Measuring redundancy level on the Web

Alexander Afanasyev
UCLA
afanasev@cs.ucla.edu

Jiangzhe Wang
UCLA
lucas@cs.ucla.edu

Chunyi Peng
UCLA
chunyip@cs.ucla.edu

Lixia Zhang
UCLA
lixia@cs.ucla.edu

ABSTRACT

This paper tries to estimate redundancy level on the Web by employing information collected from existent search engines. To make measurements feasible, a representative set of Internet sites was collected using a random sampling of the Internet catalogs DMOZ and Delicious. Each page in the set was identified using a random 32-word phrase extracted from the content of the page. These phrases were used to perform search engine queries and infer the number of pages with the same content. Though the presented method is far from being perfectly accurate, it provides an approximation of a lower-bound for visible redundancy of the web—long phrases will likely belong to duplicate pages, and only the pages indexed by search engines are really visible to users. Obtained results showed a surprisingly low level of duplication averaged over all content types, with less then ten duplicates for most of the pages. This indicates that besides well-known classes of high-redundant content (news, mailing list archives, etc.), content duplication and plagiarism are not globally widespread across all types of webpages.

Categories and Subject Descriptors

H.3 [**Information Systems**]: Information storage and retrieval; H.3.3 [**Information Search and Retrieval**]: Clustering—*redundancy measurement*

General Terms

Redundancy, Hidden Web, Visible Web

Keywords

Redundancy measurement, search engine comparison, random sampling, document identification

AINTEC'11, November 9–11, 2011, Bangkok, Thailand.

1. INTRODUCTION

The simplicity of information distribution in electronic form has created a number of problems in various areas. One of the biggest challenges is the increasing complexity of relevant information retrieval due to content multiplication and plagiarism. In particular, a piece of information, being published once on a website, can be duplicated on thousands websites (e.g., a personal blog's entry is reposted on thousands of other blogs). As a result, it can be virtually impossible to track down the original source, which sometimes is essential to discover additional relevant information (new posts of the original author). Because of the increasing commercialization of the Web (i.e., people try to attract more visitors to their websites to get more clicks on pay-per-click links), more people and automated systems are interested in cloning the information from one website to another. Thus, the replication phenomenon can potentially become ubiquitous, and nobody yet knows the degree of its globalization.

We define two webpages to be redundant if the significant portion of the textual content of one page is repeated in the exact form on the other page. Though not fully deterministic, this definition captures the essentials of the information duplication, and, at the same time, allows us to perform a global-scale analysis that otherwise would be impossible.

In this paper we are trying to answer the question of whether the redundancy is a real problem across all types of Web pages or not. To find the answer we first randomly select a set of webpages (based on DMOZ[1] and Delicious[2] catalogs). After that, for each page in the set we discover how many pages on the Web replicate a portion of the content of this page. In this step we solely rely on the existent search engines, such as Google, Bing, and Yahoo, because they have already indexed a large portion of the Internet, and essentially all "visible" Web is just the content indexed by these search engines. This decision limited us to a very small

[1] `http://www.dmoz.org`
[2] `http://www.delicious.com`

portion of page content that we can use for comparison: search engines do not generally answer queries that are more than 32 words long. However, even with such short queries, it is possible to uniquely identify page on the web (e.g., using a search for phrases in quotes). Section 2 provides detailed information about the implementation aspects of our measurements.

In our measurements we used three search engines, Google, Yahoo, and Bing, and compared the obtained results from all of them. As we show in Section 3, this yielded an interesting observation about the relation between Yahoo and Bing search engines. Though it was announced recently that Yahoo is now powered by Bing [10], our results confirm this claim by showing a very high correlation between Yahoo and Bing, unlike the results obtained from Google.

In addition to different search engines, we tried to separate our sampling sets in several different categories: recreation, sports, home, health, computer, food, games, research, culture. However, we explicitly excluded very redundancy-prone categories, like news and mailing lists, as we expected they would be clear outliers in our measurements.

2. METHODOLOGY

To perform our experimental evaluation we implemented three basic components: *sampler*, *phrase extractor*, and *querier* (Figure 1). In this section we present in detail each of the implemented components.

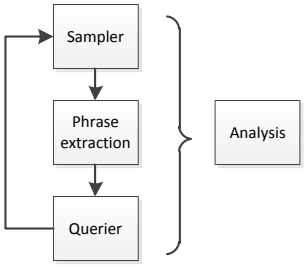

Figure 1: Main components of redundancy measurements

2.1 The sampler

On a high level, the design of the sampler is practically identical to a common search crawler. The sampler obtains a list of initial start pages, downloads the content of the pages, extracts links, adds the extracted list to a download queue, and recursively repeats download–add steps until a predetermined condition (e.g., a limit of the total number of downloaded pages) is reached. However, there are two important differences between functionality of the crawler and the sampler.

First, the sampler does not simply add all of the extracted links to the download queue (excluding various search engine policies and exclusions in `robots.txt` files), but uses a random algorithm to choose a link for further processing.

Second, the sampler has important restrictions on types of start pages. For example, if the task is to get a sample of links related to news, CNN's website will not be a good starting point, because it will generally provide links to CNN articles. As a result, no matter which random algorithm we choose, we will not receive a representative sampling set of all available news articles on the Web. One of the best starting points in the news example is a news aggregator. If it provides many links to a wide range of news websites and news articles, by randomly choosing a portion of them, we will get a set of high-diversity links.

We chose the simplest, but yet powerful random algorithm to select links for the sampling set:

1. Select the first link from the download queue and shift the queue.
2. Download page content for this link.
3. Extract all links from the page and add them to the download queue.
4. For each extracted link we throw a dice: If the random value is less than a predefined threshold, we add the link to the sampling set.
5. If the number of elements in the sampling set is less than a predefined threshold and there are links in download queue, start from step #1.

We created several sampling sets based on different topics of our choice. To approximate topic separation we performed sampling of a DMOZ on-line catalog starting from pages the correspond to different categories. The advantage of using such a catalog is twofold. First, by definition this catalog contains a diversity of links. Second, the links in the catalog are moderated, which limits the number of spam sites in our sampling set. We configured our sampler to pick 2% of the discovered links, which provides a good enough approximation of pure random links in particular category from DMOZ.

Another source of the high-diversity links that we used is the on-line bookmark service, Delicious. This service provides users with an ability to save their bookmarked links online, and at the same time gives all other users the ability to browse all these links. Because a user's choice to make a bookmark can be considered a random process, there is no real need to perform any serious randomization when picking links while crawling. Nevertheless, we decided to introduce a small level of randomness by picking 80% of the discovered links to mitigate the effect of sequentially bookmarked links.

Figure 2 illustrates the diversity of links in our sampling sets. The image show a mapping of the top-level

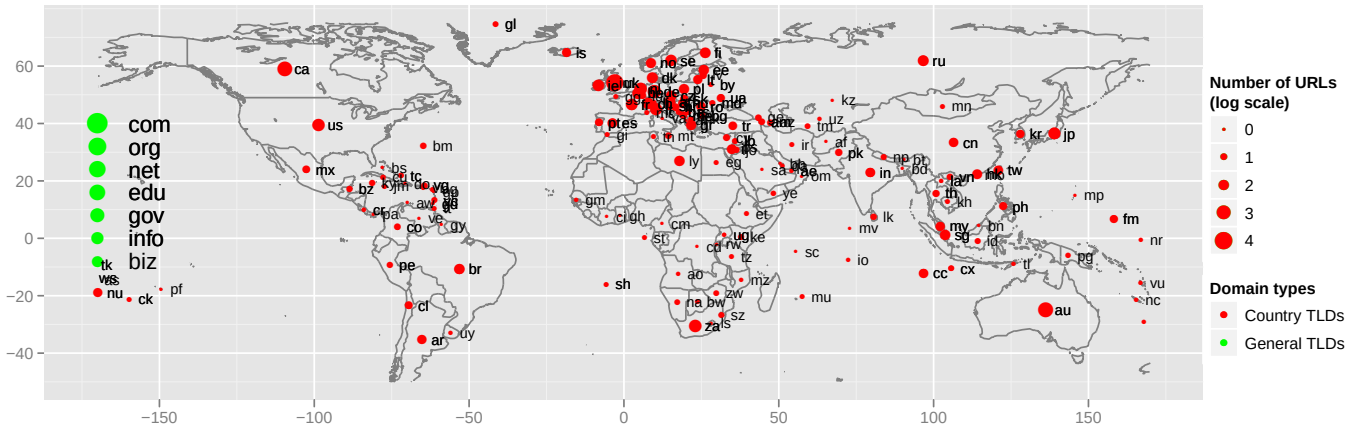

Figure 2: Coverage of country top level domain names (TLDs) in the sampling sets

domain names present in sampled URLs. It is clear that we have a high variety of domain names (practically all country TLDs are present) and domain name frequencies reflect, to some degree, penetration of the Internet in the world [2], assuming that **.com** domains are attributed mostly to the United States. This observation shows that our sampling sets have a good degree of world-wide representativeness.

2.2 The phrase extractor

Another crucial component of our system is the phrase extractor that, for each webpage, forms a set of phrases to uniquely identify (or partially identify) this page among all others in a search engine. However, in a raw form, this textual information is not very useful.

2.2.1 *Text reduction*

Textual representation of navigation links, copyright notices, and other similar information is an essential part of HTML, but has nothing to do with our redundancy measurements. Running HTML-to-text conversion on various webpages allowed us to develop two simple heuristics that significantly improve the quality of the phrase extractor.

The first heuristic eliminates all textual information that came from the separately visualized HTML tags (such as `<div>`, `<p>`, etc.) and have less than ten words.

The second heuristic eliminates from the reduced textual representation all sentences that are less than ten words long. To extract sentences from the text we rely on MorphAdorner Java library [1], which is open-source and implements "intelligent" algorithms for sentence splitting. To make sentence splitting more predictable, we consider all separately visualized HTML tags as complete sentences.

2.2.2 *Identification in search index*

Even after elimination of short phrases, textual representation of web pages still contains a lot of text, ranging from tens to thousands of words. Because our objective is to use a search engine to identify a set of pages with the same (or partially the same) content, we are able to use only a limited number of words in a query (e.g., Google allows searches on phrases that are up to 32 words).

Our way to solve the page identification problem is to randomly select up to 32 consecutive words from the page and perform an exact phrase search using only these words. (Note that if in the original text there are three consecutive sentences, and our short-sentence-elimination heuristic removes the second sentence, we will select two phrases, separately, the first and the third sentence.) Unfortunately, in some cases even 32 consecutive words may not be able to uniquely identify the content. For example, if the random selection is unlucky enough to select a copyright notice, a search engine will return a volume of results that have no meaning for our redundancy measurements. Although this fact presents a level of uncertainty in our results, it does not largely affects overall results.

Another potential problem arises from different interpretations of special characters. Initial experiments revealed that Google ignores all periods, commas, question and exclamation marks, colons, semicolons, and brackets, but treats "&" in phrases as a word. To deal with this problem we adapted our phrase selection algorithm accordingly and manually verified correctness of the algorithm on small subset of generated queries.

2.3 The querier

After extracting phrases from webpages in the sampling set, we made the decision to obtain potential duplicate webpages by querying the phrases against a search engine. There are two reasons for this choice. First, it saves us time and effort compared to direct crawling and building of a large webpage index from scratch.

Second, it is much easier to convert our measurement results into a practical use in the future. More specifically, if the results show that the duplication level is extremely high for a certain set of topics, we could share our result with commercial search engines and possibly contribute to a better and simplified search result. Last, if our measurements were performed on a set of webpages that were not indexed by any search engine (i.e., the "hidden" web), our measurement result would not be practically valuable. Because we chose publicly available links from DMOZ and Delicious catalogs, the links in our sampling sets are guaranteed to be included in search indices (with minor exception of pages banned by search engines).

We now illustrate the details of retrieving potential duplicate pages from Google, Yahoo, and Bing. Google used to provide a search API for use by third-party applications. However, for reason unknown they discontinued their previous generation API, but did not fully open a new generation API (AJAX API). For this reason we decided to use the generally available HTML search interface for Google. To search for "Isaac Newton" in Google, the GET request URL would be `http://www.google.com/search?q="Isaac+Newton"`.

After receiving a query result page, the task is to extract (1) the estimated number of results and (2) target links (a target link here means a URL that identifies a query result). We use the HTMLParser library [11] to identify both components using CSS selectors (`DIV #resultStats` and `A.l` for Google). Also, it is possible to get a whole set (up to the first 1000) of the links returned by Google using additional GET requests with specified a "start" parameter. For instance, `start=20` returns query results between the 20th and the 29th. For Yahoo, we adapted the querier implementation for Google that uses different search URLs and CSS selectors. For Bing, we registered our application and obtained a Bing API key that allowed us to obtain results for the queries in XML format.

Unfortunately, search engines (and Google in particular) do not like being automatically queried and impose a variety of limitations and blocks. For example, if a computer issues queries with high frequency, Google blocks its IP address and presents a reCAPTCHA code to validate human involvement in these queries. In other words, it is virtually impossible to perform many queries (and technically it goes against the terms-of-use for search engines). So, we abandoned the idea of extra queries per phrase and based our results solely on the search engine's estimate.

To tackle the search engine limitation problem we deployed the querier on a cluster of separate machines located in a number of different networks. For each machine we performed one query in ten seconds for each search engine. Even with such low-rate queries, Google and Yahoo were temporarily (for 24 hours) blocking our computers from the search engine. In total, we were able to perform and analyze about 100,000 queries for each search engine; results of this analysis are presented in the Section 3.

3. MEASUREMENT RESULTS

In our analyses we eliminated clear outliers that resulted from imperfections in our random phrase extraction process. In particular, we invalidated all records in the database that correspond to empty result sets (due to our link discovery process, a page has to be present in the search indices; see Section 2.1). Also, we excluded all high-frequency results (result set that larger than 1,000,000), because they are likely to have been generated by a very common (e.g., copyright notice) or invalid phrase extracted from the page.

3.1 Power-law distribution of redundancy

Our first finding is that the most of the phrase queries across all search engines resulted in a very small number of search results. As can be seen from Figure 3, over 86% Google and over 97% Yahoo and Bing queries yielded the number of results in a range from 1 to 60.

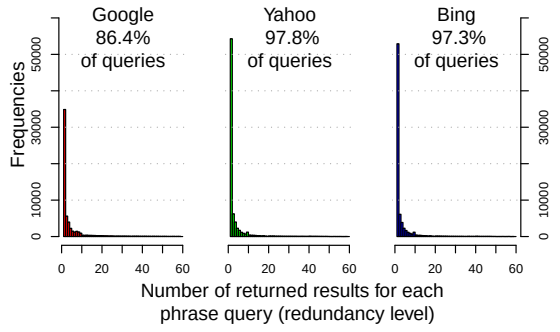

Figure 3: Distributions of page redundancies for Google, Yahoo, and Bing

More interestingly, a largest component of these graphs corresponds to queries that resulted in exactly one search result. If we look at the same results in log-log scale (Figure 4), we can see that similar behavior applies not only to the initial area (say, pages with a redundancy level from 1 to 60), but also to any other region. In other words, the distribution of page redundancies loosely follows a power-law distribution.

The observation that we make from these log-log plots is that results from Google follow a power law distribution across all redundancy levels, while Yahoo and Bing show a sharper distribution for "small" redundancy levels (from 1 to 10^2), but after that they express almost a uniform distribution. We have several potential explanations for this behavior. First, it may indicate that Yahoo/Bing more effectively eliminate duplicate/similar

pages from the result sets. Second, it also may mean that Google makes more optimistic predictions about the result set, while Yahoo and Bing employ slightly more pessimistic strategies. Third, Google may be less restrictive (more aggressive) to page indexing, resulting in a bigger search index, and thus making more pages available during the query resolving process. Finally, differences between search engines could be due to different interpretations of phrase queries. We have not fully investigated this possibility, but in some cases search engines ignored our exact phrase searches and presented us with results for a keyword search. If this is the case, then the results for large redundancy levels could be eliminated. However, even with such elimination, our initial observation (that a large number of queries results in a very small number of pages) still holds.

To understand more of the behavior of redundancy distribution in the initial region (e.g., for redundancy levels from one to ten) we built a cumulative distribution function (CDF) of page redundancies in log-log scale (Figure 6). From this figure we can easily obtain the percentage of queries (i.e., pages) that resulted in a defined number of results (i.e., duplicate pages). On the graph, we marked the most interesting region—the initial region with redundancy levels from one to ten. In addition to our previous observation (Figure 3) that most of the pages have no more than 60 redundant pages, now we see that the same can be concluded even for a tighter region. According to CDF, Google in 75% and Yahoo and Bing in 90% of the cases gave us sets containing up to ten results.

We have already seen that the major redundancy component corresponds to pages without redundancy at all. Using CDF we can say that in our experimental evaluation about 40% of the pages belong to this category. The next biggest component, resulting in 50% and 70% for Google and Yahoo/Bing, corresponds to pages with a redundancy level up to two. In a large number of cases such behavior can be explained by the fact that the search engines consider pages from different domain names to be different pages; but most of the time, pages with and without the `www.` prefix point to the same content.

A manual analysis of several queries with a redundancy level from three to ten, gave us some understanding of one of the potential sources of redundancy. There are many cases when some important part of the page content (e.g., detailed information about a company) is duplicated several times within a single site, as well as on various different sites. For example, the queries made to Yahoo *"Mirage Systems Inc. is very different than other container manufacturers" "At Mirage we do one thing and we do it very well" "We produce high quality harness / container"* resulted in six pages. Two of them belong to the company Web-site `www.miragesys.com` (the main and "about us" page present the same paragraph with short information about the company); two belong to the people directory website `www.zoominfo.com`; and the remaining two belong to other websites with the same company description. This example shows a very natural way for people to duplicate information. However, it would be very useful to be able to discover the initial source (and potentially distribution path) for the same piece of information. Such a feature may not be very interesting for small result sets described in the example above, but may greatly help to reorder output results based on the distribution path length from the original source.

3.2 Redundancy levels for different topics

To understand the difference in redundancy levels across different topics, we build a collection of histograms in log-log scale for each pair of search engine and category (Figure 5). We can see, results for each category are consistent across all evaluated search engines relative to other categories. Results from Bing/Yahoo almost duplicate each other, indicating a close relation between the search engines. These results confirm the announcement [10] that since August 2010 Bing powers the Yahoo search engine. During our evaluations we were unaware of this announcement, and the nearly identical behavior of Yahoo and Bing was very suspicious. Though results from Yahoo and Bing are very close to each other, they still have a large variance. This may be a result of a slight randomization (geographical or topological proximity, user preferences, etc.) while query processing. At the same time, this may indicate that Yahoo and Bing are still two separate search engines with separate data centers, but use the same algorithms to crawl and process queries.

The redundancy levels for each category presented on Figure 5 are consistent with overall statistics for each search engine (Figure 4). There is an almost ideal power law distribution in the results from Google, and a sharper power law distribution in the initial redundancy zone ($1\text{-}10^2$) for Yahoo/Bing. One interesting observation that can be made from these graphs is that category "Recreation" stands out among all other categories. This category includes a variety of topics (audio, autos, aviation, birding, boating, bowling, camps, climbing, collecting, crafts, drugs, fireworks, fishing, food, gambling, games, gardening, and many others[3]), and we have not fully investigated which one of this subcategories may have influenced our results the most. Because we used an Internet catalog structure to categorize links, we have considerable imperfections in such division. In particular, the category "Sports" is included in the category "Recreation."

[3]All categories from `http://www.dmoz.org/Recreation/`

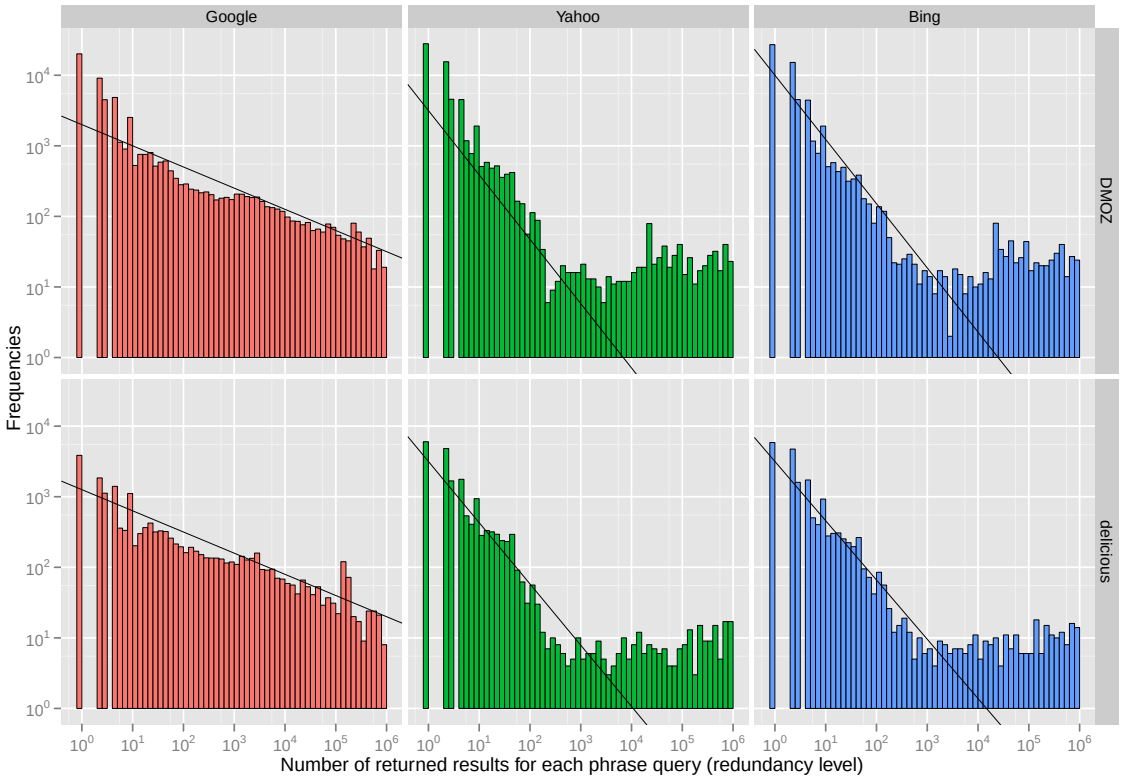

Figure 4: Distributions of page redundancies for Google, Yahoo, and Bing for each sampling set

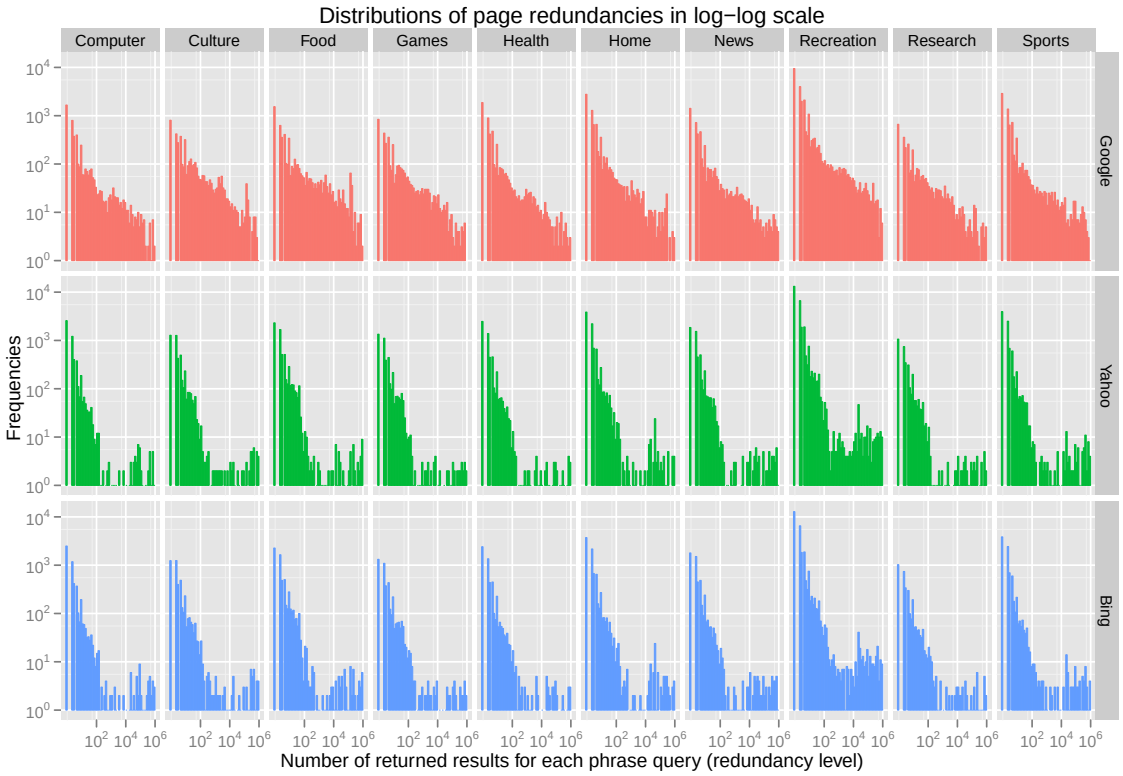

Figure 5: Distribution of page redundancies in log-log scale by engine and category

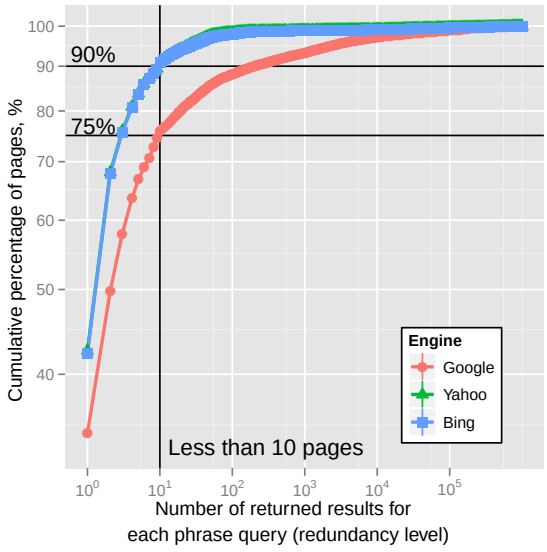

Figure 6: Cumulative distribution function (CDF) of page redundancies in log-log scale

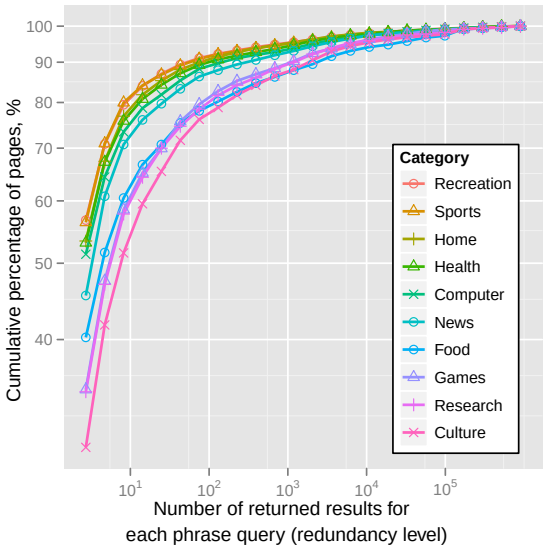

Figure 7: CDF of page redundancies by category in log-log scale for Google

To see relations between different categories more clearly, we plot the CDF of page redundancies by category in log-log scale. Because the results for each search engine have practically the same relative relations between categories, we present only one graph that corresponds to Google (Figure 7). This graph also shows that the category "Recreation" has larger redundancy levels compared to all other categories. In addition, it also shows that the second high-redundancy category is "Sport." Another interesting observation concerns the other end of the scale. The less redundant pages belong to the category "Culture" (URLs sampled from the Delicious catalog using tags *culture+blog*, *culture+ magazine*, *culture+ inspiration*, *culture+ design*, *culture+ art*, *culture+ writing*, *culture+ technology*, *culture+ music*, *culture+ uk*, *culture+ movies*, and *culture+ development*). This means that the content on culture-related pages is considerably less "interesting" than the content on recreation/sports pages. This reflects the general fact that there are a lot of fans who will be happy to copy information about their teams to other websites, and that the audience of culture-related websites are not likely to do the same thing.

Our measurements show that while there is a high replication level in some areas (such as areas corresponding to the "Recreational" category and highly likely in news/mailing list categories that we have eliminated from our measurements), overall the replication cannot be considered a serious problem yet. In about 40% of cases the content uniquely identifies the source, and in more than 80% of cases, a query based on a random phrase from the page yields a very small result set (1-60), which can easily be processed by a human being. However, identification of the original source and information flow (who copied and from where copied, or just from where) would be very useful for arranging results in high-redundancy cases (e.g., for popular categories).

4. RELATED WORK

There is a large amount of existing work that deals with redundancy. Redundancy has been defined in various ways, based on textual information, internal structure, hashes of the content, external (link) structure, etc.

Brin et al. [3] defined similarity as a significant overlap in the exact sentences (e.g., two documents are considered related if they contain more than 20% of the same sentences). Shivakumar and Garcia-Molina [12] employed a vector-space model of the documents and terms to compute an asymmetric document closeness measure. Both of these approaches require access to the full content of each page, which results in storage and computation scalability issues when applied to the entire body of Web pages.

To resolve these issues, other approaches tried to use content structure. Among them, the most popular (used by several search engines) is the shingling algorithm [4]. It retrieves every available document, calculates short syntactic sketches for each document, and then pairwise compares these sketches to all the documents in the set. The main purpose of this algorithm is to cluster similar documents in big collections. A similar idea was applied by Lin and Ho [9] in the design of InfoDiscoverer. On a high level, it partitions pages into several content blocks according to `<table>` HTML tags and tries

to identify the informative value (good or redundant) of each content block using a calculation of entropies based on the occurrence of the terms (features) in the set of pages. This work mainly focuses on finding the redundant parts in a webpage, such as identifying the informative blocks from redundant blocks (like advertisements, banners, navigation panels).

To reduce computation load several proposed algorithms apply hash functions to calculate page similarities. Haveliwala et al. [6] explored a locality-sensitive hashing technique, which, for each webpage, produces a hash that has high a probability of being in a collision with hashes of similar webpages. Similarly, Charikar [5] constructed new locality-sensitive hashing schemes using rounding algorithms. Although such techniques provide faster processing, they still require the whole set of documents to perform actual clustering, which is not realistic for our measurements.

There is also some work on utilizing external links and other information to define page similarity. For example, Hou et al. [8] utilized hyperlink transitivity and page importance to measure webpage similarity.

One of these algorithms could be an ideal tool for finding pages with similar content, and thus estimate redundancy. Unfortunately, it is virtually impossible to apply any of these algorithms to the whole Internet; all of them require (at least once) downloading all the documents. Instead, we employed a more realistic and small-scale approach, where we crawled a small portion of randomly sampled pages and then discovered those which may duplicate the sampled pages using popular search engines through the query interface.

To the best of our knowledge, there is not much work that measures redundancy level on the real Web. The only work that we are aware of is that conducted by Henzinger [7]. He compared the performance of two "state-of-the-art" algorithms developed by popular search engines—Broder's shingling algorithm [4] and Charikar's random projection-based approach [5])—using a large number (about 1.6B) of web pages. The results show that neither of the algorithms work well for finding near-duplicate pairs on the same sites, while both achieve high precision for near-duplicate pairs on different sites. The measurement work by Hezinger focuses on performance evaluation of the algorithms, whereas our measurements aim to understand characteristics of the Web itself, i.e., redundancy level on the Web.

5. CONCLUSION

In this work we randomly sampled more than 100,000 links from two Internet services DMOZ and Delicious, (link directory and bookmark service), and from ten different categories: recreation, sports, home, health, computer, news, food, games, research, culture. Our measurements are based on 100,000 exact-phrase queries for each of the most popular search engines (Google, Yahoo, and Bing). The results showed that information from most of webpages is not replicated at all (i.e., redundancy equal to one) or duplicated on a very limited number of other webpages. Though replication level depends on the content type (the recreational/sports category has a much larger replication than the culture category), the redundancy distribution has similar overall characteristics: majority of pages are with minimal or no replication and there is a long tail of highly redundant pages (power-law-like distribution). This long tail is attributed partially to the imperfections of our technique, and partially to the natural way of information distribution.

6. REFERENCES

[1] Academic and Research Technologies, Northwestern University. MorphAdorner. `http://morphadorner.northwestern.edu/`, 2009.

[2] A. Afanasyev, N. Tilley, B. Longstaff, and L. Zhang. BGP routing table: Trends and challenges. In *Proc. of High Technologies and Intellectual System conference*, 2010.

[3] S. Brin, J. Davis, and H. Garcia-Molina. Copy detection mechanisms for digital documents. In *Proc. of SIGMOD*. ACM, 1995.

[4] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the Web. In *Proc. of WWW6*, 1997.

[5] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proc. of STOC*. ACM, 2002.

[6] T. Haveliwala, A. Gionis, and P. Indyk. Scalable techniques for clustering the web. In *WebDB (Informal Proceedings)*, volume 129, 2000.

[7] M. Henzinger. Finding near-duplicate web pages: a large-scale evaluation of algorithms. In *Proc. of SIGIR*, 2006.

[8] J. Hou and Y. Zhang. Utilizing hyperlink transitivity to improve web page clustering. In *Proc. of 14th Australasian database conference*, 2003.

[9] S.-H. Lin and J.-M. Ho. Discovering informative content blocks from Web documents. In *Proc. of SIGKDD*, pages 588–593, 2002.

[10] A. Ostrow. Bing now powers yahoo search. Online: `http://mashable.com/2010/08/24/bing-powers-yahoo-search/`, August 24, 2010.

[11] D. Oswald, S. Raha, I. Macfarlane, and D. Walters. HTML Parser. `http://htmlparser.sourceforge.net/`, 2006.

[12] N. Shivakumar and H. Garcia-Molina. SCAM: A copy detection mechanism for digital documents. In *Proc. of DL*, 1995.