Factors Affecting Website Reconstruction from the Web Infrastructure

Frank McCown
Old Dominion University
Computer Science Department
Norfolk, Virginia, USA 23529
fmccown@cs.odu.edu

Norou Diawara
Old Dominion University
Department of Mathematics and Statistics
Norfolk, Virginia, USA 23529
ndiawara@odu.edu

Michael L. Nelson
Old Dominion University
Computer Science Department
Norfolk, Virginia, USA 23529
mln@cs.odu.edu

ABSTRACT

When a website is suddenly lost without a backup, it may be reconstituted by probing web archives and search engine caches for missing content. In this paper we describe an experiment where we crawled and reconstructed 300 randomly selected websites on a weekly basis for 14 weeks. The reconstructions were performed using our web-repository crawler named Warrick which recovers missing resources from the Web Infrastructure (WI), the collective preservation effort of web archives and search engine caches. We examine several characteristics of the websites over time including birth rate, decay and age of resources. We evaluate the reconstructions when compared to the crawled sites and develop a statistical model for predicting reconstruction success from the WI. On average, we were able to recover 61% of each website's resources. We found that Google's PageRank, number of hops and resource age were the three most significant factors in determining if a resource would be recovered from the WI.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-based services*; H.3.7 [**Information Storage and Retrieval**]: Digital Libraries—*Collection*

General Terms

Design, Experimentation, Measurement

Keywords

digital preservation, search engine caches, web archiving

1. INTRODUCTION

The size and prevalence of the Web today is indicative of how easy it is to produce web content. Businesses, governments, organizations and individuals alike publish incredible amounts of data on the Web every day. At the same time, web pages and websites disappear almost as quickly as they come on-line. Many web pages or entire websites go missing over time for a variety reasons: they are moved to different locations; abandoned due to lack of interest, relevance, time or money; or lost when a hard drive crash, virus, fire, death or other catastrophic event occurs and no backup for the site can be found.

Recognizing the ephemeral and dynamic nature of the Web, the Internet Archive (IA) has set out to archive as much of the Web as possible for future generations. Individuals who have lost their websites are often relieved to find at least some of their lost content [22] in the IA or in the caches of search engines like Google [19].

To automate the process of recovering lost websites, we have built a web-repository crawler named Warrick which searches for missing resources in the vaults of several web repositories (IA, Google, MSN and Yahoo) [28]. Warrick works much like a typical web crawler except it downloads pages directly from web repositories rather than the Web. Warrick has been made freely available on the Web and used by individuals and third parties to recover lost websites. For example, Warrick was used to recover former Congressman Mark Foley's websites when they were shut-down after his resignation in September 2006 [23]. Warrick was also used to recover the WWW'06 conference website when a fire destroyed the building housing the web server [25].

In this paper, we describe an experiment using 300 randomly sampled websites from `dmoz.org`. Once a week for 14 weeks we crawled each website and reconstructed it with Warrick. We compare the crawled sites with our reconstructions to evaluate how much of the websites could have been recovered had the websites been taken off-line or lost. We examine a number of characteristics from our weekly snapshots of these 300 websites such as birth rate, decay and rate of change, and we ran a regression analysis to explain which characteristics best determine the recoverability of resources from our websites. From our analysis we have constructed a predictive model which can be used to determine how much of a website could be recovered if it were lost today.

2. BACKGROUND AND RELATED WORK

Recent measurements [4, 16] showing low overlap of search engine indexes testify to the fact that the Web is too large for any single search engine to index it all. And although a noble effort, the Internet Archive cannot hope to archive the entire Web. But when used collectively, search engines (using their caches) and web archives can save a very large

JCDL'07, June 17–22, 2007, Vancouver, British Columbia, Canada.

portion of the Web. We call this collective preservation effort the Web Infrastructure (WI).

Search engines and web archives most commonly discover web resources using traditional web crawling. More recently, new methods of discovery have been adopted by search engines and web archives in an effort to increase the coverage and quality of their holdings. Using the Google Sitemap Protocol [14] and Yahoo Site Explorer [39], webmasters can expose previously uncrawlable (deep web) content to the search engine's crawlers. MSN, Google and Yahoo have also begun to support resource discovery using OAI-PMH directly or indirectly [24, 31]. IA is able to discover some resources from users who have the Alexa toolbar [2] installed on their browsers. All of these efforts increase the preservation capacity of the WI.

Web resources are stored in the WI in canonical and non-canonical formats. In an attempt to capture the Web just as it existed when crawled, IA archives all resources in their canonical format. All three search engines also store HTML resources in their canonical format (with a few minor exceptions by Yahoo), but other textual resources like PDF, PostScript and Microsoft Office documents (Word, Excel, PowerPoint) are converted and cached as HTML. Images are cached as thumbnails in compliance with copyright law [33], and many resources like JavaScript, style sheets, Flash files, etc. are not cached at all.

We first explored reconstructing websites from the WI in [28] where we reconstructed 24 websites with our first incarnation of Warrick. HTML resources were the most successfully recovered format; for several websites we were able to recover 100% of the HTML resources. We also examined the caching behavior of search engines on four decaying web collections. The caching experiment uncovered very different crawling and caching policies of Google, MSN and Yahoo. Google was the most successful at caching our web collections and in some cases kept numerous resources cached months after they had been deleted from our web server.

In [25], we examined challenges to website reconstruction presented by the different URL canonicalization policies of web repositories. We showed how lister queries (queries that initially ask a web repository to list all the URLs they have stored) minimized many of the URL canonicalization problems. Lister queries were also used for producing three different crawling policies which were evaluated for efficiency.

Growth, change and decay patterns in the Web have been researched for over a decade [5, 11, 12, 32]. A number of studies have examined link rot in regards to general web pages (e.g., [3, 20]), academic citations (e.g., [21]) and digital libraries (e.g., [30]), and several studies have examined finding replacements for missing web pages [10, 17]. Our study adds to this body of work by tracking the growth, change and decay of 300 'typical' websites over 14 weeks and examining what and how much can be recovered from the WI if one of these websites was suddenly lost. Our work addresses the question, How much preservation can be had for free if we were to do nothing to protect our website from loss?

3. QUANTIFYING RECONSTRUCTIONS

We have perviously defined a **reconstructed website** to be the collection of recovered resources that share the same URIs as the resources from a lost website or from some previous version of the lost website [25, 28]. For websites composed of static files, recovered resources are equivalent to the files that were lost. For sites produced dynamically using CGI, PHP, etc., the recovered resources would match the client's view of the resources and would be useful to the webmaster in rebuilding the server-side components.

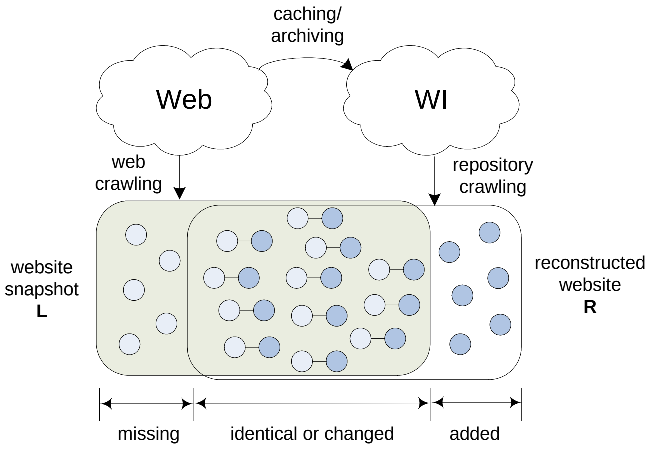

Figure 1: Comparing a crawled website to its reconstruction.

To quantify the difference between a lost website (L) and a reconstructed website (R), we first classify all resources from L and R that are uniquely identified by a URI. For each resource l_i in L, we examine its corresponding resource r_i in R that shares the same URI and categorize it as *identical* (r_i is byte-for-byte identical to l_i) or *changed* (r_i is not identical to l_i). We categorize all resources in L that do not share a URI with any resource in R as *missing* and those resources in R that do not share a URI with any resource in L as *added*. We use the four classifications to assign a three dimensional **recovery vector** (**r**) in the form of (changed, missing, added) to each resource: we assign (0,0,0) to *identical* resources, (1,0,0) to *changed* resources, (0,1,0) to *missing* and (0,0,1) to *added*.

Although we would not have access to L if it were truly lost, we may crawl a currently existing website and then reconstruct it as if it were suddenly lost. Figure 1 illustrates taking a snapshot of a website by web crawling (left). The same website is reconstructed from the WI (right), and the crawled and recovered resources are then compared and categorized (bottom) using their URIs. Note that the union of L and R would be equivalent to the intersection of L and R if we could perfectly reconstruct the website from the WI, that is, have no missing or added resources.

A measure of change between the lost website L and the reconstructed website R can be described by summing the recovery vectors and normalizing them like so:

$$\text{difference}(L, R) = \left(\frac{r_c}{|L|}, \frac{r_m}{|L|}, \frac{r_a}{|R|} \right) \quad (1)$$

This **difference vector** is intuitively the percentage of resources that were changed, missing or added. A website that was reconstructed with all identical resources (a perfect reconstruction) would have a difference vector of (0,0,0). A completely unrecoverable website would have a difference vector of (0,1,0).

The difference vector is a useful summary of the reconstruction and can be useful for determining a level of reconstruction success. To quantify success, we need to examine several factors of the reconstruction. A simple definition of success would be the percent of resources that were recovered $(1 - d_m)$. But if some of the recovered resources were changed in such a way to make them less useful to us (e.g., a thumbnail was recovered instead of the full-sized image), we would want to assign some type of penalty to the changed resources (d_c). If our reconstruction resulted in many added resources (d_a) that hindered our ability to separate the "important" parts of the website from the chaff, we again may want to assign some sort of penalty. By assigning penalties to the components of the difference vector, we can compute a reconstruction success level that matches our intuitive notion of success.

To determine reconstruction success, we define a **penalty adjustment** which we may apply to each individual recovery vector or to the final difference vector. The penalty adjustment is composed of weights (P_c, P_m, P_a) which are defined over the interval of [0,1] with 0 being no penalty and 1 being the maximum penalty. We can adjust the weights depending upon the level of importance we would like to assign to resources in each recovery status category.

For example, suppose we lost a website of mostly PDFs but recovered 75% of them in an HTMLized format. We may assigned a weight of 1 to P_m to give the maximum penalty of not being able to recover the other 25% of the PDFs. We may assign 0.5 to P_c since the text of the PDFs that was recovered was helpful, but we lost the important PDF formatting of the text. We could use another penalty adjustment of 1 for P_c for those PDFs that contained only images since none of the images could be recovered from the HTMLized PDFs. We might want to assign a penalty of 0.2 to P_a if the added resources caused the reconstruction to take a significantly longer amount of time or if the added resources were not useful to us or caused us additional time in locating the resources that were important to us.

Once the penalty adjustment weights have been selected, they can be applied individually to each recovery vector before computing the difference vector:

$$\mathbf{r} = (r_c \cdot P_c,\ r_m \cdot P_m,\ r_a \cdot P_a) \qquad (2)$$

Alternatively, a single penalty adjustment could be applied to the final difference vector:

$$\text{difference}(L, R) = \left(\frac{r_c \cdot P_c}{|L|}, \frac{r_m \cdot P_m}{|L|}, \frac{r_a \cdot P_a}{|R|} \right) \qquad (3)$$

To measure how successful the reconstruction was, we take the L1 norm (the sum of the vector components) of the difference vector after applying the penalty adjustment:

$$\text{success} = d_c + d_m + d_a \qquad (4)$$

The closer the value of success is to zero, the more successful the reconstruction. Note that $d_c + d_m$ is always ≤ 1, and $d_c + d_m + d_a$ is always ≤ 2.

4. EXPERIMENT DESIGN & DEPLOYMENT

4.1 Sampling Websites

We initially wanted to choose a random set of websites that were representative of the Web at large. Sampling uniformly from the Web is currently not possible [35], so we sampled from the Open Directory Project (ODP) at `dmoz.org`. The ODP indexes a wide variety of websites in over 40 languages, and all search engines have an equal chance of indexing it.

We randomly selected URLs from the ODP that had a path depth of zero (`http://foo.org/`) or one (`http://foo.edu/~bar/`) in order to limit the selection to the root pages of websites. We crawled each website starting from the selected seed URL, and we crawled every resource that was accessible, regardless of MIME type.

We used Heritrix [29] as our crawler since it is built for doing deep crawls of multiple websites at the same time. We configured Heritrix to respect the robots exclusion protocol and delay an appropriate amount of time per request in order to avoid over-burdening any particular site [36]. To avoid common crawler traps, we limited the maximum path depth to 15 and maximum hop from the root page to 15. To avoid re-crawling the same resource multiple times, we normalized URLs to lowercase and stripped out common session IDs. And for simplicity, we restricted the download to port 80 and did not follow links to other hosts within the same domain name.

We continued to sample from the ODP data and crawl websites until we had found 300 accessible websites that matched a minimum set of qualifications. First, we rejected any websites that were entirely blocked by robots.txt or contained noindex/nofollow meta tags in the root page (only eight sites fit this description). Second, the websites had to contain valid content; websites with expired domains (two sites) or under reconstruction (one site) were rejected. And in order to ensure that our selected websites could be completely reconstructed within a one week time period, we rejected any websites that contained more than 10K resources when crawled (26 websites). Although Warrick is capable of reconstructing websites of any size, websites with more than 10K resources typically take more than a week to reconstruct due to the limited number of daily queries imposed by the web repositories. In terms of size, the sampled websites exhibited the power-law distribution that has been previously measured on the Web [1] where most sites had few resources and few sites had many resources.

4.2 Data Collection

For 14 weeks (late August to late November), we crawled each of the 300 websites using the same crawling policy as described previously. Crawls were preformed on weekends when traffic is typically low on most web servers. We also reconstructed all 300 websites weekly by running two Warrick processes on each of five servers. By running Warrick on different servers, we were able to make the most efficient use of the limited number of daily queries available from the web-repository APIs. The weekly crawls and reconstructions produced approximately 5 GB and 500 MB of compressed data, respectively.

We configured Warrick to start each website reconstruction with the base URL for each website. When the same resource was found in multiple repositories, Warrick selected the canonical version over the non-canonical version. If more than one canonical version was available, the most recent version was selected.

We used the Knowledgeable crawling policy for Warrick which was shown in earlier work [25] to be the most efficient policy in terms of number of repository queries and recovered files. This means that each repository was initially asked to list the URLs of resources it had stored (what we call lister queries), and only resources for which Warrick could find a link were recovered. For example, if lister queries revealed that a repository stored resources A (the base URL for the website), B and C, but the recovered resource A only contained a link to B, then C was not recovered. Therefore resources that were not connected in the reconstructed website's graph were not recovered. We correct for this potential bias when calculating success later in the paper. Note that lister queries do not always reveal all resources stored in a repository since search engines often limit their responses to the first 1000 (or fewer) results.

Over the course of the experiment, several of the websites became inaccessible. Three websites reported their bandwidth had been exceeded for a couple of weeks, and a few others appeared to be off-line or misconfigured for a few weeks. Two websites were inaccessible when they did not renew their domain name, but both re-appeared intact as the same site a few weeks later. One website's domain name quit resolving on week 10 and never became accessible again. A couple of websites changed domain names. When this happened, we added the new domain name to our list of sites to crawl and reconstruct. In this paper we have only computed statistics for successfully crawled websites.

5. EXPERIMENT RESULTS

5.1 Recovery Success

We first compare the reconstructions with the crawled sites to determine how successful the reconstructions were each week. As discussed in Section 3, we can assign a penalty adjustment that encodes the importance we give to resources based on their classification of changed, missing or added. We have defined five general levels of success in increasing order of laxity by applying various penalty adjustments to the final difference vector and relaxing how we categorize some recovered resources:

s1. (1,1,1) - Missing, changed and added are equally undesirable.
s2. (1,1,0) - Missing and changed are equally undesirable, but added resources are not.
s3. (1,1,0) - The definition of changed is relaxed by removing textual resources that are 'similar' from changed.
s4. (0,1,0) - Missing resources are undesirable, but changed and added are not.
s5. (0,1,0) - The definition of missing is relaxed by removing potentially recoverable resources.

In s3, we used a similarity algorithm based on shingles to determine how similar two textual resources were. We define **textual resources** to be those with a MIME type of 'text/*' or those with MIME types associated with PDF, PostScript or Microsoft Office documents. We classify resources as 'similar' if the crawled and recovered resources share at least 75% of their shingles. Shingling (as proposed by Broder et al. [6]) is a popular method for quantifying similarity of text documents when word-order is important. We used shingles of size 5 as was done in [12], and we stripped away all HTML markup before computing shingles. This allowed us to ignore markup changes and compare canonical documents (like PDFs) to their HTMLized equivalents.

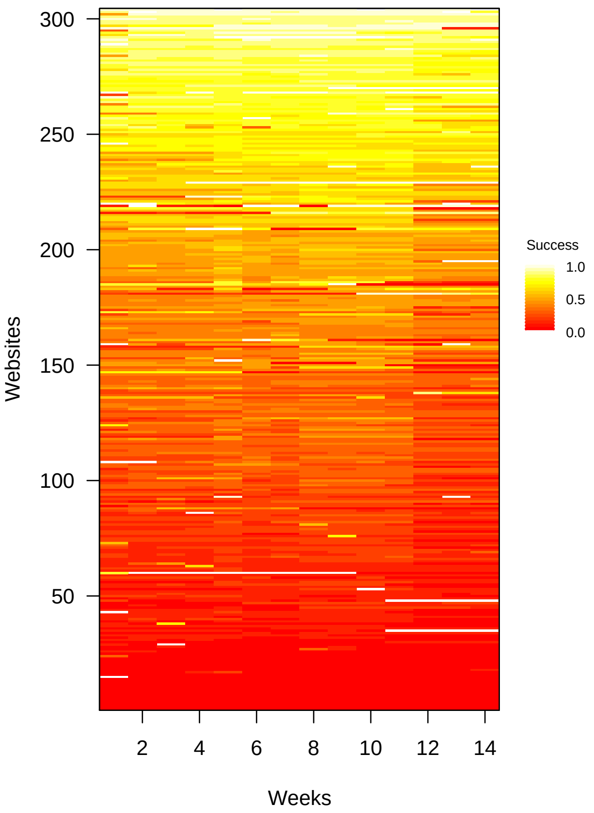

Figure 2: Success of reconstructions by week.

For s5, we configured Warrick to track those resources that were known to be stored in at least one repository but were not recovered due to the selection algorithm of the Knowledgable policy (as discussed in Section 4). Therefore if a resource was not recovered because we could not find a link to it, we could check to see if a lister query revealed the resource was stored in any of the repositories. If so, we could potentially have recovered it, and so we do not need to classify it as missing.

The s5 level is the most generous definition of success since it does not penalize for changed resources, and it eliminates the bias of using the Knowledgable policy in our reconstructions. The value 1 - s5 is intuitively the percentage of recovered resources for a website. For simplicity and clarity, we will use the 1 - s5 measure in other sections of this paper.

We first examine an overall picture of how successful the reconstructions were over time. In Figure 2 we plotted each website's weekly success (using s5) with the most successful reconstructions at the bottom (graphs with other previously defined penalty adjustments looked similar). Each horizontal line marks the reconstruction success rate for the same website each week. The figure is not intended to give detailed information about any one website; instead it shows that most websites were reconstructed to the same degree each week since the colors vary vertically but to a much lesser degree horizontally. But you can see there are some exceptions. For example, site number 2 was successfully reconstructed every week (red all the way across), but site 148 experienced a huge increase in success on week 6 when it

Table 1: Descriptive statistics for reconstructions.

	Mean	Median	Std	Min	Max	Websites with $s^* = 0$
s1	0.7761	0.8164	0.3266	0	2	3%
s2	0.7137	0.7817	0.2606	0	1	3%
s3	0.6250	0.6796	0.2726	0	1	5%
s4	0.4567	0.4278	0.2867	0	1	6%
s5	0.3901	0.3477	0.2764	0	1	10%

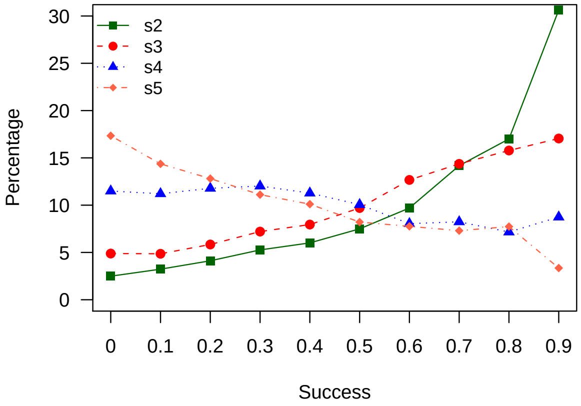

Figure 3: Distribution of s2 to s5 success.

went from yellow to red (upon manual observation, site 148 changed the dynamic portion of their site which accounts for the increase in success).

The descriptive statistics for the reconstructions are given in Table 1 along with the percent of websites that experienced at least one reconstruction where the measured success was perfect (zero). As we expected, the success values dropped closer to zero as we relaxed the penalty adjustments. Where only 3% of the websites ever had a perfect reconstruction under the s1 level, 10% did under the s5 level.

When we examine the distribution of the success levels (Figure 3), we see that the s2 and s3 levels are skewed to the right– about 16% of the reconstructions resulted in a 0.9 score or worse for s3, and 31% resulted in a 0.9 or worse for s2. The s4 and s5 levels (which do not penalize for changed resources) favor scores much closer to zero. Under the s5 measure, almost 17% of all reconstructions resulted in better than a 0.1 score. Note that we do not include s1 in the figure since it is distributed over the interval [0 to 2]; its distribution was skewed to the right similar to s2 and s3.

5.2 Content Type

The two most common types of content found in our 300 websites were HTML and images, accounting on average for 40% and 53% of all content, respectively. Other textual resources like PDF, PostScript and Microsoft Office made up only a small fraction (2%) of all resources. All other resources combined made up 5% of the content on average. HTML and textual resources proved to be the most recoverable. On average, we were able to recover 77% of the HTML resources and 75% of the textual resources. We recovered only 42% of the images and 32% of resources with some other MIME type.

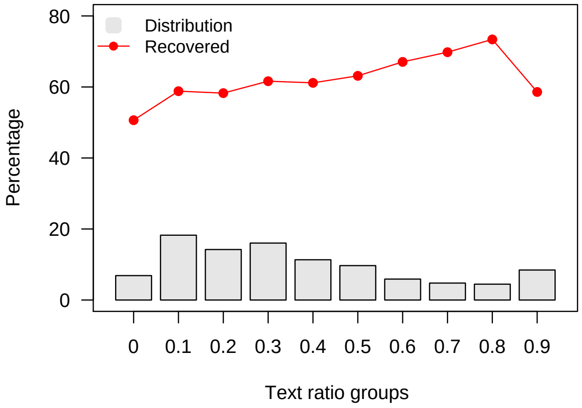

Figure 4: Distribution and recovery of websites based on ratio of textual resources.

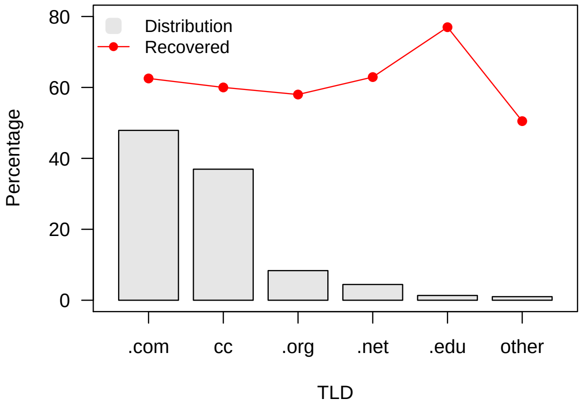

Figure 5: Distribution and recovery by TLD.

To see how recovery affects the amount of textual resources in a website, we computed the ratio of textual resources (HTML, PDF, MS Office, etc.) to other resource types for each website and placed each site in groups where the text ratio ranged from [r to $r + 0.1$). Figure 4 shows the distribution of websites (bars) based on the ratio of textual resources making up the site. The average percentage of recovered resources for the sites is shown as a line (this is equivalent to 1 - s5, as discussed in Section 5.1). The figure shows that a majority of sites had text ratios between 0.1 and 0.6. Although the recovery line grows higher for each group, there is a significant drop from 73% for group 0.8 to 59% for group 0.9. The percentage of textual resources in a website is thus not the only factor dictating its recoverability from the WI.

5.3 Top-Level Domain

The 300 websites represented a variety of top-level domains (TLDs). As shown in Figure 5, almost half of the sites were from the .com domain, and almost 40% were from a country code (cc) domain (there were 25 distinct cc TLDs). Only four sites were from .edu, two from .tv and only one from .info. From Figure 5, we also see the that most TLDs had a recovery rate around 60% with the exception of the four .edu sites which performed remarkably better.

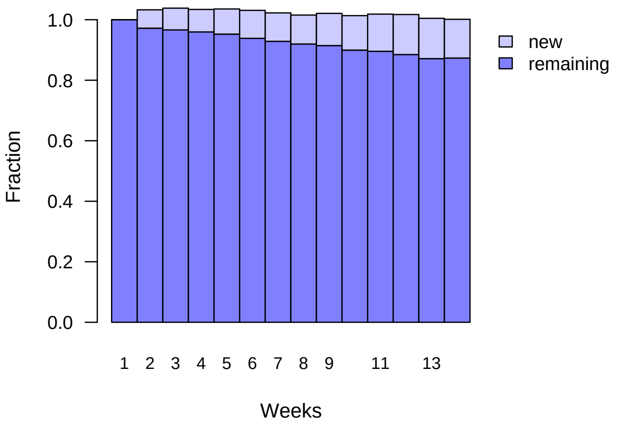

Figure 6: Fraction of resources from the first crawl still existing after n weeks (dark bars) and new resources (light bars).

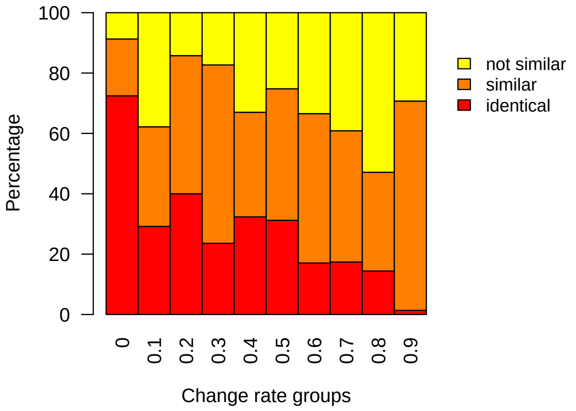

Figure 7: Recovery category of HTML resources grouped by change rate.

5.4 Birth and Decay

The 300 websites exhibited little growth during the experiment. Half of the websites did not add any new resources during the 14 weeks. We calculated the weekly birth rate of new resources (as performed in [32]) by examining the fraction of new URLs that we crawled each week that were not seen in any of the previous crawls. The average birth rate was a relatively stable 0.049. Only on week 9 did the average birth rate increase substantially, and that was due to a single website that added 10K new URLs that week (and dropped almost all of them the following week). The URLs appear to be dynamically generated, likely due to a configuration error on their web server. Discounting this website drops the average birth rate to 0.014.

There was also little decay in the websites. Over one third of the websites (39%) retained on week 14 all of the resources that were originally crawled on week 1. For those websites that did decay, new resources often replaced old ones. Figure 6 shows the fraction of new resources crawled each week (light bars) and the fraction of resources from week 1 that were also crawled on week n (dark bars). The bars are normalized so the number of resources in the first week is one. The figure illustrates that resources from week one slowly decayed (the dark bars gradually get smaller each week) and were usually replaced by new resources at different URLs (since the light bars hovered around 1.0). By week 14, the websites had lost about 13% of their resources on average.

5.5 Change Rate

We would expect many of the sampled websites to exhibit a broad range of dynamism. Some websites may remain unaltered for long periods of time, and others may undergo numerous changes each day. We can measure change for the resources in our sample by comparing the crawled resource on week n with week $n-1$. The change rate is the number of times we observed a change divided by the number of times we downloaded the resource minus one [11]. So a resource with a change rate of one means the resource changed every time it was crawled.

When we examined the distribution of change rates for the 300 websites, we found most of the resources (76%) did not change once during the 14 week period, and only 8% of the resources registered a change every time. Over a third of the websites (37%) did not have any resources that changed.

The resource type that exhibited the most amount of change were HTML resources. Most images, PDFs, style sheets, etc. remained relatively static. Whereas 44% of the HTML resources changed at least once, and 15% of them changed every time, only 0.8% of the images changed more than once during the experiment.

We were curious to see how the change rates would affect the recovery status (identical vs. changed) of recovered resources. So we examined the final week's reconstructions (since the change rates are most accurate by the final week) and placed all the recovered resources into groups where the change rates varied from $[r$ to $r + 0.1)$. For each group, we examined the percentage of resources that were identical, similar (shared 75% of their shingles) or not similar to their crawled counter-parts. In Figure 7 we plot our findings for HTML resources (since non-HTML resources exhibited little change). According to the figure, HTML resources that exhibited less than a 0.1 change rate had the highest percentage of identical recovered resources (72%). HTML resources with a change rate above 0.9 were rarely recovered in an identical state, but most (69%) were similar to their recovered counterparts.

We were surprised by the sharp drop in identical resources for group 0.1, but upon manual examination, we found several hundreds of pages from a single website that contained a MySQL error message embedded in them for two weeks in a row. If the dynamically generated pages had not been misconfigured when we crawled them, they would likely have been identical to the pages recovered from the WI.

We also manually examined the HTML resources with changed rates less than 0.1 that were 'not similar' to their recovered counterparts. Most of the resources actually appeared to be similar to the recovered pages, but sometimes non-English pages were transformed when cached, and our comparison function did not account for all transformations.

5.6 Age

Determining the age of resources on the Web can be tricky. When a resource is downloaded, the only indication about its age can be derived from the Last-Modified timestamp. The

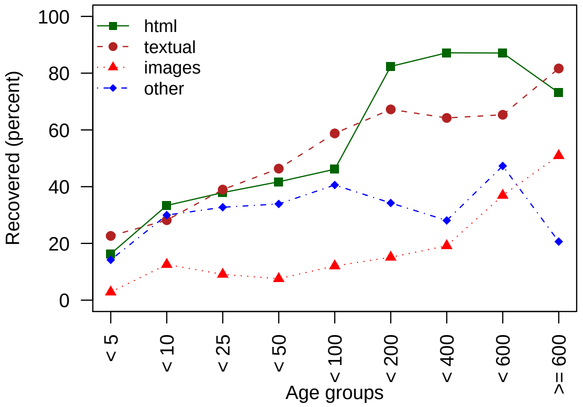

Figure 8: Percentage of resources recovered by age.

Last-Modified date is when the file was last modified, not when it was created, so it is a lower bound on the resource's age. Additionally, web servers sometimes report incorrect timestamps, and they do not report timestamps for dynamic pages. The only resources for which we can know their true age (with an error of a few days) are those that appear for the first time in a subsequent crawl. Even then it is possible for the resource to have been accessible at the same URL for a long time, but only before the crawl was a link to the resource added to the main website graph. Despite these limitations, we define a resource's age as the number of days between the current access time and the first access time or Last-Modified timestamp, which ever is oldest.

Only 36% of the HTML resources in our crawls had a Last-Modified timestamp, but more than 99% of textual and image resources had them. On the final round of crawling, 59% of HTML resources were less than one year in age. If we assume that the HTML resources missing a Last-Modified date that were crawled on the first week were also created that week, the percentage jumps to 85%. Images and textual resources were significantly older: 53% of images and 59% of textual resources were at least one year old.

To understand the relationship between age and recoverability, we grouped all crawled resources into 10 bins based on age. The bin breaks can be seen on the x-axis of Figure 8 (the first bin are resources less than 5 days old, the second less than 10 days old, etc.). We graphed each resource type in Figure 8 based on the percentage of resources that were recovered in that age group. From the figure we can see a general increase in recovery success for all four types of resources as they age. As we would expect, the newest resources were generally the least recoverable. But the drop for HTML and 'other' resource types in the final age category indicate that age may be not the single most significant predictor of recoverability.

5.7 Repository Contributions

We were interested in knowing which repositories were the most helpful in reconstructing the websites. Table 2 shows the percentage of resources that each repository contributed to the reconstructions. The table also lists the average number of weekly requests issued to each repository per website and the repository's **efficiency ratio**. A repository's efficiency ratio is the total number of recovered resources from the repository divided by the total number of issued repository requests.

Table 2: Statistics for web repositories.

Repository	Weekly contrib	Weekly requests per site	Efficiency ratio
IA	22.9%	127.2	0.38
Google	27.4%	152.6	0.38
MSN	32.4%	101.3	0.62
Yahoo	17.1%	232.2	0.24

In previous work [28], Google was the largest provider of resources with MSN in second place. Our new findings show MSN to be the largest contributor. One likely reason would be that in our previous study Warrick used page-scraping to obtain cached results, but in this study Warrick used the Google API. We have performed studies on the Google API that suggest it is serving from a smaller index than its web user interface [26]. A future version of Warrick will go back to page-scraping since Google has deprecated its SOAP-based web search API [8].

The significantly higher requests per website and lower efficiency ratio for Yahoo is likely due to the fact that Yahoo must often be asked twice if it has a particular URL stored, one request with a 'www.' prefix and another without. This behavior is documented in [25].

5.8 Crawler Directives

All four web repositories honor the robots exclusion protocol (robots.txt) which protects certain URL paths from being crawled. There were 63 websites in our sample (21%) that had a valid robots.txt file, and 14 of the files did not block any URL paths. Two sites (one selling sporting goods and another on-line video games) specifically denied the IA crawler (ia_archiver) access to their entire website but only blocked a handful of URL paths for other crawlers. One website placed a robots.txt file on their site on week 4 that gave explicit permission for most search engines to crawl their entire site but blocked access to all other crawlers. The file was removed on subsequent weeks, possibly because the webmaster discovered that the rogue crawlers s/he wanted to block typically ignore robots.txt anyway. There were two sites that gave specific directives to googlebot, but none for msnbot or slurp (Yahoo).

By far the most popular URL paths being blocked were cgi-bin, images and administrative paths. This implies that many potentially valuable resources are not being preserved in the WI because of the high resource demands the WI places on some websites or the perceived danger of having administrative content replicated in the WI. Crawler burdens will likely continue to be a problem until more efficient Web discovery methods are adopted [31].

Some webmasters like their websites being indexed by search engines but would prefer they not be cached. Reasons may include the loss of potential website traffic and the lack of control to quickly remove embarrassing or false content from the Web [34]. All four web repositories will refrain from caching or archiving an HTML page if it contains a `noarchive` meta tag.

Examining all HTML pages, we found only two websites in our sample using `noarchive` meta tags, both from the

.de ccTLD. The first site was protecting a personal blog from being cached, and the other was protecting all the PHP content from the commercial site. Interestingly, the second site only targeted Google; all other robots were allowed to cache the site. In another recent study [27], we found the use of noarchive meta tags to affect only 2% of pages indexed by Ask, Google, MSN and Yahoo. The low usage of noarchive meta tags suggests that few webmasters of typical sites want their pages kept out of search engine caches and web archives. It may also be that few webmasters are even aware of the existence of, or reasons for, using noarchive meta tags. Whatever the reasons, the current low adoption of opt-out caching and archiving mechanisms is encouraging from a web preservation standpoint.

6. RECONSTRUCTION MODEL

6.1 Factors for Successful Reconstruction

There are many factors which may contribute to the success of website reconstruction from the WI. For example, a website composed mostly of textual resources would likely be more successfully reconstructed than a site of mainly binary zip files since we know that all four repositories show a preference for textual resources over other types. We would also expect a website that is strongly connected to the web graph to be more recoverable than one with few inlinks since having greater inlinks increase the chance of a crawler finding the site. Older websites and sites that are more static in nature are also likely to be more recoverable.

In order to determine which factors contributed the most to reconstruction success of our websites, we ran several statistical tests on the recovered resources, examining several variables:

External backlinks: Websites with more inlinks (also called backlinks) to their root pages from other websites are more likely to be discovered by other crawlers and could possibly be crawled more frequently due to their importance. Lacking a large crawl of the entire Web, we used the backlink facility of Google, MSN and Yahoo to determine the known backlinks to the root page of each website every week. This measure is not as precise as we would like since Google does not reveal all known backlinks [9] and IA does not have a mechanism to reveal backlinks.

Internal backlinks: We would expect web crawlers to more easily find resources that contain a large number of backlinks within the site. Resources with few links may also likely be new additions to the website.

Google's PageRank: We would expect Google to revisit a website frequently if it has a high PageRank, and therefore the website would contain a larger footprint within Google than a site with a low PageRank. Google is the only search engine that publicly reports its 'importance' measure for a website, but it is possible other search engines assign similar importance values to the same websites. Google's PageRank can be obtained manually using the Google Toolbar although Google representatives in the past have reported the value is several months old [13].

Hops from root page: Crawlers often place hop count limits when crawling websites, so we would expect websites with its pages closer to the root page to be better reconstructed than sites with pages far from the root.

Path depth: Like hops, crawlers may reject URLs with long path depths.

MIME type: Websites with mostly HTML pages are likely to be more successfully reconstructed than sites with mostly images.

Query string parameters: Crawlers may reject dynamic pages with many query string parameters.

Age: Websites that have very old resources are more likely to be stored in the WI than websites with new resources. This is especially true since only resources that are at least 6-12 months old are accessible from IA [18].

Resource birth rate: Websites that are producing new content at new URIs are less likely to be reconstructed than websites that are not increasing in URIs.

TLD: It is possible that a bias exists for the web repositories for particular TLDs [37, 38].

Website size: It is possible that very large websites (in terms of number of resources) may have fewer of their resources cached/archived than smaller sites.

Size of resources: We speculate that the longer amount of time to download large resources may hinder their being cached or archived.

We did not factor the use of Flash, JavaScript, etc. by the websites since we used a crawler that was likely of equal or lesser technical capability when compared to the crawlers used by the web repositories. Our crawler (Heritrix) and their crawlers are likely to discover the same number of resources on the same website.

6.2 Analysis

We applied several statistical tests using SAS software (version 9.1) to the recovered and missing resources (143,001 observations) from the final week of reconstructions when the age variable was most accurate. We first examined the Pearson's correlation coefficient to see if there was a correlation between any of the above mentioned variables. The highest correlation (0.428, $p < 0.0001$ where p is the p-value of the test of zero correlation) was between hops and the website's size (we transformed the website size and resource size by taking the log of each to make the data fit more accurately the normal distribution, an assumption of this test). The positive correlation matches our intuition that it takes more hops to reach resources from the root page in larger websites. There was also a mild correlation between hops and path depth (0.388, $p < 0.0001$) which we would expect since URLs with greater path depth are often located further down the web graph from the root page.

There was a mild negative correlation between age and number of query parameters (-0.318, $p < 0.0001$). This may be because dynamically produced pages are easier to add to a website (for example, by adding more records to a database) and because determining the age of dynamic pages is problematic as discussed in Section 5.6.

Finally, there was also a mild positive correlation between external links and PageRank (0.339, $p < 0.0001$), website size (0.301, $p < 0.0001$), and hops (0.320, $p < 0.0001$). We would expect there to be some correlation between external links and PageRank since Google's PageRank is at least partially influenced by external backlinks. We may explain the correlation between external links and website size by reasoning that larger websites tend to attract more links, either because the effort to create a larger website may imply the website is more important or of higher quality or because websites with many pages are more easily found by search engines and therefore will garner more links over time [7].

Since none of the correlations were above 0.5, we did not remove any of the variables from our model. Next we ran a generalized linear model analysis to determine which of the variables were most important in explaining the model. We added the website's host name to the analysis since it is possible that, all things being equal from the twelve parameters, two websites may still experience different levels of recovery. The resulting analysis had an R-square value of 0.468041 (DF = 322, $p < 0.0001$), meaning that the model explains about half of the variations we observed. According to the type III sum of squares analysis, all thirteen variables were significant at the $p < 0.0001$ level except website size which was significant at the $p < 0.05$ level.

We then performed a multiple regression analysis with the ten continuous variables (ignoring the categorical variables of host, TLD and MIME type since categorical variables are not appropriate to this analysis) to determine how the variables impact the model. The analysis had an R-squared of 0.1943 (DF = 10, $p < .0001$), and the parameter estimates are shown in Table 3. An analysis showing the three most significant variables (if none of the others were available) produced PageRank, hops and age (R-squared = 0.1496).

The parameter estimates confirm our initial hypotheses on the effect of each variable in the overall success of website reconstruction. The only parameter which did not fit our intuition was resource size. According to the analysis, resources have a slightly better chance of being recovered as their size increases. This may be because very small resources are not indexed by some search engines or have a higher chance of being dropped during the de-duping processes. A caveat to resource size is that search engines often limit the amount of data they will cache from any particular resource. In a previous experiment [27], we found Yahoo will not cache more than 215 KB from a textual resource; Google (977 KB) and MSN (1 MB) also have limits.

The results of our multiple regression analysis can help predict how much of a website can be recovered if it were to be lost today. The model has a rather low R-squared value which indicates there are other parameters affecting website reconstruction which we have not measured. One reason our model does not have a higher R-squared value is because IA and the three search engines have very different crawling and caching priorities. Had our reconstructions been performed with only IA or only the search engines, our analysis would likely have been different. We must also remind the reader that calculating age, external backlinks and PageRank are not finely tuned processes: each have a wide range of error. And we speculate that webmasters submitting URLs directly to search engines and website discovery methods similar to Google's Sitemap Protocol (discussed earlier in Section 2) account for some of the unexplained portions of our model.

7. CONCLUSIONS

We have taken snapshots of 300 websites from the Web and WI over a period of three months. We discovered that most of the sampled websites were relatively stable; over a third of the websites never lost a single resource over the entire experiment, and half of the websites never added any new resources. We also found that 37% of the websites did not have a single resource that registered a change during the 14 weeks. More than half of all images and textual resources were more than a year old, but at least 59% of the HTML resources were less than a year old (or had been modified within the year).

Table 3: Regression parameter estimates.

Variable	Param Est	$P_r > \|t\|$
Intercept	0.76071	$< .0001$
External backlinks	-3.96E-7	$< .0001$
Internal backlinks	0.00004	$< .0001$
Birthrate	-0.13361	$< .0001$
PageRank	0.08162	$< .0001$
Website size	-0.04074	$< .0001$
Hops	-0.04184	$< .0001$
Path depth	-0.06044	$< .0001$
Query params	-0.04342	$< .0001$
Resource size	0.00248	.0018
Age	0.00014	$< .0001$

We found many of the websites encouraged crawling by web repositories. Only 21% of the websites had a valid robots.txt, and a fifth of those did not block any URLs from being crawled. Only two websites used `noarchive` meta tags to keep their resources from being cached and archived.

From our analysis, the typical website indexed by ODP can expect to get back 61% of its resources if it were lost today (77% textual, 42% images and 32% other). The three most significant things a website can do to improve its chances are to improve its PageRank, decrease the number of hops a crawler must take to find all the website's resources, and of course create stable URLs for all resources. Google provides a number of tips for webmasters to improve their website PageRank scores, including admonitions to increase external backlinks, get listed in directories like the ODP, and use few query string parameters [15]. Our study has confirmed the intuitive notion that websites that are crawler-friendly are more likely to be better preserved by the WI.

8. ACKNOWLEDGMENTS

This work is supported in part by NSF Grant IIS-0610841.

9. REFERENCES

[1] L. A. Adamic and B. A. Huberman. Zipf's law and the Internet. *Glottometrics*, 3:143–150, 2002.

[2] Alexa toolbar. `http://download.alexa.com/`.

[3] Z. Bar-Yossef, A. Z. Broder, R. Kumar, and A. Tomkins. Sic transit gloria telae: towards an understanding of the web's decay. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 328–337, 2004.

[4] Z. Bar-Yossef and M. Gurevich. Random sampling from a search engine's index. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 367–376, 2006.

[5] B. E. Brewington and G. Cybenko. How dynamic is the Web? *Computer Networks*, 33(1–6):257–276, 2000.

[6] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the Web. *Computer Networks & ISDN Systems*, 29(8-13):1157–1166, 1997.

[7] J. Cho and S. Roy. Impact of search engines on page popularity. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 20–29, 2004.

[8] D. Clinton. Beyond the SOAP search API, Dec. 2006. http://google-code-updates.blogspot.com/2006/12/beyond-soap-search-api.html.

[9] M. Cutts. GoogleGuy's posts, June 2005. http://www.webmasterworld.com/forum30/29720.htm.

[10] Z. Dalal, S. Dash, P. Dave, L. Francisco-Revilla, R. Furuta, U. Karadkar, and F. Shipman. Managing distributed collections: Evaluating web page changes, movement, and replacement. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 160–168, 2004.

[11] F. Douglis, A. Feldmann, and B. Krishnamurthy. Rate of change and other metrics: a live study of the World Wide Web. In *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, 1997.

[12] D. Fetterly, M. Manasse, M. Najork, and J. Wiener. A large-scale study of the evolution of web pages. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 669–678, 2003.

[13] J. Galt. Google says: Toolbar PageRank is for entertainment purposes only, 2004. http://forums.searchenginewatch.com/showthread.php?t=3054.

[14] Google Sitemap Protocol. https://www.google.com/webmasters/tools/docs/en/protocol.html.

[15] Google webmaster help center: Webmaster guidelines, 2007. http://www.google.com/support/webmasters/bin/answer.py?answer=35769.

[16] A. Gulli and A. Signorini. The indexable web is more than 11.5 billion pages. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 902–903, May 2005.

[17] T. L. Harrison and M. L. Nelson. Just-in-time recovery of missing web pages. In *HYPERTEXT '06: Proceedings of the 17th ACM conference on Hypertext and Hypermedia*, pages 145–156, Aug. 2006.

[18] Internet Archive FAQ: How can I get my site included in the Archive? http://www.archive.org/about/faqs.php.

[19] Jon. How the Google cache can save your a$$, Dec. 2005. http://www.smartmoneydaily.com/Business/How-the-Google-Cache-can-Save-You.aspx.

[20] W. Koehler. An analysis of web page and web site constancy and permanence. *Journal of the American Society for Information Science*, 50(2):162–180, 1999.

[21] S. Lawrence, D. M. Pennock, G. W. Flake, R. Krovetz, F. M. Coetzee, E. Glover, F. A. Nielsen, A. Kruger, and C. L. Giles. Persistence of web references in scientific research. *Computer*, 34(2):26–31, 2001.

[22] C. Marhsall, F. McCown, and M. L. Nelson. Evaluating personal archiving strategies for Internet-based information. In *Proceedings of IS&T Archiving 2007*, May 2007.

[23] F. McCown. Mark Foley websites - reconstructed, 2006. http://www.cs.odu.edu/~fmccown/foley/.

[24] F. McCown, X. Liu, M. L. Nelson, and M. Zubair. Search engine coverage of the OAI-PMH corpus. *IEEE Internet Computing*, 10(2):66–73, Mar/Apr 2006.

[25] F. McCown and M. L. Nelson. Evaluation of crawling policies for a web-repository crawler. In *HYPERTEXT '06: Proceedings of the 17th ACM conference on Hypertext and Hypermedia*, pages 145–156, 2006.

[26] F. McCown and M. L. Nelson. Agreeing to disagree: Search engines and their public interfaces. In *JCDL '07: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital Libraries*, 2007.

[27] F. McCown and M. L. Nelson. Characterization of search engine caches. In *Proceedings of IS&T Archiving 2007*, 2007.

[28] F. McCown, J. A. Smith, M. L. Nelson, and J. Bollen. Lazy preservation: Reconstructing websites by crawling the crawlers. In *Proceedings from the 8th ACM International Workshop on Web Information and Data Management (WIDM '06)*, pages 67–74, 2006.

[29] G. Mohr, M. Kimpton, M. Stack, and I. Ranitovic. An introduction to Heritrix, an archival quality web crawler. In *Proceedings of the 4th International Web Archiving Workshop (IWAW '04)*, Sept. 2004.

[30] M. L. Nelson and B. D. Allen. Object persistence and availability in digital libraries. *D-Lib Magazine*, 8(1), 2002.

[31] M. L. Nelson, J. A. Smith, I. Garcia del Campo, H. Van de Sompel, and X. Liu. Efficient, automatic web resource harvesting. In *Proceedings from the 8th ACM International Workshop on Web Information and Data Management (WIDM '06)*, pages 43–50, 2006.

[32] A. Ntoulas, J. Cho, and C. Olston. What's new on the Web? The evolution of the Web from a search engine perspective. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 1–12, 2004.

[33] S. Olsen. Court backs thumbnail image linking. *CNET News.com*, July 2003. http://news.com.com/2100-1025_3-1023629.html.

[34] S. Olsen. Google cache raises copyright concerns. *CNET News.com*, July 2003. http://news.com.com/2100-1038_3-1024234.html.

[35] M. Thelwall. Methodologies for crawler based web surveys. *Internet Research*, 12(2):124–138, 2002.

[36] M. Thelwall and D. Stuart. Web crawling ethics revisited: Cost, privacy, and denial of service. *Journal of the American Society for Information Science and Technology*, 57(13):1771–1779, 2006.

[37] M. Thelwall and L. Vaughan. A fair history of the Web? Examining country balance in the Internet Archive. *Library & Information Science Research*, 26(2):162–176, 2004.

[38] L. Vaughan and M. Thelwall. Search engine coverage bias: Evidence and possible causes. *Information Processing & Management*, 40(4):693–707, 2004.

[39] Yahoo Site Explorer. http://siteexplorer.search.yahoo.com/.