Access Patterns for Robots and Humans in Web Archives

Yasmin AlNoamany, Michele C. Weigle, Michael L. Nelson
Old Dominion University
Norfolk, VA, USA
{yasmin, mweigle, mln}@cs.odu.edu

ABSTRACT

Although user access patterns on the live web are well-understood, there has been no corresponding study of how users, both humans and robots, access web archives. Based on samples from the Internet Archive's public Wayback Machine, we propose a set of basic usage patterns: Dip (a single access), Slide (the same page at different archive times), Dive (different pages at approximately the same archive time), and Skim (lists of what pages are archived, i.e., TimeMaps). Robots are limited almost exclusively to Dips and Skims, but human accesses are more varied between all four types. Robots outnumber humans 10:1 in terms of sessions, 5:4 in terms of raw HTTP accesses, and 4:1 in terms of megabytes transferred. Robots almost always access TimeMaps (95% of accesses), but humans predominately access the archived web pages themselves (82% of accesses). In terms of unique archived web pages, there is no overall preference for a particular time, but the recent past (within the last year) shows significant repeat accesses.

Categories and Subject Descriptors

H.3.7 [**Digital Libraries**]: Web Archives—*Retrieval models*

Keywords

Web Archiving, Web Server Logs, Web Usage Mining, User Access Patterns, Web Robot Detection

1. INTRODUCTION

The web has become an integral part of our lives, shaping how we get news, shop, and communicate. In turn, web archives have become a significant repository of our recent history and cultural heritage. The Internet Archive [19] is the largest and oldest of the various web archives, holding over 240 billion web pages with archives as far back as 1996 [13]. Access to this vast archive is available through the Wayback Machine [30], which sees about 82 million requests per day, based on our dataset.

Previous work has studied how users access the live web [31] and search engines [12], but few studies have investigated how users access web archives. Understanding the current demand for access to web archives can provide insights into how to make the best use of limited archiving and access resources.

In this paper, we provide an analysis of user accesses to a large web archive. We examine a set of anonymized Wayback Machine server access logs from February 2012. We investigate the differences between human and robot accesses of the Wayback Machine, identify four major web archive access patterns, and uncover the temporal preference for web archive access. In particular, we find that robots (such as crawlers and spiders) account for 91% of all sessions and 93% of all page requests. Yet, robots only outnumber humans 5:4 in terms of raw, unfiltered requests and 4:1 in terms of megabytes transferred. Humans download more information per session, as they typically download embedded resources (e.g., images and stylesheets), which robots ignore.

We introduce four basic user access patterns of web archives: Dip, Slide, Dive, and Skim. Dip is requesting for a single URI. Slide is browsing different archived copies of the same URI. Dive is following the hyperlinks of a page, but staying near the same datetime. Skim is requesting only the TimeMaps (list of all archived copies for a specific original resource). Robots exhibit the Dip and Skim patterns equally, both about 49% of their sessions, and almost exclusively request TimeMaps. Humans exhibit the Dip (39%) and Dive (30%) patterns the most and access archived pages significantly more than TimeMaps.

This paper is organized as follows. Definitions of important terms and a review of related work on web usage mining and web archive studies are presented in Section 2. Section 3 contains the patterns for accessing web archives along with an explanation and example for each pattern. A description of the Wayback Machine's web server logs, the dataset we used in the analysis, and the methodology of this study are presented in Section 4. Section 5 contains the results from analyzing the Wayback Machine access logs. Future work and conclusions are presented in Section 6.

2. RELATED WORK

Despite the significance of web archives in preserving web heritage, the aspect of web archive usage has been overlooked. The only previous related work is a study of the search behavior characterization for web archives [7]. We highlight this work, but first we define the terms for our discussion.

JCDL'13, July 22–26, 2013, Indianapolis, Indiana, USA.

2.1 Memento Terminology

Memento [33] is an HTTP protocol extension which enables time travel on the web by interlinking the current resources with their prior state. Memento introduces content negotiation in the datetime dimension using a special HTTP header, Accept-Datetime [32]. Memento defines the following terms:

- URI-R denotes the original resource. It is the resource as it used to appear on the live web; it may have 0 or more mementos (URI-Ms).
- URI-M is an archived snapshot for the URI-R at a specific datetime, which is called Memento-Datetime. e.g., URI-M$_i$= URI-R@t_i.
- URI-T denotes a TimeMap, a resource that provides a list of mementos (URI-Ms) for a URI-R with their Memento-Datetimes, e.g., $URI{-}T(URI{-}R) = \{URI{-}M_1, URI - M_2, ..., URI - M_n\}$. (We will also refer to a TimeMap as TM URI-R, to emphasize the URI-R in later examples)

Although we use Memento terminology, the logs we analyze are from the public access Wayback Machine and not the Memento API.

2.2 Web Usage Mining

The breadth and depth of research in the area of web usage mining is massive and increasing [3, 14, 23, 5]. Web usage mining involves discovering usage patterns from web data using data mining [25]. The results obtained from web usage mining can be used in different applications, such as web traffic analysis, site modification, system improvement, personalization, business intelligence, and usage characterization. Our study provides traffic analysis and usage characterization by providing abstract models for accessing web archives.

Adams et al. explored the usage patterns of scientific and historical data repositories [1]. However, their study focused on a variety of archive types (e.g., public vs. private, digital but non-web resources) and does not directly address the issue of archiving the web. The only web usage mining research that has been conducted on the usage of web archives is the study of search behavior characterization of web archives based on a quantitative analysis of the Portuguese Web Archive (PWA) search logs [7]. The authors introduced a comparison between search patterns of web archives and web search engines. Despite the different information needs for web archives and web search engine users, the search patterns for web archives had shown adoption of web search engine technologies. They found that most web archive users conducted short sessions. In our study, the sessions that are composed of one request contribute the most to the number of sessions. One important finding from analyzing the search interactions of the PWA logs is that the users prefer older documents. This is in contrast to what we found, that web archive users have significant repetitions for requests in 2011 (the year prior to our sample).

The challenge that faces web usage mining is detecting the robots who camouflage their identity and pretend to be humans. The robot detection problem has been examined in several studies [27, 9, 15, 11]. Doran et al. characterized robot detection techniques into four categories: syntactical log analysis, traffic pattern analysis, analytical learning techniques, and Turing test systems [10]. We used syntactical log analysis (simple processing by finding the self-identified robots) and traffic pattern analysis (specifying features for contrasting robots with humans).

3. ABSTRACT MODELS FOR ACCESSING WEB ARCHIVES

Through our analysis, we discovered four major patterns for web archive access. We present the model for each pattern along with an example from the logs in Figures 1-4. Each example consists of three columns: the client IP, the access time, and the requested URI. The times annotating the transition arrows in Figure 2-4 represent the inter-request time in the given examples. Note that we use TM URI-R to denote a TimeMap in the figures. We use Memento terminology (URI-T, URI-M, and URI-R) in the definitions. We refer to the original resource for URI-T and URI-M with URI-R(URI-T) and URI-R(URI-M), respectively.

3.1 Pattern 1: Dip

Dip is the pattern where a user accesses only one URI. The request can be for a URI-T (Figure 1(a) and the first example) or a URI-M (Figure 1(b) and the second example).

$$Dip = \{\text{URI-X}_i \,|\, i = 1 \text{ and URI-X} \in \{\text{URI-T, URI-M}\}\}$$

3.2 Pattern 2: Slide

Slide is the pattern in which a user accesses the same URI-R at different Memento-Datetimes. In this pattern, the user requests a URI-R and walks through time browsing its different copies (Figure 2).

$$\begin{aligned} Slide &= \{\text{URI-X}_i \,|\, i > 1, \text{URI-X} \in \{\text{URI-T, URI-M}\} \\ &\quad\ \text{and URI-R(URI-X}_{i}) = \text{URI-R(URI-X}_{i-1})\} \end{aligned}$$

Navigation between different URI-Ms can be done in many ways, e.g., directly from URI-M$_1$ to URI-M$_2$ (URI-R@t_1 ⇒ URI-R@t_2) or from URI-M$_1$ to URI-M$_2$, but in the middle the user returns to the TM URI-R to choose between the available datetimes (URI-R@t_1 ⇒ URI-T ⇒ URI-R@t_2).

3.3 Pattern 3: Dive

Dive is when a user accesses different URI-Rs at nearly the same datetime. In this pattern, the user accesses one URI-R at a specific time, URI-R$_1$@t_0, then navigates to different hyperlink(s) of URI-R$_1$'s page (e.g., URI-R$_2$@t_0) and so on (Figure 3).

$$\begin{aligned} Dive &= \{\text{URI-X}_i \,|\, i > 1, \text{URI-X} \in \{\text{URI-T, URI-M}\} \\ &\quad\ \text{and URI-R(URI-M}_{i}) <> \text{URI-R(URI-M}_{i-1})\} \end{aligned}$$

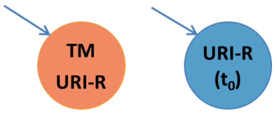
3.4 Pattern 4: Skim

Skim is when a user accesses a number of different TimeMaps for different URI-Rs (Figure 4). Skim does not include any access for mementos.

$$Skim = \{\text{URI-X}_i \,|\, i > 1 \text{ and URI-X} \in \{\text{URI-T}\}\}$$

4. METHODOLOGY

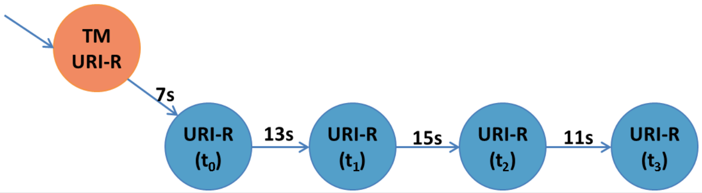
In this study, we introduce an analysis of the user access patterns of web archives. The analysis was conducted on



(a) URI-T (b) URI-M

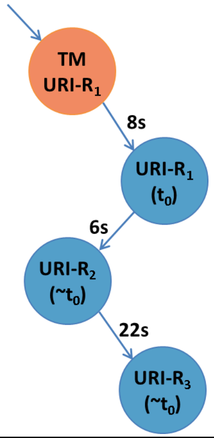
0.100.61.20	02/Feb/2012:06:48:24	http://wayback.archive.org/web/*/http://iyasizuku.com
0.1.134.90	02/Feb/2012:07:08:28	http://web.archive.org/web/19961022174810/http://altavista.com

Figure 1: Dip: A simple access to either a TimeMap or a memento.



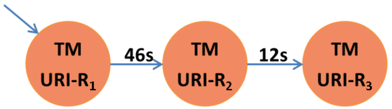
0.248.211.54	02/Feb/2012:07:04:52	http://wayback.archive.org/web/20000715000000*/http://google.com
0.248.211.54	02/Feb/2012:07:04:59	http://web.archive.org/web/20000301105534/http://google.com/
0.248.211.54	02/Feb/2012:07:05:12	http://web.archive.org/web/20051101145803/http://www.google.com
0.248.211.54	02/Feb/2012:07:05:27	http://web.archive.org/web/20080730200402/http://www.google.com/
0.248.211.54	02/Feb/2012:07:05:38	http://web.archive.org/web/20110215024256/http://www.google.com/

Figure 2: Slide: Accessing the same URI-R at different Memento-Datetimes.



0.106.160.155	02/Feb/2012:07:07:10	http://wayback.archive.org/web/*/http://my-ru.net
0.106.160.155	02/Feb/2012:07:07:18	http://web.archive.org/web/20100709124643/http://my-ru.net/
0.106.160.155	02/Feb/2012:07:07:24	http://web.archive.org/web/20100709124643/http://my-ru.net/home.php
0.106.160.155	02/Feb/2012:07:07:46	http://web.archive.org/web/20100706170736/http://my-ru.net/carousel.php

Figure 3: Dive: Browsing different URI-Rs at (approximately) the same Memento-Datetime.



0.10.212.177	02/Feb/2012:06:45:24	http://wayback.archive.org/web/*/laquadrature.net
0.10.212.177	02/Feb/2012:06:46:10	http://wayback.archive.org/web/*/parti-du-plaisir.com
0.10.212.177	02/Feb/2012:06:46:22	http://wayback.archive.org/web/*/humanite.fr

Figure 4: Skim: Traversing only TimeMaps for different URI-Rs.

```
0.247.222.86 - - [02/Feb/2012:07:03:46 +0000] "GET http://wayback.archive.org/web/*/http://www.aura.vu
HTTP/1.1" 200 96433 "http://www.archive.org/web/web.php" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8)
AppleWebKit/535.7 (KHTML, like Gecko) Chrome/16.0.912.77 Safari/535.7"

0.247.222.86 - - [02/Feb/2012:07:03:55 +0000]
"GET http://web.archive.org/web/20020404020224/http://www.aura.vu/ HTTP/1.1" 200 18875
"http://wayback.archive.org/web/*/http://www.aura.vu" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8)
AppleWebKit/535.7 (KHTML, like Gecko) Chrome/16.0.912.77 Safari/535.7"}
```

Figure 5: Sample of the Wayback Machine access log.

the Internet Archive's Wayback Machine access logs. The first step in preparing the Wayback access logs for usage mining was transforming the raw log file into server sessions through web-log preprocessing, which included data cleaning, user identification, and session identification [6]. Then, we performed feature extraction, robot detection, and user access pattern detection.

4.1 Wayback Machine Access Logs

A Web server log file is a plain text file that records the activity information of the submitted requests from the users on the web server. The Wayback Machine access logs contain the following fields: client IP, access time, HTTP request method (GET or HEAD), URI, protocol (HTTP), HTTP status code (200, 404, etc.), bytes sent, referring URI, and User-Agent. For privacy purposes, the Internet Archive anonymized the client IP address. A segment from the Wayback Machine server logs, which we will call Wayback access logs, is shown in Figure 5. The first line shows a request for a URI-T. The second line shows a request for a URI-M.

4.2 Dataset

The Wayback Machine allows users to browse archived copies of web pages across time. The Wayback access logs were sampled using two probability techniques [29]: cluster sampling, which is choosing a cluster of data randomly, and random sampling, where each sampling unit has an equal chance of being included. We performed cluster sampling by choosing a week (Feb. 2-8, 2012) and random sampling by taking a random slice from each day of the week. Each sample comprised a slice of 2M requests to the Wayback Machine web server. Table 1 shows the characteristics of each dataset. It contains the following features for each sample:

- **Duration:** the difference between the last request time and the first request time of each sample in HH:MM:SS format.
- **GET:** the percentage of requests that used the GET method.
- **Embedded:** the percentage of requests that were for embedded resources of web pages (such as images and CSS files, etc.).
- **SI Robots:** the percentage of requests by self-identified robots based on the User-Agent field.
- **NullRef:** the percentage of requests that had an empty referral field.
- **s2xx:** the percentage of successful requests (2xx status code).
- **s3xx:** the percentage of redirections (3xx status code).
- **s4xx:** the percentage of client errors (4xx status code).
- **s5xx:**: the percentage of server errors (5xx status code).
- **Cleaned:** the percentage of requests remaining after removing requests for embedded resources, HEAD requests, and requests that resulted in status codes other than 200, 404, or 503.
- **Sessions:** the number of sessions.

The last three columns of the table show the mean, standard deviation, and corresponding standard error between the samples. We use the Feb. 2, 2012 sample in our analysis because as we see from Table 1, it is a representative sample.

In the Feb. 2 sample, we note that HTTP 3xx accounts for 52% of the total number of requests. This is related to the default Wayback Machine behavior. First, the Wayback Machine rewrites all of the hyperlinks of a memento's embedded resources with the mementos's timestamp. Second, in the resolution of these URIs, the Wayback Machine will redirect the request of the embedded resources and hyperlinks to the nearest (timestamp) available memento. Furthermore, the Wayback Machine responds with a 302 status first when the requested URI-R is not available on the Wayback Machine, and then responds with a 404 status.

4.3 Data Cleaning

The first step in preprocessing our dataset was data cleaning, i.e., removing log entries that were not needed for the mining process [17, 28]. In similar studies for log analysis, robots that identify themselves in the User-Agent field were removed. Because robots crawl web archives intentionally, we did not exclude their requests in the cleaning step. We eliminated the following items which were irrelevant in terms of user behavior:

- Requests that were generated automatically by the web browser for embedded resources of the requested web page (such as graphic files, page style files, etc.).
- Entries with an HTTP status code other than HTTP 200, 404, or 503. We kept only these because we considered them to be requests executed by the user.
- Requests using the HEAD request method (as suggested by [16]).
- Static resources of the Internet Archive web site and the URIs of the liveweb service, which the Internet Archive introduces to redirect the user to the live web when the copy is not found on the Wayback Machine.
- Invalid requests from web sites which included a link for malformed URI-Rs (for example, about:blank) among their embedded resources, so that each request on their web sites caused automatic requests to the Wayback Machine server. Similar behavior had been detected by Omodei [21].

Days	Feb 2	Feb 3	Feb 4	Feb 5	Feb 6	Feb 7	Feb 8	Mean	SD	SE
Duration	0:33:12	0:31:15	0:40:34	0:42:57	0:29:35	0:25:45	0:24:33	0:32:33	0:06:29	0:02:27
GET	98.4%	99.3%	97.7%	97.9%	99.4%	99.7%	99.8%	99%	0.8%	0.3%
Embedded	47.4%	34.8%	43.7%	42.7%	41.9%	44.7%	46.8%	43.1%	3.9%	1.5%
SI Robots	6.2%	12.0%	7.7%	7.7%	2.9%	3.5%	3.8%	6.3%	3.0%	1.1%
NullRef	42.6%	56.6%	47.5%	47.0%	49.4%	42.6%	43.9%	47.1%	4.6%	1.7%
s2xx	33.7%	32.4%	34.2%	33.2%	34.1%	33.4%	33.6%	33.5%	0.6%	0.2%
s3xx	51.8%	52.3%	50.8%	52.2%	51.7%	51.9%	53.2%	52.0%	0.7%	0.3%
s4xx	11.7%	13.1%	12.0%	11.6%	11.2%	10.3%	10.1%	11.4%	0.9%	0.4%
s5xx	2.8%	2.3%	3.0%	2.9%	3.0%	4.4%	3.1%	3.1%	0.6%	0.2%
Cleaned	21.3%	23.0%	17.6%	17.7%	20.7%	18.1%	16.9%	19.3%	2.2%	0.8%
Sessions	37,634	31,731	32,159	28,750	36,087	35,848	32,117	33,475	2,896	1,094

Table 1: Features for each sample of 2M records, Feb. 2-8, 2012.

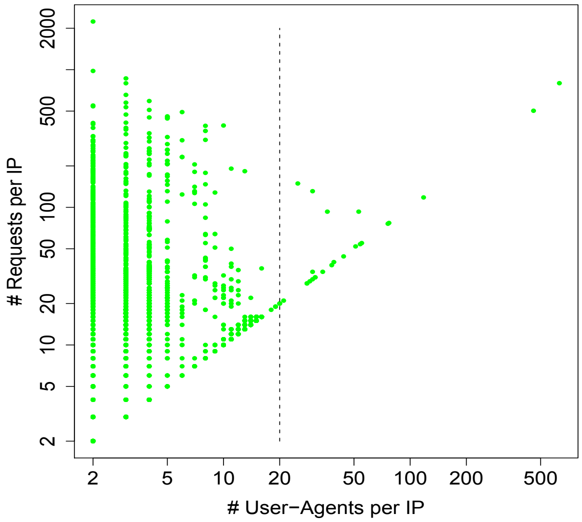


Figure 6: The number of user requests per IP against the number of User-Agents per IP.

4.4 User Identification

To identify the users, the log files were first sorted by the IP, then by the request time. At first, we identified users by the 2-tuple (IP, User-Agent), but we found instances of malicious robots who not only did not self-identify, but who also changed their User-Agent with every request. There are some legitimate cases for humans to have different User-Agents and the same IP, for instance, two simultaneous users coming from behind a firewall. So, we needed to determine a reasonable threshold for the number of User-Agents per IP to detect malicious robots.

Figure 6 shows the relationship between the number of User-Agents per IP and the number of requests per IP (for those IPs with more than one User-Agent). We excluded self-identified robots from the graph to avoid biasing the results. We found only 24 users who had changed their User-Agent field more than 20 times. The median value for the number of User-Agents per IP for the dataset is 3 (excluding users who had one User-Agent only). We concluded that 20 different User-Agents per IP is a good threshold for this dataset. For IPs with at most 20 different User-Agents, we used the 2-tuple (IP, User-Agent) to identify individual users. The 24 IPs with more than 20 different User-Agents were classified as 24 separate robots.

4.5 Session Identification

A session is the set of web pages that are requested by particular user [17]. Session identification is performed by dividing a web server log file into web server sessions. First, we group all the requests based on the IP and User-Agent (as described in Section 4.4). Second, we apply a threshold timeout, so that if the time elapsed between two consecutive requests is longer than this threshold, the second request is considered to be the first request of the new session. There have been several suggested timeout thresholds including 25.5 minutes [5], 30 minutes [27, 14], and 60 minutes [2]. Others proposed 10 minutes as a conservative threshold to capture the time for staying on one page [16, 24]. In our study, we divided the requests of each user into individual sessions based on a 10 minute timeout threshold. Future research is required to verify that searching and browsing models for the current web are valid for browsing the past web (i.e., web archives).

After identifying the sessions, we extracted features for each session to be used further in the analysis. A session, S, is 7-tuple.

$$S = \langle URI, S_l, S_d, BS, \bar{S}_{rt}, stdev(S_{rt}), IH \rangle$$

The following is the description of each item:

- URI is the set of URIs that the user visited in the session. The set of URIs are defined as:

$$\begin{aligned} URI \quad = \quad & \{\text{URI}_i \mid i \text{ is an integer, } 1 \leq i \leq S_l \\ & \text{and URI} \in \{\text{URI-T, URI-M}\}\} \end{aligned}$$

- S_l, session length, is the number of webpages the user requested during the session.
- S_d, session duration, is calculated by subtracting the timestamp of the first request of the session from the timestamp of the last request of the session.
- BS is the browsing speed of each session in requests/second. $BS = S_l/S_d$.
- $\bar{S}_{rt}$ is the mean inter-request time of the session.
- $stdev(S_{rt})$ is the standard deviation of the inter-request time of the session.
- IH, image-to-HTML, is the ratio between the number of image files and the number of HTML files per session.

4.6 Robot Detection

Because of the increasing numbers of web crawlers that are engaged in web harvesting, many studies have been con-

	# Detected Robots	
Heuristics	**out of 37,634 sessions**	**out of 426,317 requests**
SI Robots	1,410	68,967
#UA per IP	24	2,747
Robots.txt	55	90
Browsing Speed	1,601	47,320
Image-to-HTML	33,244	326,019
Total Robots	**34,203 (90.9%)**	**396,627 (93.9%)**

Table 2: The number of detected robots from applying each heuristic independently and the number of the records after applying all the filters together.

Filters	% Excluded Requests (out of 2M)
Status Code	51.8%
Embedded Resources	47.4%
Static and Liveweb	10.0%
Invalid Requests	3.7%
HEAD	1.6%
All Filters	**78.7%**

Table 3: The characteristics of data cleaning filters. Some of requests fell into multiple categories, so the percentages add up to more than 100%.

ducted for investigating the robot detection problem [27, 15]. In this study, we used different types of robot detection techniques [10]. First, we applied syntactical log analysis by checking the User-Agent field to identify the self-identified robots. Second, we applied traffic pattern analysis techniques to distinguish humans from robots based on their navigational behavior. In this section, we describe heuristics we used for distinguishing robots from humans.

4.6.1 User-Agent Check

The User-Agent check is applied for requests from crawlers and robots which declared their identity to the web server through the User-Agent field (SI robots). We excluded these robots by applying this heuristic before the calculations of the session features to avoid biasing the results.

4.6.2 Number of User-Agent per IP

As explained earlier (Section 4.4), we used 20 as a threshold for the maximum number of different User-Agents for each IP. Users with more than 20 different User-Agents were classified as robots.

4.6.3 Robots.txt file

Web site administrators put a list of access restrictions to specify which parts of their web site are not allowed to be visited by robots. We labeled the sessions in which users downloaded the robots.txt file for the Wayback Machine (http://web.archive.org/robots.txt) as robots.

4.6.4 Browsing Speed (BS)

The importance and the effect of BS has been discussed and used for detecting robots many times [20, 28, 22]. We use $BS \leq 0.5$ (i.e., no faster than one request every two seconds) as a threshold for human browsing speed [4]. We observed that this threshold is appropriate for our dataset, so we classify the sessions with $BS > 0.5$ as robots.

4.6.5 Image-to-HTML Ratio (IH)

Human sessions should have more images than robot sessions because of the embedded images present in most HTML pages. Robots tend to retrieve only HTML pages, while ignoring image formats. We used the IH metric calculated previously for each session to detect robots. In [26], 1:10 IH had been suggested as a good threshold for distinguishing robots from humans. We label a session with less than one image file for every 10 HTML files as a robot. IH is the only heuristic that does not require a session have at least two requests. This heuristic is the best predictor for robots [27, 26], and it has a strong effect on our dataset.

5. RESULTS AND ANALYSIS

In this section, we explain the results of preprocessing the dataset (described in Sections 4.3-4.5) and of applying the heuristics for robot detection (described in Section 4.6). We analyze the resulting data and contrast the behavior and access patterns of humans and robots. We conclude with an analysis of the temporal preference of human users.

5.1 Traffic Analysis

To extract the user access patterns for web archives from the Wayback access logs, we first applied data preprocessing techniques (data cleaning, user identification, session identification) to convert the log file into web server sessions. The raw log file contains 2M requests from which we determined 21,932 unique IPs. Because of the stateless nature of the log files, we identified the users based on the IP and User-Agent to identify 33,841 users who created 37,634 different sessions.

The characteristics of each filter (3xx status code, embedded resources, static resources and liveweb, invalid requests, HEAD) and the total number of excluded requests after applying all the filters together are shown in Table 3. The number of records in the Feb. 2 sample was decreased from 2M to 426,317 (21.3% of the requests in the raw file).

5.2 Robots vs Humans

Table 2 contains the results of applying the heuristics for detecting robots. The rules are not mutually exclusive, but we calculated the number of requests which had been labeled as robots from each filter separately. IH had the largest effect on detecting robots. We used a 1:10 IH as a threshold for distinguishing robots from humans. We found that 99.93% of the sessions which were detected by this heuristic had 0 images. BS is also important, because it classified a significant number of robots who had a BS (more than 0.5 requests/second) impossible for humans.

Users	# Requests (Filtered)	# Requests (Raw)	# Sessions	# Transferred MB	# URI-Ts	# URI-Ms
Robots	396,627 (93.0%)	1,002,573 (50.1%)	34,203 (90.9%)	20,010	378,201 (95.4%)	18,426 (4.6%)
Humans	29,690 (7.0%)	810,049 (40.5%)	3,431 (9.1%)	4,459	5,505 (18.5%)	24,185 (81.5%)

Table 4: HTTP activity of robots and humans.

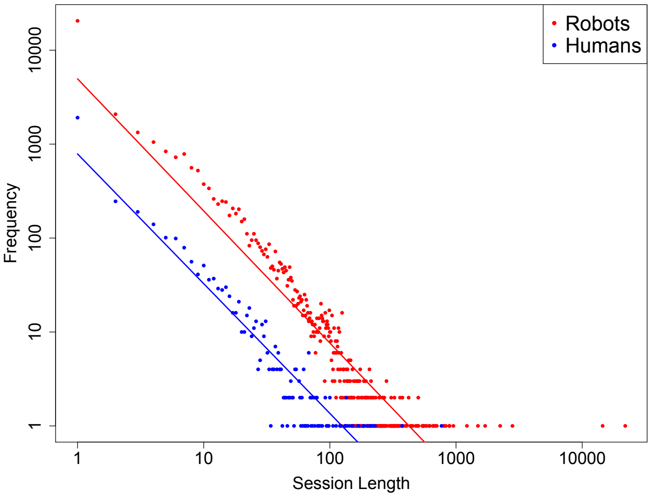


Figure 7: The frequency of session lengths (# of requests) for humans and robots.

Table 4 contains the summary of the activity of humans and robots. From the table, we notice that the sum of the percentage of raw requests from humans and robots did not equal 2M requests. The reason is that there are many accesses that were created by invalid requests to the web server. Furthermore, there are many requests to embedded resources only, which were filtered. The percentage of human requests after cleaning and separating robot requests is only 1.5% of the 2M requests.

The significant discovery here is the 10:1 ratio of robot sessions to human sessions. This ratio is a strong motivation for building an API interface that serves robot accesses in order to decrease the load of robots on the Wayback Machine. A typical human session costs more than a robot session as humans average 1.30 MB/session and robots average 0.58 MB/session. Human requests include automatic downloads of the embedded resources of the web pages they access, and robots usually ignore downloading these embedded resources.

We discovered that most of the robots had a breadth search strategy in downloading the web pages from the Wayback Machine; more than 95% of the robots downloaded TimeMaps only, as shown in Table 4. On the other hand, of all human requests, only 18.5% were for TimeMaps.

5.2.1 *Session Length* (S_l)

After detecting the robots, we separated them from humans and analyzed their behaviors individually. Figure 7 shows the session length frequency for robots in red and for humans in blue. We notice from the figure that many more robots have longer sessions than humans. The $\bar{S}_l$ for robots is 10 requests/session, while humans have an $\bar{S}_l$ of 9 requests/session.

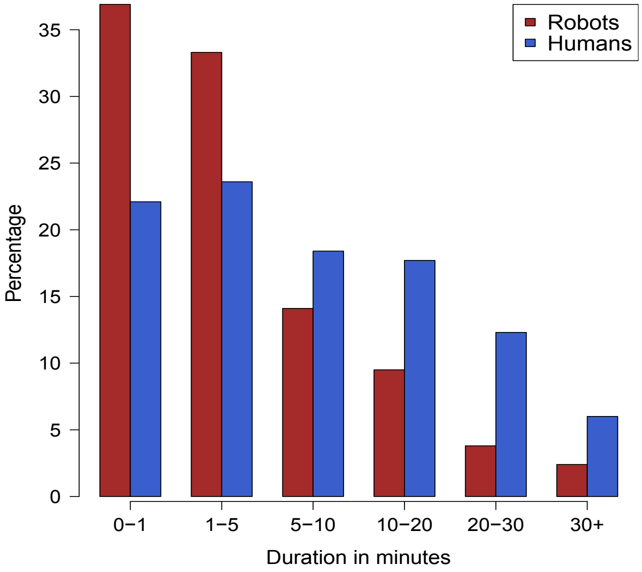


Figure 8: The percentage of sessions for each interval. The number of humans and robots sessions are 2024 and 17019 (excluding the sessions with one request), respectively.

5.2.2 *Session Duration* (S_d)

We computed session duration by subtracting the time of the first request from the time of the last request for each session. Session duration requires at least two requests. Figure 8 shows the percentage of sessions with different session durations for robots and humans. We can see that the majority of the sessions were short, taking into consideration that we did not count the time spent on the last requested web page. The $\bar{S}_d$ is 10 minutes for robots and 5 minutes for humans.

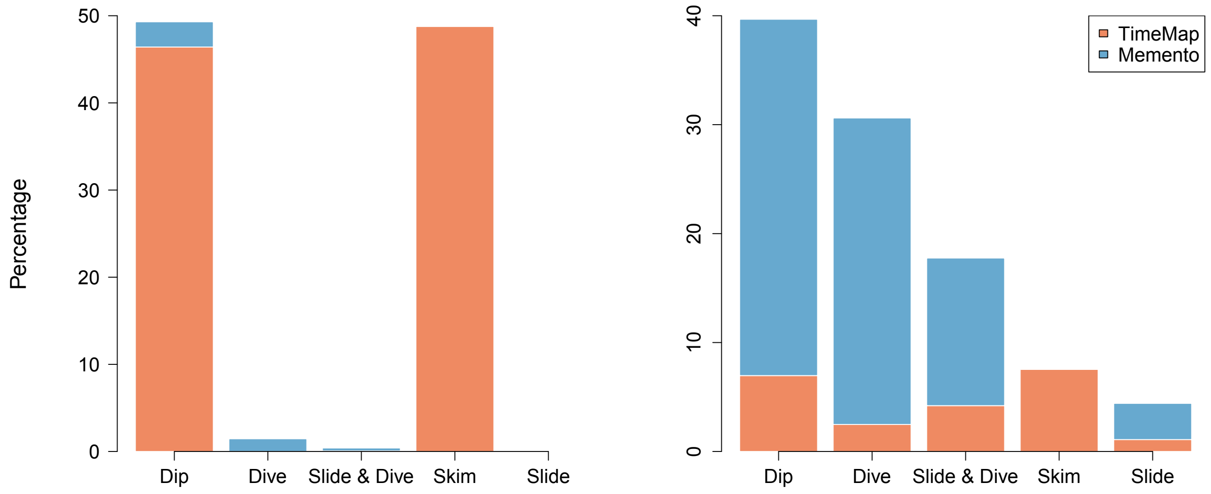
5.2.3 *Inter-Request Time*

We also calculated $\bar{S}_{rt}$ and $stdev(S_{rt})$ for each session. We found that the median values of $\bar{S}_{rt}$ for human and robot sessions are 19 seconds and 40 seconds, respectively. The median of $stdev(S_{rt})$ values is 37 seconds for robots and 11 seconds for humans. This indicates that robots tend to have more irregular periods between HTML requests than humans, and this matches the finding by Tan et al. [27].

5.3 Web Archive User Access Patterns

How do users go through web archives? Do they go in deeply from URI-R$_1$ to URI-R$_2$, do they browse broadly from URI-M$_1$ to URI-M$_2$ for the same URI-R, or do they use a combination of these two patterns? Are robot accesses similar to human accesses?

In this section, we answer the previous questions by extracting the user access patterns for web archives from our



(a) Robots (34203 sessions)

(b) Humans (3431 sessions)

Figure 9: Robots and humans exhibit different access patterns.

filtered dataset. The requested URIs for each session were extracted and then identified based on their type, URI-M or URI-T. We also extracted the URI-R of each requested URI to compare it with the other URI-Rs from the same session. Because of the existence of different forms of URIs which refer to the same website [18], we applied URI canonicalization for the URI-Rs to normalize them under one host [8].

We discovered four basic building blocks (Dip, Slide, Dive, Skim) of the user access patterns for web archives. Figures 1-4 show the models along with examples of the four patterns. The percentages of each pattern exhibited in robot and human sessions are shown in Figures 9(a) and 9(b) along with the percentages of requests to TimeMaps and mementos for each pattern.

Dip

Dip is the pattern where the user requests a single URI. This URI can be a URI-M or a URI-T. It represents the most repeated pattern for humans (33% of all sessions) and robots (49% of all sessions). URI-Ms contribute to 83% of human sessions that exhibit the Dip pattern, although 94% of the robot Dips are requests for URI-Ts.

Slide

The user who is interested in travelling through time, browsing different copies of the same URI-R, creates the Slide pattern. There are only a few humans who access the web archives broadly then navigate away (4.2% of all sessions). Robot sessions do not have this pattern with a noticeable percentage (0.1% of all sessions).

Dive

Dive represents the second highest percentage of human sessions, 29.7%. In this pattern, the user goes deeply for browsing hyperlinks of URI-Ms. The robot sessions which were composed of this pattern crawl the web sites deeply, but they are not a significant number of sessions.

Skim

Skim is the pattern for which the users access different numbers of TimeMaps. Robot sessions exhibit this pattern 48.7% of the time. Investigating the relationship between the topics of the URI-Rs of the requested TimeMaps during a single session is one of our goals for upcoming research.

User	Pattern	Median	Mean	SD
Robots	Slide	3	3	1.4
	Dive	3	15	53.2
	Skim	3	21	267.0
Humans	Slide	3	4	3.4
	Dive	4	8	14.3
	Skim	3	6	7.2

Table 5: Statistics for the length of all Slides, Dives, and Skims

Slide and Dive

A large number of human sessions consist of at least two occurrences of the Dive and Slide patterns. In these sessions, the users request URI-R$_1$ and browse its different copies at different times (URI-R$_1$@t_1 ⇒ URI-R$_1$@t_2 ⇒ URI-R$_1$@t_3), then dive through a hyperlink (URI-R$_2$@t_3) from URI-R$_1$@t_3, then repeat Dive or Slide. In contrast, users may start by going deeply through different mementos for different URI-Rs (Dive pattern), then go broadly through one of these mementos to browse other captures at different times (Slide pattern) (e.g., URI-R$_1$@t_1 ⇒ URI-R$_2$@t_1 ⇒ URI-R$_3$@t_1 ⇒ URI-R$_3$@t_2, etc.). The percentage of human sessions that were composed of a combination of these two patterns is 17.2%. We calculated the number of Slides

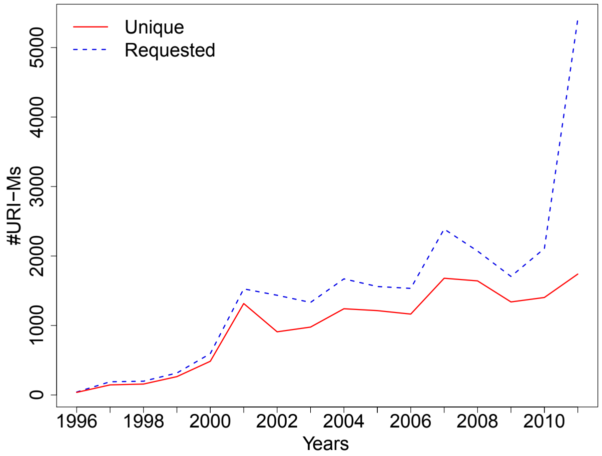


Figure 10: Distributions of the years for the unique and requested mementos by humans.

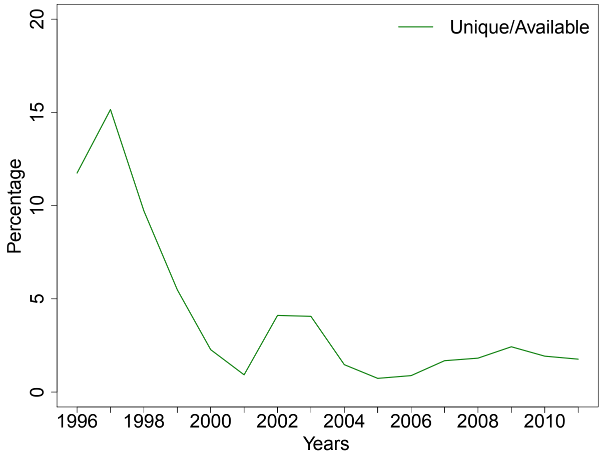


Figure 11: The proportion of unique URI-Ms requested out of the potential requested for each year.

and Dives for these sessions and found 1167 Slides and 1942 Dives. For robot sessions that were composed of Slide and Dive, we found 328 Slides and 571 Dives.

Pattern Length

Each pattern is made up of a number of requests, which we call the pattern length. We calculated the pattern length for all sessions. The median, mean, and standard deviation of the lengths of each pattern for robots and humans are summarized in Table 5. For humans, the longest mean pattern length is an 8-request Dive, while for robots the longest mean pattern length is 21 requests in a Skim.

5.4 Temporal Analysis

Figure 10 shows both the unique and total number of mementos referenced grouped by the year of their Memento-Datetime. Although there is no clear temporal preference for any one year of the unique mementos, there were a significant number of repeated requests for mementos from 2011. This locality of reference suggests that there is an important benefit to be gained by caching the mementos from the recent past. Figure 11 shows that the total number of mementos available for 2011 was similar to previous years. In both Figures 11 and 10, pre-2001 data is included although in those years the archives are too sparse for meaningful comparison with later years.

6. CONCLUSIONS AND FUTURE WORK

We introduced the basic building blocks (Dip, Slide, Dive, and Skim) for user access patterns for web archives through an analysis of the Internet Archive's Wayback Machine access logs. We applied heuristics for detecting robots and found that robot sessions outnumber human sessions 10:1. This suggests that there is utility in building an API interface that serves robot accesses. Robots account for 91% of sessions and 93% of requests to the Wayback Machine, yet robots outnumber humans 5:4 only in terms of raw, unfiltered requests and 4:1 in terms of megabytes transferred. We found that humans download more information per session due to embedded resources, which robots ignore. We also analyzed human and robot access patterns to emphasize the similarities and the differences between them. We found that robots mainly exhibit the Dip and Skim patterns, with about 49% of their sessions for each pattern, and that they access TimeMaps almost exclusively. Humans exhibit the Dip and Dive patterns the most with 39% and 30% of their sessions, respectively. Unlike robots, humans mainly access archived pages rather than TimeMaps. Finally, we provide an analysis for the temporal preferences of humans based on the Memento-Datetime (by year) of their requests and discovered significant repetitions for requests in 2011. This suggests that there is a benefit to be gained by caching mementos from the recent past.

Web server logs are a rich source for information about web archives. We are planning to extend our analysis to serve other applications of web usage mining, such as personalization for making dynamic recommendations to web archive users based on their navigational behavior patterns by using data mining techniques. Further, we will study the validity of applying searching and browsing models for the current web to web archives. We also expect that Slides and Dives that users create on web archives may create stories around a particular event. We plan to extend our study on a large data set to detect stories that humans might create from their access patterns, which will be integrated into the live web to produce automatic stories about specific event for the users.

7. ACKNOWLEDGMENTS

This work was supported in part by the NSF (IIS 1009392) and the Library of Congress. We thank Kris Carpenter Negulescu (Internet Archive) for access to the anonymized Wayback Machine logs.

8. REFERENCES

[1] I. Adams, E. L. Miller, and M. W. Storer. Analysis of workload behavior in scientific and historical long-term data repositories. Technical Report UCSC-SSRC-11-01, University of California, Santa Cruz, Mar. 2011.

[2] P. Anick. Using terminological feedback for web search refinement: a log-based study. In *Proceedings of the 26th annual international ACM SIGIR conference on*

Research and development in informaion retrieval, SIGIR '03, pages 88–95. ACM, 2003.

[3] T. T. Aye. Web log cleaning for mining of web usage patterns. In *3rd International Conference on Computer Research and Development*, ICCRD, pages 490–494. IEEE, Mar. 2011.

[4] G. Castellano, A. M. Fanelli, and M. A. Torsello. LODAP: a log data preprocessor for mining web browsing patterns. In *Proceedings of the 6th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, volume 6 of *AIKED '07*, pages 12–17, Feb. 2007.

[5] L. D. Catledge and J. E. Pitkow. Characterizing browsing strategies in the World-Wide web. *Computer Networks and ISDN Systems*, 27(6):1065–1073, Apr. 1995.

[6] R. Cooley and B. Mobasher. Data Preparation for Mining World Wide Web Browsing Patterns. *Knowledge and Information Systems*, 1:5–32, 1999.

[7] M. Costa and M. J. Silva. Characterizing Search Behavior in Web Archives. In *Proceedings of Temporal Web Analytics Workshop*, TWAW, 2011.

[8] M. Cutts. SEO advice: URL canonicalization. `http://www.mattcutts.com/blog/seo-advice-url-canonicalization/`, Jan. 2006.

[9] M. D. Dikaiakos, A. Stassopoulou, and L. Papageorgiou. An investigation of web crawler behavior: characterization and metrics. *Computer Communications*, 28(8):880–897, May 2005.

[10] D. Doran and S. S. Gokhale. Web robot detection techniques: overview and limitations. *Data Mining and Knowledge Discovery*, 22(1-2):183–210, June 2010.

[11] W. Guo, Y. Zhong, and J. Xie. A Web Crawler Detection Algorithm Based on Web Page Member List. In *Proceedings of 4th International Conference on Intelligent Human-Machine Systems and Cybernetics*, pages 189–192, Aug. 2012.

[12] B. J. Jansen and A. Spink. How are we searching the World Wide Web? A comparison of nine search engine transaction logs. *Information Processing & Management*, 42(1):248–263, Jan 2006.

[13] B. Kahle. Wayback Machine: Now with 240,000,000,000 URLs. `http://blog.archive.org/2013/01/09/updated-wayback/`, Jan. 2013.

[14] R. Kumar and A. Tomkins. A characterization of online browsing behavior. In *Proceedings of the 19th international World Wide Web conference*, WWW '10, pages 561–570. ACM, 2010.

[15] S. Kwon, M. Oh, D. Kim, J. Lee, Y.-G. Kim, and S. Cha. Web Robot Detection based on Monotonous Behavior. In *Proceedings of the Information Science and Industrial Applications*, volume 4, 2012.

[16] H. Liu and V. Kešelj. Combined mining of web server logs and web contents for classifying user navigation patterns and predicting users' future requests. *Data Knowledge Engineer*, 61(2):304–330, May 2007.

[17] Z. Markov and D. T. Larose. *Data Mining the Web: Uncovering Patterns in Web Content, Structure, and Usage*. John Wiley & Sons, Inc., 2007.

[18] F. McCown and M. L. Nelson. Evaluation of crawling policies for a web-repository crawler. In *Proceedings of the 17th conference on Hypertext and hypermedia*, HT '06, pages 157–168, Aug. 2006.

[19] K. C. Negulescu. Web Archiving @ the Internet Archive. Presentation at the 2010 Digital Preservation Partners Meeting, `http://1.usa.gov/XSjDG8`, 2010.

[20] P. Nithya and P. Sumathi. Novel Pre-Processing Technique for Web Log Mining by Removing Global Noise, Cookies and Web Robots. *International Journal of Computer Applications*, 53(17):1–6, 2012.

[21] M. Omodei. Trends in Use of Pandora Archive. International Internet Preservation Consortium, `http://bit.ly/11jtwi2`, 2012.

[22] K. S. Reddy, G. P. S. Varma, and I. R. Babu. Preprocessing the web server logs: an illustrative approach for effective usage mining. *ACM SIGSOFT Software Engineering Notes*, 37(3):1–5, May 2012.

[23] D. S. Sisodia and S. Verma. Web usage pattern analysis through web logs: A review. In *Proceedings of 9th International Conference on Computer Science and Software Engineering*, JCSSE, pages 49–53, May 2012.

[24] M. Spiliopoulou, B. Mobasher, B. Berendt, and M. Nakagawa. A Framework for the Evaluation of Session Reconstruction Heuristics in Web-Usage Analysis. *INFORMS Journal on Computing*, 15(2):171–190, Apr. 2003.

[25] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web usage mining: discovery and applications of usage patterns from Web data. *ACM SIGKDD Explorations Newsletter*, 1(2):12, Jan. 2000.

[26] A. Stassopoulou and M. D. Dikaiakos. Web robot detection: A probabilistic reasoning approach. *Computer Networks*, 53(3):265–278, Feb. 2009.

[27] P.-N. Tan and V. Kumar. Discovery of Web Robot Sessions Based on their Navigational Patterns. *Data Mining and Knowledge Discovery*, 6(1):9–35, Jan. 2002.

[28] D. Tanasa and B. Trousse. Advanced data preprocessing for intersites Web usage mining. *IEEE Intelligent Systems*, 19(2):59–65, Mar. 2004.

[29] C. Teddlie and F. Yu. Mixed Methods Sampling: A Typology With Examples. *Journal of Mixed Methods Research*, 1(1):77–100, Jan. 2007.

[30] B. Tofel. Wayback for Accessing Web Archives. In *Proceedings of International Web Archiving Workshop*, IWAW, 2007.

[31] M. Tsagkias and R. Blanco. Language intent models for inferring user browsing behavior. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '12, pages 335–344. ACM, 2012.

[32] H. Van de Sompel, M. L. Nelson, and R. Sanderson. HTTP framework for time-based access to resource states – Memento. `https://datatracker.ietf.org/doc/draft-vandesompel-memento/`, 2012.

[33] H. Van de Sompel, M. L. Nelson, R. Sanderson, L. L. Balakireva, S. Ainsworth, and H. Shankar. Memento: Time Travel for the Web. Technical Report arXiv:0911.1112, 2009.