A., Özmutlu, H.C., & Özmutlu, S. (2002). Multitasking information seeking and searching processes. Journal of the American Society for Information Science and Technology, 53(8), 639–652.

Spink, A., Park, M., Jansen, B.J., & Pedersen, J. (2005). Multitasking during web search sessions. Information Processing & Management, 42(1), 264–275.

DOI: 10.1002/asi