On the Evolution of Clusters of Near-Duplicate Web Pages

Dennis Fetterly
Microsoft Research
1065 La Avenida
Mountain View, CA 94043
fetterly@microsoft.com

Mark Manasse
Microsoft Research
1065 La Avenida
Mountain View, CA 94043
manasse@microsoft.com

Marc Najork
Microsoft Research
1065 La Avenida
Mountain View, CA 94043
najork@microsoft.com

Abstract

This paper expands on a 1997 study of the amount and distribution of near-duplicate pages on the World Wide Web. We downloaded a set of 150 million web pages on a weekly basis over the span of 11 weeks. We then determined which of these pages are near-duplicates of one another, and tracked how clusters of near-duplicate documents evolved over time. We found that 29.2% of all web pages are very similar to other pages, and that 22.2% are virtually identical to other pages. We also found that clusters of near-duplicate documents are fairly stable: Two documents that are near-duplicates of one another are very likely to still be near-duplicates 10 weeks later. This result is of significant relevance to search engines: Web crawlers can be fairly confident that two pages that have been found to be near-duplicates of one another will continue to be so for the foreseeable future, and may thus decide to recrawl only one version of that page, or at least to lower the download priority of the other versions, thereby freeing up crawling resources that can be brought to bear more productively somewhere else.

1. Introduction

In a 1997 study, Broder *et al.* [4] presented a technique for estimating the degree of similarity among pairs of documents. Their technique, which they call *shingling*, is purely syntactic; it does not rely on any linguistic knowledge other than the ability to tokenize documents into a list of words. Their technique extracts all k-word sequences of adjacent words (herein called *shingles*); two documents are considered equal if they contain the same set of shingles, and highly similar if their sets of shingles significantly overlap. Broder *et al.* used an unbiased deterministic sampling technique to reduce the set of shingles to a small, yet representative, subset. This sampling reduces the storage requirements for retaining information about each document, and it reduces the computational effort of comparing documents. Broder *et al.* applied their technique to a set of 30 million web pages obtained from an AltaVista crawl, and grouped these pages into clusters of very similar documents. They found

that roughly one third of the pages in their data set were near-duplicates of other pages.

There are several techniques that are very similar to shingling, but that use sequences of adjacent characters instead of word sequences. Manber developed such a technique to find similar files in a file system [11]. Heintze developed a similar technique for detecting near-duplicate documents [9]. He applied it to a set of technical reports to determine how similar they are, and proposed to use it track updates to technical papers, and as a method for detecting plagiarism.

Shivakumar and Garcia-Molina proposed a different technique for locating plagiarized documents. Their technique segments a document into a set of non-overlapping chunks, and stores a hash value for each chunk in a table. They consider word-sized, sentence-sized, document-sized, and variable-sized chunks (variable-sized chunks are sequences of words whose length is determined by the hash values of individual words).

The study of web clusters is related to the study of "mirrors" on the web (see for example [1, 7, 2, 6]). Two web sites are mirrors of one another if a substantial portion of the pages on one site is duplicated (or near-duplicated) on the other site, and if each replicated page on both sites is addressed by URLs with the same suffix. Some mirrors are caused by an organization replicating their content across multiple web servers, for the purpose of geographic distribution or rebranding. Others are caused by independent organizations maintaining collections of standards documents, e.g. universities maintaining collections of RFCs and Unix manual pages. One could characterize a set of mirror sites as a large set of clusters, where each cluster covers all versions of a replicated page.

In a 1999 study, Bharat and Broder [1] examined 180 million URLs to look for patterns of shared URL suffices, identifying about 30,000 candidate mirror pairs. They then downloaded around 20 URLs from each candidate mirror and checked if the corresponding pages were near-duplicates. This technique validated almost 17,000 of their candidate mirror pairs as actual mirror pairs.

Broder *et al.* proposed to use shingling to characterize how pages change over time. The PageTurner study [8] realized this goal. In that study, we downloaded a set of 150 million

web pages on a weekly basis, over the span of 11 weeks, and measured the amount of change per URL, and investigated which aspects of a web page are predictive of change. We used a variant of Broder *et al*'s shingling technique to reduce the amount of data retained for each downloaded document. In the PageTurner study, we considered only the evolution of individual pages, whereas this study considers the evolution of clusters of near-duplicate pages.

This paper expands on Broder *et al.*'s 1997 of the amount and distribution of near-duplicate pages on the World Wide Web. The study at hand is based on the data set collected by the PageTurner experiment. Here, we consider which of these pages are near-duplicates of one another. We found that 29.2% of all web pages are very similar to other pages, and that 22.2% are virtually identical to other pages. Apart from covering a larger set of web pages than Broder *et al.*, the main contribution of our study is to explore the temporal aspects of clusters; that is, we investigate how clusters of near-duplicate pages evolve over time. We also found that clusters of near-duplicate documents are fairly stable: Two documents that are near-duplicates of one another are very likely to still be near-duplicates 10 weeks later. This result is of significant relevance to search engines: Web crawlers can be fairly confident that two pages that have been found to be near-duplicates of one another will continue to be so for the foreseeable future, and may thus decide to recrawl only one version of that page, or at least to lower the download priority of the other versions, thereby freeing up crawling resources that can be brought to bear more productively somewhere else. We are currently examining our data for the presence of mirrors, and are studying the evolution of such mirrors.

The remainder of this paper is structured as follows: Section 2 describes our experimental framework, contrasting our techniques to those used in previous studies. Section 3 presents the results of our investigation. Finally, Section 4 puts our results into perspective and identifies avenues of future work.

2. Experimental Framework

Our results are based on data collected in the course of the "PageTurner" project [8]. This project was aimed at measuring the amount of textual changes in individual web pages over time. To this end, we crawled a set of slightly over 150 million web pages once a week, starting in November 2002 and continuing for 11 weeks. For every downloaded page, we recorded a vector of 84 features, together with other salient information, such as the URL, the HTTP status code (or any DNS or TCP error), the document's length, the number of words, etc.

We computed the feature vectors using a modified version of the document shingling technique due to Broder *et al.* [4], which uses a metric of document similarity based on syntactic properties of the document. In order to compare two documents, we map each document into a set of

k-word subsequences (groups of adjacent words or "shingles"), wrapping at the end of the document, so that every word in the document starts a shingle.

Two documents are considered to be identical if they map to the same set of shingles; they are considered to be similar if they map to similar sets of shingles. Quantitatively, the similarity of two documents is defined to be the number of distinct shingles appearing in both documents divided by the total number of distinct shingles. This means that two identical documents have similarity 1, while two documents that have no shingle in common have similarity 0.

We collected the data using the Mercator web crawler [12], customized to collect feature vectors for each downloaded page. Our feature vector extraction module substituted HTML markup by whitespace, and then segmented the document into 5-word shingles, where each word is an uninterrupted sequence of alphanumeric characters. Next, it computed a 64-bit checksum of each shingle, using Rabin's fingerprinting algorithm [3, 13]. We call these fingerprints the "pre-images". Next, the module applied 84 different (randomly selected but fixed thereafter) one-to-one functions to each pre-image. Each function is a 64-bit Rabin fingerprinting function. The 84 fingerprinters use different randomly-chosen primitive polynomials of degree 64. The functions are one-to-one because Rabin fingerprints map distinct 64-bit values to distinct results. We cannot prove that these functions are min-wise independent [5], but they seem acceptable in practice. For each function, we retained the pre-image which results in the numerically smallest image. This resulted in a vector of 84 pre-images, which is the desired feature vector. Given the feature vectors of two documents, two corresponding elements of the vectors are identical with expectation equal to the similarity of the documents.

It is worth noting that this variant of shingling differs from those described by Broder *et al.* [4] in their 1997 paper on "syntactic clustering of the web". They, like us, use Rabin's fingerprinting algorithm to map shingles to pre-images, but our approaches differ in how pre-images are sampled. We use a fixed number of one-to-one functions to map pre-images to images and then select for each one-to-one function the pre-image that maps to the minimal image. We end up with a constant number of features, regardless of the number of words in the original document; this technique was first employed by Broder, Burrows, and Manasse in 1998. It works even for documents that contain very few words, unlike our earlier techniques. Finally, this technique allows us to measure the similarity between two documents by comparing only corresponding elements from two feature vectors.

Broder *et al.* suggest two alternative methods for sampling the set of shingles: The "MOD_m" technique selects all those pre-images whose value modulo m is zero; in other words, this method produces a variable-length feature set (as opposed to an ordered, fixed-sized vector), where shorter documents will have fewer features, and very short documents might have no features at all. A variant, the "MOD_{2^i}"

method, attempts to rectify these problems by adjusting i such that a roughly constant number of elements is selected. The MIN_s method selects the s numerically smallest pre-images (or all the pre-images, if there are fewer than s). In all of these earlier techniques, measuring the similarity between two documents is more complicated that simply comparing only corresponding elements from two feature vectors; it requires computing the intersection of two feature sets.

Crawling left us with 44 very large logs (produced by the eleven crawls on the four crawlers), each spanning multiple files, one per day. The logs totaled about 1,200 GB, whereas the sampled documents took up a mere 59 GB.

As they were, these logs were not suitable for analysis yet, because the URLs occurred in non-deterministic order in each log. We sorted the logs to remove this nondeterminism, and then merged and condensed the logs. Each combined record contained information that allowed to determine how documents changed over time, but in particular it contained six "supershingles" for each of the 11 versions of each URL. Each "supershingle" represents the concatenation of 14 adjacent pre-images. Broder *et al.* [4] described supershingling, but their method for aggregating features into supershingles is highly sensitive to the insertion of words into a document.

Due to the independence of the one-to-one functions used to select pre-images, if two documents have similarity p, each of their supershingles matches with probability p^{14}. For two documents that are 95% similar, each supershingle matches its counterpart with probability 49%. Given that we retain six supershingles, there is a probability of almost 90% that at least two of the six supershingles will agree for documents that are 95% or more similar.

To more efficiently discover pairs of documents sharing at least two supershingles, we perform the concatenation of all pairs of distinct supershingles, with the lesser-numbered supershingle first, thereby producing 6 choose 2 (i.e. 15) "megashingles". Two documents that are 95% similar thus will have an almost 90% chance of having a megashingle in common, while two documents which are 80% similar only have a 2.6% chance of having a megashingle in common.

We use megashingles to compute clusterings of near-duplicate documents (that is, documents having two supershingles in common). In particular, we maintain 15 hash tables, one of each pair of supershingles. [1] The hash tables map megashingles to "document identifiers" (integers identifying a record in the condensed logs of the PageTurner experiment). For each document we are considering, we compute its 15 megashingles (based on the 6 supershingles found in the condensed logs). For each megashingle, we check the

---

[1]Numerical matches between shingles produced by two different fingerprinting functions is coincidental. By the same token, numeric matches between supershingles that are the combination of unrelated shingles is coincidental, as are numeric matches between megashingles that are derived from unrelated supershingles. Maintaining a separate hash table for each combination of supershingles ensures that such coincidental matches are not interpreted as near-similarity between documents.
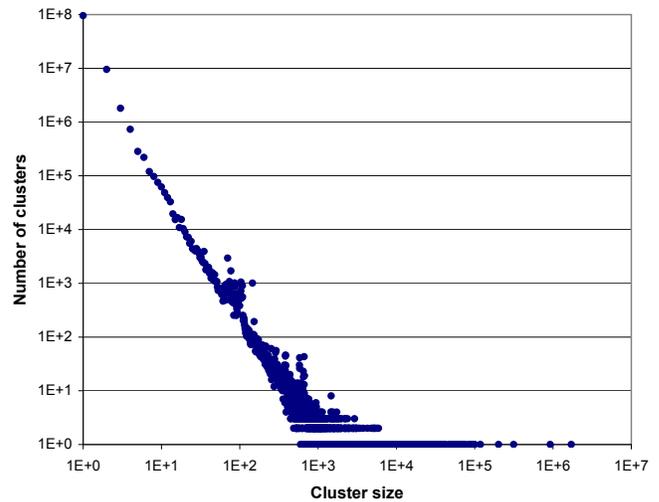

Figure 1. Distribution of sizes of clusters of very similar documents in week 1

corresponding hash table to determine if that megashingle has been encountered before. If the megashingle is not contained in the hash table, we add it together with the corresponding document ID. Conversely, if the hash table already contains a mapping from this megashingle to another document ID, the two documents belong in the same cluster. To maintain the clustering, we build a Union-Find data structure [10].

Using hash tables of megashingles and union-find forests of document ID sets, our running time is close to linear in the number of documents. This is in sharp contrast to the approach taken by Broder *et al*. Their 1997 experiment required more than 10 CPU days to cluster 30 million documents at the 50% similarity level; we are able to cluster 150 million documents in about 3 hours, using three year newer hardware.

3. Results

We used the clustering algorithm described in the previous section to partition the documents into sets of very similar documents, that is, documents that have at least two supershingles in common. We clustered each of the 11 weeks separately. Of the 150,836,209 documents that we downloaded during the first week, 57,305,947 were similar to some other document downloaded that week, while 93,530,262 documents were not similar to any other document. The documents that were similar to others fell into 13,283,856 clusters. In other words, 44,022,091 documents, or 29.2% of all the documents downloaded, were near-duplicates of the 13,283,856 "canonical" documents representing the clusters.

Figure 1 shows the distribution of the sizes of clusters of very similar documents in the first week. The horizontal axis depicts the cluster size, on a logarithmic scale ("1E+3" indicates clusters of size 10^3); the vertical axis shows the number of disjoint clusters of that size (again on a logarith-

Table 1. Characterization of the 20 largest clusters of very similar documents

#URLs	#Hosts	#Machines	Description	Representative URL
921374	6286	3	Pornography	http://fr.gncix.cc/page1.html
315788	335	199 (+ 2)	Tucows software download pages	http://games.fastnet.it/news.html
201869	15	15	Health food store	http://www.usrma.com/vitamins/jodahl/jodahl.htm
118530	4	2	Directory of car sites	http://www.100topauto.com/SiteMap
100428	11977	14	Pornography	http://hot.fuckjpg.com/
92629	43906	31422 (+ 1306)	Nothing but title "Untitled document"	http://gc.dk/
91207	53532	2	Pornography	http://www.the-girl.net/Sex/sex
84421	121	1	CitySearch login page	http://boise.citysearch.com/review/29/add
75766	4	2	Directory of clip art sites	http://www.100topclipart.com/SiteMap
70081	3	1	Directory of career sites	http://career.100topcareer.com/SiteMap
69589	3	1	Directory of art-related sites	http://www.100topart.com/SiteMap
64297	29713	10 (+ 18418)	Pornography	http://gay-day.com/
63714	3	2	Directory of music sites	http://music.100topmusic.com/SiteMap
63298	3	1	Directory of education-related sites	http://www.100topeducation.com/SiteMap
61318	3	2	Directory of business-related sites	http://www.100topbusiness.com/SiteMap
60923	3	1	Directory of cartography sites	http://www.100topmap.com/SiteMap
59848	4	2	Directory of golf sites	http://www.100topgolf.com/SiteMap
59358	22	20 (+ 1)	Tucows themes pages	http://freethemes.ip.pt/golf.html
57870	4	1	Directory of news sites	http://www.100topnews.com/SiteMap
53555	162	133 (+ 1)	Tucows software download pages	http://tucows.mts.net/preview/194076.html

mic scale). The bulk of the curve fits a linear trend, which (given the logarithmic scale of the axes) suggests a power law distribution with an exponent of roughly 2.5.

Note that there are several regions in the curve where the datapoints deviate from the trendline. In particular, for cluster sizes of around 100, we see numerous clusters where the number of clusters of a given size exceeds the trendline by a factor of 5 to 10. We have investigated some of these outliers. In every case we have examined, the anomaly is due to mirroring. For example, for cluster size 108, over 90% of the clusters of that size each consist of URLs from the Tucows collection, each of these clusters contains very similar (possibly identical) Tucows[2] web pages that are served by around 100 different web servers.

Figure 2 is similar to figure 1, but shows the cluster size distribution for the last week of the PageTurner crawl (black points) superimposed onto that of the first week (light red points). As can be seen, the trendline of the last week has been vertically translated down by a tiny amount, reflecting the fact that some URLs that were live during the first week have subsequently become unreachable. The slopes of the two curves do not differ, indicating that the exponent of the power law distribution remained constant.

Table 1 lists the 20 largest clusters. The first column shows the number of distinct URLs in each cluster (listed in decreasing order). The second column shows the number of distinct symbolic host names of these URLs. The third column shows the number of distinct machines (we assume two different symbolic host names are served up by the same machine if they resolve to the same IP address[3]). The fourth column describes the content of each cluster. Finally, the fifth column gives an exemplary URL from that cluster.

For 17 of the 20 clusters shown in table 1, the web pages contained in each cluster do not differ in any meaningful way. For example, the web pages in the third cluster are store fronts of an online health food retailer. The 201,869 store front pages are identical to one another, with the exception of the name of the franchisee changing. In other words, a search engine would be well-advised to index only a canonical representative from these 17 clusters. On the other hand, the pages in each of the three Tucows clusters, while being very similar to all other pages in the same cluster, differ in one important aspect: the name of the software item to download. Given that this change constitutes the most relevant part of the page, a search engine should index all the pages (or at least all the changing portions).

As we explained above, we consider two documents to be *very similar* if they have two supershingles in common. We used this notion of similarity when performing the clustering whose distribution is shown in figures 1 and 2. We have performed another clustering using a more stringent definition of

---

Table 2. Characterization of the 20 largest clusters of virtually identical documents

#URLs	#Hosts	#Machines	Description	Representative URL
118529	4	2	Directory of car sites	http://www.100topauto.com/SiteMap
92629	43906	31422 (+ 1306)	Nothing but title "Unitled document"	http://gc.dk/
75766	4	2	Directory of clip art sites	http://www.100topclipart.com/SiteMap
70081	3	1	Directory of career sites	http://career.100topcareer.com/SiteMap
69588	3	1	Directory of art-related sites	http://www.100topart.com/SiteMap
64297	29713	10 (+ 18418)	Pornography	http://gay-day.com/
63714	3	2	Directory of music sites	http://music.100topmusic.com/SiteMap
63298	3	1	Directory of education-related sites	http://www.100topeducation.com/SiteMap
61318	3	2	Directory of business-related sites	http://www.100topbusiness.com/SiteMap
60923	3	1	Directory of cartography sites	http://www.100topmap.com/SiteMap
59848	4	1	Directory of golf sites	http://www.100topgolf.com/SiteMap
57870	4	2	Directory of news sites	http://www.100topnews.com/SiteMap
51399	3	1	Directory of floral sites	http://www.100topflower.com/SiteMap
48267	2	1	Directory of weather-related sites	http://www.100topweather.com/SiteMap
47461	2	1	Directory of book-related sites	http://www.100topbook.com/SiteMap
41092	4	1	Directory of government and political sites	http://www.100topgovernmentsites.com/SiteMap
39888	3	2	Directory of reference sites	http://www.100toplibrary.com/SiteMap
39338	3	1	Directory of chat sites	http://www.100topchat.com/SiteMap
38398	2	1	Directory of humor sites	http://www.100topjoke.com/SiteMap
36792	3	1	Directory of software sites	http://www.100topsoftware.com/SiteMap

document similarity, where two documents are considered to be *virtually identical* if they have all six supershingles in accord, which is indicative of similarity exceeding 99%. Using this metric, 45,492,400 were similar to some other document downloaded that week, while 105,343,809 documents were not similar to any other document. The documents that were similar to others fell into 12,019,235 clusters. In other words, 33,473,165 documents, or 22.2% of all the documents downloaded, were near-duplicates of the 12,019,235 "canonical" documents representing the clusters.

Figure 3 superimposes the distribution of sizes of clusters of very similar documents (black points) upon the distribution of sizes of clusters of virtually identical documents (light orange points). As in figures 1 and 2, the axes are on a logarithmic scale, and the distributions roughly fit a power law. However, the exponent of the distribution of sizes of clusters of virtually identical documents is visibly smaller. Other gross features of the two distributions are similar; e.g. both spot a similar anomaly for cluster sizes around 100.

Table 2 lists the 20 largest clusters of virtually identical documents. The meaning of the columns is identical to those of table 1. As is to be expected, using this more stringent similarity metric eliminates all the "interesting clusters" (that is, those whose constituent documents contain small but meaningful differences). This suggests that search engines that want to exclude all but one web page of each cluster from their index should use this similarity metric when performing the clustering.

Figure 4, 5 and 6 illustrate how clusters grow and shrink

over time. In order to describe the figures, we have to introduce some notation. Let $C_n(u)$ be the cluster of URLs containing URL u in week n. The *cluster containment coefficient* $\Phi_n^m(u)$ of week m in week n for URL u is defined to be:

$$\Phi_n^m(u) = \frac{|C_m(u) \cap C_n(u)|}{|C_n(u)|}$$

Likewise, the *cluster similarity coefficient* $\Psi_n^m(u)$ of weeks m and n for URL u is defined to be:

$$\Psi_n^m(u) = \frac{|C_m(u) \cap C_n(u)|}{|C_m(u) \cup C_n(u)|}$$

We further define U_j^i to be the set of all URLs occurring in a cluster of size between i and j in week 1, i.e.

$$U_i^j = \{u : i \leq |C_1(u)| \leq j\}$$

The curves in these three figures show averages of Φ and Ψ over all the URLs in a set U_i^j. Each curve represents different choices of i and j; that is, it aggregates all the URLs in clusters whose sizes in week 1 were in the range $[i, j]$.

Figure 4 shows the rate at which documents depart from the clusters they occupied in the first week. The figure shows seven curves; each curve represents clusters whose sizes ranged between i and j in week 1. A data point (n, v) on curve i–j indicates that

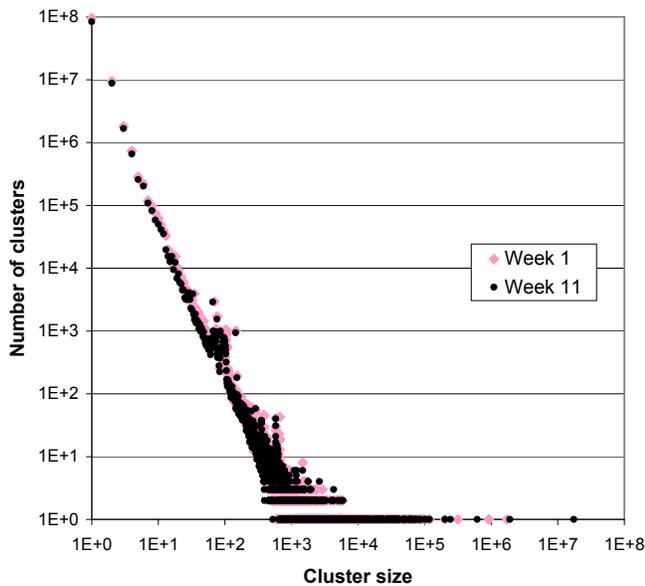$$v = \frac{\sum_{u \in U_i^j} \Phi_1^n(u)}{|U_i^j|}$$

Figure 2. Distribution of sizes of clusters of very similar documents in week 1 vs. week 11

Figure 3. Distribution of sizes of clusters of very similar vs. virtually identical documents in week 1

Figure 4. Containment of week n clusters in week 1 clusters

More colloquially speaking, the data point (n, v) shows that in week n, a fraction v of the URLs that were clustered together in the first week remain in the same cluster. There are a number of features in figure 4 that deserve mention.

First, note that the curve for clusters of size 1 is a straight line at 1. In other words, clusters of size one have perfect retention – not surprising since they contain only one URL, and the document behind the URL will always resemble itself.

Second, the general trendline of the curves representing non-singleton clusters shows a slight downward slope. Clusters of 10 or fewer URLs appear to have better retention than larger clusters, but beyond that, there seems to be little correlation between cluster size and rate of retention. Except for the category of cluster containing more than 100,000 documents (which encompasses only five clusters, as can be seen from table 1), the average retention rate in each category is between 78% and 95%.

Third, there is a prominent dip in retention in the third week, affecting clusters of all sizes (other than singleton clusters). This dip is caused by a disk failure we encountered during our third week of crawling, which caused us to loose all information about URL from a quarter of the hosts in our sample set.[4] Losing information about some URLs may cause clusters to split: Some fraction of the URLs in a cluster will be handled by the machine with the failed disk (and thus move in week 3 into the "could not download" cluster), while those URLs in the cluster that are handled by non-failed crawlers will remain in the original cluster. We also

see a more uniform impact of the crash on retention based on the size of the cluster, because larger clusters are more likely to span multiple hosts.

Fourth and finally, the curve of the clusters containing more than 100,000 documents – which, as mentioned above, contains only five clusters – is far noisier than the curves of smaller clusters. We have two explanations for this phenomenon: First, one of the five clusters originates from two web server machines, another from just three; a transient failure of one of those machines will cause the corresponding cluster to fragment. Second, our observation of the pages in a cluster is drawn out over a span of days (possibly up to a week). Therefore, if the content of the pages in a cluster changes during that week, the cluster appears to have fragmented from our point of view. This effect is amplified by clusters that originate from different web servers if the synchronization of content among those servers is time-delayed.

Figure 5 shows the similarity between each cluster in week 1 and its counterpart in subsequent weeks. As in figure 4, the

---

[4]The Mercator web crawler distributes URLs across crawler machines by hashing the URL's host component; in other words, URLs referring to the same web server are handled by the same crawling machine.
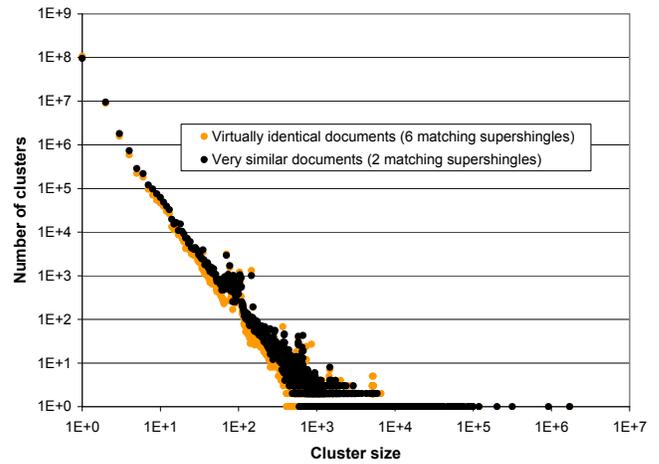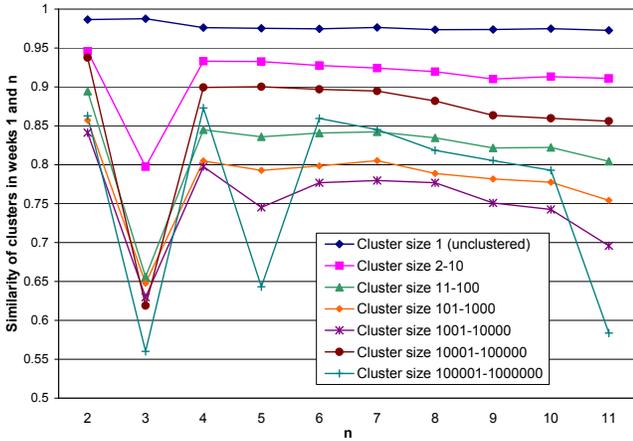
Figure 5. Similarity of week 1 and week n clusters

Figure 6. Containment of week 1 clusters in week n clusters

figure shows seven curves; each curve in this figure represents clusters whose sizes ranged between i and j in week 1. A data point (n, v) on curve $i - j$ indicates that

$$v = \frac{\sum_{u \in U_i^j} \Psi_1^n(u)}{|U_i^j|}$$

More colloquially speaking, the data point (n, v) shows that in week n, what fraction v of the URLs that were clustered together either week were indeed clustered together in both weeks. By definition, these numbers are smaller than those in figure 4 (since $\Psi_m^n(u) <= \Phi_m^n(u)$ for all m, n, u).

Note that the curve of clusters that were of size 1 in the first week is not constant at 1, as it was in figure 4. This is partly due to the fact that URLs that constituted a singleton cluster in the first week might become unavailable in subsequent weeks, and thus move to the large "could not download" cluster.

Despite this, it is remarkable how similar figures 4 and 5 are. This suggests that the evolution of clusters in our sampled set was dominated by URLs becoming unavailable. However, it should be acknowledged that our study was not designed to witness the birth of new web pages, and consequently could not observe any such new pages being added to existing clusters. Performing such a study remains an intriguing avenue for future research.

In an attempt to isolate the arrival rate of URLs into existing clusters, we computed the reverse cluster containment coefficient, as depicted in figure 6. Once again, each curve in this figure represents clusters whose sizes ranged between i and j in week 1. A data point (n, v) on curve $i - j$ indicates that

$$v = \frac{\sum_{u \in U_i^j} \Phi_n^1(u)}{|U_i^j|}$$

As it turns out, this figure does not reveal any interesting arrival statistics, but it does shed light on the retention dip for very large clusters that showed up so prominently in figure 4. We observe a correlated dip in this figure, but the
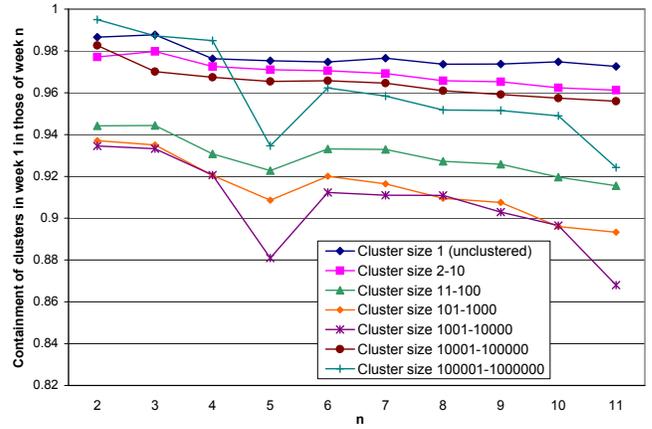
coefficients are much closer to 1. This is inconsistent with the primary explanation being the failure of a web server, suggesting that instead the content of the pages in the cluster has changed in the midst of our observation. Our raw data contains timestamps indicating when we fetched the pages; we are in the process of investigating whether this "version skew" was caused by the temporal spread of our observation, or by slow replication of content among mirrored web sites.

We now turn to considering the relationship between clustering of documents and the amount of change from week to week. Figure 7 shows the correlation between how much a web page changes week over week and the size of the cluster it is contained in. The horizontal axis indicates how many features two successive successful downloads of a URL have in common. A value of 0 indicates that two successive downloads of a URL have no features in common, a value of 84 means that they agree in all features, and a value of 85 means that they are bitwise identical, including the HTML markup. The bars in the graph are divided into seven colored regions, encoding cluster size on a logarithmic scale. The region representing clusters of size 1 (i.e. documents that do not have near-duplicates) dominates. The vertical axis of the graph quantifies this by showing what percentage of documents from any given change-bucket fall belong to clusters of a given size.

Two feature in this figure stand out: First, clustered documents are more likely than unclustered documents to either change by a small amount, or to change almost completely. Documents that are contained in small clusters of ten or fewer documents are disproportionately likely to contribute to either phenomenon. Clustered documents are also more likely to change only in their markup. Second, we note that very large clusters exhibit change almost exclusively by a middling amount, or not at all.

Figure 8 shows this last observation more clearly by normalizing the data of the previous graph. Recall that in figure 7, a bar associated with a cluster size range r (indicated
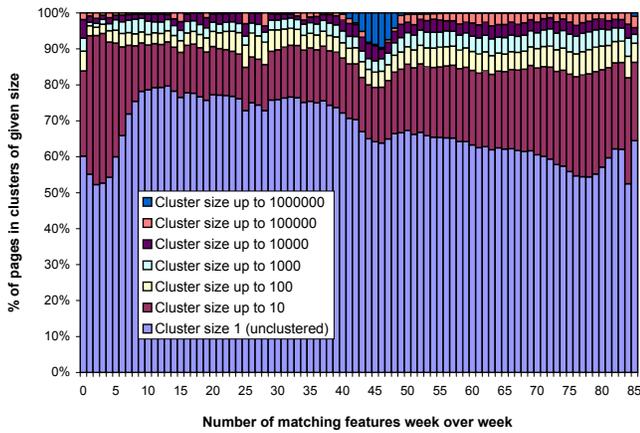
Figure 7. Cumulative distribution by document change amount of cluster size

Figure 8. Relative contribution of cluster size to change rate

Figure 9. Change amount distribution by cluster size

by a color coding) of height y in change bucket x indicates a fraction y of all the documents in change bucket x belong to clusters in cluster size range r. The average height Y of cluster size range r is the average height of all 86 bars associated with cluster size range r. Figure 8 shows $\frac{y}{Y}$ for each of the change buckets and cluster size ranges. the vertical axis is shown on a logarithmic scale. Were it shown on a linear scale, the area below each curve would be identical. Values above 1 indicate change buckets where a given cluster size range is contributing more than its fair share.

The figure illustrates that clusters containing more than 100,000 documents are disproportionately likely to have 40 to 50 unchanged features from week to week, are proportionally likely to not change at all, and are disproportionately unlikely to exhibit any other change rate. Similarly, clusters containing between 10,001 and 100,000 documents are disproportionately like to change in fewer than half of their features.

Figure 9 shows us the absolute distribution of change rate in each cluster size range. Each bar is divided into 6 regions, corresponding to the following six change clusters, from top to bottom: *complete change* (0 common features), *large change* (1-28 common features), *medium change* (29-56 common features), *small change* (57-83 common features), *no text change* (84 common features), and *no change* (subset 85: 84 common features and a common checksum). The height of each region is indicative of the fraction of documents falling into a cluster of the appropriate size, and having a certain week-to-week change. Comparing the heights of the colored bars, we once again see that documents drawn from large clusters are far less likely to change than other documents. Documents in clusters containing more than 100 and at most 10,000 URLs are more likely to change than documents in smaller or larger clusters.
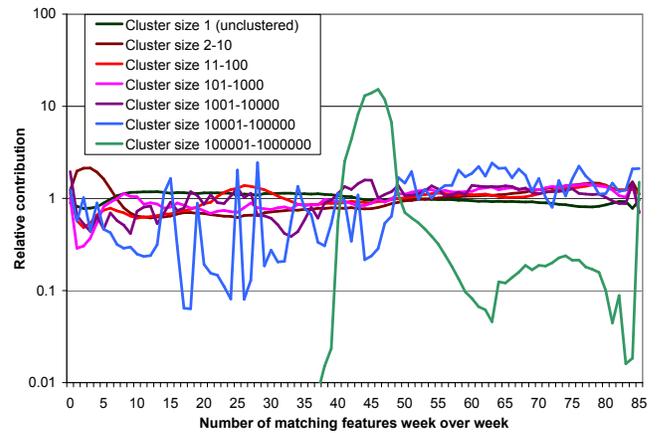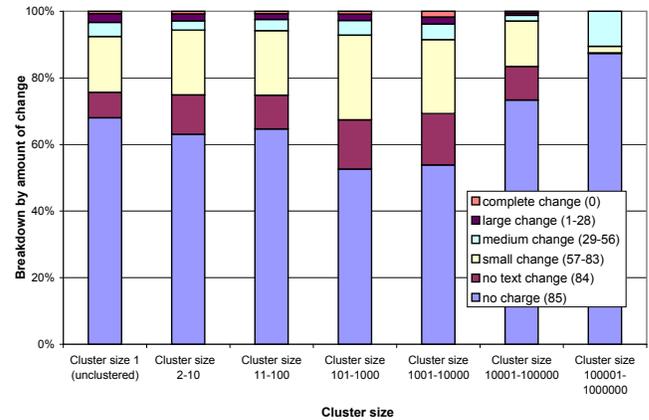
4. Conclusions

This paper describes a large-scale study on the prevalence and evolution of clusters of very similar ("near-duplicate") web pages. It confirms Broder *et al.*'s observation of widespread duplication of web pages. In particular, we found that about 28% of all web pages are duplicates of some pages in the remaining 72%, and 22% are virtually identical. We examined the documents in the 20 largest clusters and categorized them.

The present study also examines the rate at which documents exit clusters, and found that clusters are fairly stable over time; clusters of intermediate size are generally the least stable. This finding has practical importance, since it implies that search engines do not need to perform frequent recrawls of any but one of the web pages in a cluster. Of course, omitting duplicate pages from a crawl will change the observed link structure; namely, the number of in-links of pages referred to by duplicate pages will decrease. It will also affect link-based quality metrics; for example, the PageRank of a page will decrease if we omit duplicate pages referring to

it. Although search engines could compensate for this (for example by adding by adding "phantom links"), they might well decide not to do so, since replicated links can be viewed as multiple, colluding endoresements, which should count for less than independent endorsements. Either way, our techniques afford search engines a choice.

Finally, we investigated the relationship between cluster size and rate and degree of change. We found that documents in large clusters change less than those in smaller ones, and that documents in medium-sized clusters change the most. Combining this observation about medium-sized clusters with our previous observation about medium-sized cluster being the least stable implies that incremental web crawlers will sadly enough not be able to ignore documents in medium-sized clusters.

5. REFERENCES

[1] K. Bharat and A. Broder. Mirror, mirror on the web: a study of host pairs with replicated content. In *Proc. of the 8th International World Wide Web Conference*, May 1999.

[2] K. Bharat, A. Broder, J. Dean and M. Henzinger. A comparison of techniques to find mirrored hosts on the WWW. *Journal of the American Society for Information Science*, 51(12):1114-1122, Dec. 2000.

[3] A. Broder. Some applications of Rabin's fingerprinting method. In R. Capocelli, A. De Santis and U. Vaccaro, editors, *Sequences II: Methods in Communications, Security, and Computer Science*, Springer Verlag, 1993.

[4] A. Broder, S. Glassman, M. Manasse and G. Zweig. Syntactic clustering of the web. In *Proc. of the 6th International World Wide Web Conference*, Apr. 1997.

[5] A. Broder, M. Charikar, A. Frieze, and M. Mitzenmacher. Min-Wise Independent Permutations. *Journal of Computer and System Sciences*, 60(3):630-659, June 2000.

[6] J. Cho, N. Shivakumar and H. Garcia-Molina. Finding replicated web collections. In *Proc. of the 2000 ACM SIGMOD International Conference on Management of Data*, May 2000.

[7] J. Dean and M. Henzinger. Finding related pages in the World Wide Web. In *Proc. of the 8th International World Wide Web Conference*, May 1999.

[8] D. Fetterly, M. Manasse, M. Najork and J. Wiener. A large-scale study of the evolution of web pages. In *Proc. of the 12th International World Wide Web Conference*, May 2003.

[9] N. Heintze. Scalable document fingerprinting. In *Proc. of the 2nd USENIX Workshop on Electronic Commerce*, Nov. 1996.

[10] J. Hopcroft and J. Ullman. Set merging algorithms. In *SIAM Journal on Computing*, 2(4):294-303, 1973.

[11] U. Manber. Finding similar files in a large file system. In *Proc. of the USENIX Winter 1994 Technical Conference*, Jan. 1994.

[12] M. Najork and A. Heydon. High-performance web crawling. SRC Research Report 173, Compaq Systems Research Center, Sep. 2001.

[13] M. Rabin. Fingerprinting by random polynomials. Report TR-15-81, Center for Research in Computing Technology, Harvard University, 1981.