The Volume and Evolution of Web Page Templates

(Confidential Submission to WWW 2005 for PC Evaluation Only)

David Gibson
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120.
davgib@us.ibm.com

Kunal Punera
Dept. of Electrical and Computer Engineering
University of Texas at Austin
Austin, TX 78751
kunal@lans.ece.utexas.edu

Andrew Tomkins
IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120.
tomkins@us.ibm.com

ABSTRACT

Web pages contain a combination of unique content and *template material*, which is present across multiple pages and used primarily for formatting, navigation, and branding. We study the nature, evolution, and prevalence of these templates on the web. As part of this work, we develop new randomized algorithms for template extraction that perform approximately twenty times faster than existing approaches with similar quality. Our results show that 40–50% of the content on the web is template content. Over the last eight years, the fraction of template content has doubled, and the growth shows no sign of abating. Text, links, and total HTML bytes within templates are all growing as a fraction of total content at a rate of between 6 and 8% per year. We discuss the deleterious implications of this growth for information retrieval and ranking, classification, and link analysis.

General Terms

Algorithms, Experimentation, Measurements

Keywords

templates, data mining, web mining, data cleaning, algorithms, boilerplate

1. INTRODUCTION

Template material is common content or formatting that appears on multiple pages of a site. Almost all pages on the web today contain template material to a greater or lesser extent. Common examples include navigation sidebars containing links along the left or right side of the page; corporate logos that appear in a uniform location on all pages; standard background colors or styles; headers or dropdown menus along the top with links to products, locations, and contact information; banner advertisements; and footers containing links to homepages or copyright information. The template mechanism is used to support many purposes, particularly navigation, presentation, and branding.

There is no single dominant mechanism by which templates appear in web pages. At one extreme is web site design software that allows a user to single-handedly manage a medium-size web site, formally editing and applying templates to groups of pages as necessary. At the other extreme is the personal web site in which the owner copies the same fragment of HTML from one page to the next in order to provide a uniform look and feel, and diligently avoids the overhead of changing templates too frequently. Other familiar mechanisms include application servers that implement page templates in code; dynamically generated pages that wrap content into a template; portal servers that arrange content into cells with arbitrary content around them; and content management systems that manage templates.

On today's web, templates are a significant cause for concern. As we show below, templates are responsible for roughly 40–50% of the content on the web. The repeated occurrence across a website of content purporting to be original misleads search engines, page classification, clustering, link analysis, and other applications providing advanced text analysis on web content. Furthermore, an accurate assessment of whether the content of a page has changed is critical in several applications. First, crawlers may behave more efficiently based on knowledge of the change rate of pages. Second, alerting applications should not alert users due to template changes. And third, any applications that support trending over web data should not be misled into believing that a site has changed significantly due to a template modification.

Further, we show that the proportion of templated text on the web has been growing consistently for nearly a decade, and thus all these applications will need even greater awareness of templates in coming years.

On the other hand, effectively recognizing templates brings several advantages. Once extracted, they can be used to identify key pages on a website, such as the products page of a company, or the entry point or each school of a university. Pages that share a template can also be grouped together into a cluster that may not be apparent using other mechanisms. Finally, once templates have been identified, any analysis algorithms can realise a nearly two-fold improvement in storage and processing requirements, by exploiting redundancy.

Unfortunately, no simple and completely effective algorithm for template extraction is known. Techniques for the problem fall into two families. *Local* techniques operate on an individual page without reference to other pages, while *global* techniques consider a family of pages together and exploit the property that templates occur many times. Purely local techniques are effective at stripping away certain kinds of banners and navigational material, but these techniques

WWW2005, May 10–14, 2005, Chiba, Japan.
.

are only heuristics and are somewhat error-prone as the web changes. It is quite common, for example, for certain paragraphs of textual content in the middle of a page to be templates—detecting this without reference to global information is essentially impossible. Thus, effective template detection requires global techniques. We perform our study via two global template detection algorithms.

1.1 Our Contributions

Our contributions are primarily focused on measuring the nature and extent of templating on the web. However, in order to perform this task, we also develop efficient algorithms for template extraction which are appropriate for integration into the workflow of a traditional large-scale web crawler. These algorithms are simple to implement, and allow detection and removal of templates "on-the-fly", using very little memory footprint per site.

We report on two studies. In the first, we randomly sample two hundred sites from a large crawl containing approximately fifty million sites and two billion pages. We hand-classify this site-level sample into seven categories such as personal sites, catalog sites, community sites and so on. We then analyze the nature and prevalence of templates within sites belonging to each category. For each site, we create a uniform random sample of the crawled content from the site of approximately two hundred pages, and study the commonality of templating across this sample; thus, our study captures only templates that occur on some small fraction of the pages on a site. In this context, we present a novel visualization technique which effectively captures the template structure of an entire site.

In our second study, we consider the evolution of template usage. Using crawls from the Internet Archive [5], we study multiple snapshots of pages from two collections: the hand-classified sites from our first study, and the sites studied by Ntoulas et al. [9]. We gathered approximately 72K page instances during this study, over 1380 snapshots of time. We show how template usage has changed over the last 8 years, and offer some thoughts about what these studies portend for the future.

Our primary conclusions are the following:

Template volume: According to our studies, the volume of templated material is 40–50% of the total bytes on the web, and the growth shows no signs of slowing. This has several implications. First, as bandwidth, tooling, and browser capabilities increase, there will be ever more complex and costly templates attached to pages. As browsing patterns show, users tend to visit multiple pages when they visit a site, suggesting that a more sophisticated approach to client-side template caching would be in order. Second, for organizations like search engines and archives that store significant amounts of web content, our results suggest that documents organized by site and compressed schemes that allow implicit or explicit references to site-level templates will show significant savings. Finally, analytical operations in a domain with inline template extraction will run twice as fast as they do today, rather than spending half their time re-processing the same template bytes.

Template types: Different types of web sites show very different templating behavior. While media sites are often presented as examples of aggressive templating, and there are high-profile examples of such sites, we show that the average media site (typically small local media outlets) in fact use templates far less frequently than the rest of the web. On the other hand, catalog sites dedicated to presenting items favorably to consumers offer templates covering 60% or more of their content.

Rate of change: We show that the duration of the average template is quite similar to the duration of the detemplated region of the page. We also consider changed pages, and study the distribution of the magnitude of change. We show that this distribution is weighted more heavily towards the tail (i.e., changes of large magnitude) once the template content has been removed, suggesting that pages do in fact change more or less completely with significant frequency.

Links, text, and html: The fraction of HTML content, and hyperlinks, that appear in templates are comparable, ranging from about 35% to about 50% over a number of data sets. The fraction of detagged content in templates shows a somewhat broader range, as low as 24% in some cases to as high as 53% in others. The rates of change of these three quantities range from 6 to 8% per year. There are two key implications. First, the graph structure of the web is increasingly dominated by boilerplate, suggesting that link analysis algorithms require an understanding of templates. Second, all categories of template usage are on the rise, suggesting that navigation, layout, and publishing of textual content via templates are all important and growing tools in the toolbox of web designers.

1.2 Roadmap

The remainder of the paper proceeds as follows. Section 2 presents related work, and Section 3 gives our algorithms. Section 4 shows the results of analyzing a random sample of web sites from a large crawl, breaks the sites into categories, and presents findings regarding the structure of templates for each category. Section 5 presents a visualization that captures the template structure of an entire site, and gives examples from the categories of the previous section. Section 6 gives the results of an experiment crawling multiple copies of a set of pages from the Internet Archive [5] in order to assess the evolution of templates. Finally, Section 7 gives conclusions.

2. RELATED WORK

The problem of extracting templates from web pages was first introduced by Bar-Yossef and Rajagopalan [1]. They propose a technique based on segmentation of the DOM tree and selection of certain key nodes using properties of the content of the node (such as the number of links within the node) as candidate templates. Yi et al. [12] and Yi and Liu [11] study template extraction in order to improve data mining results by removing noisy features due to templates. They present a data structure called the *style tree* which takes into account certain metadata about each node of the DOM tree, rather than the particular content of the node.

Local algorithms based on machine learning have been proposed to remove certain types of template material. Davison [4] uses decision tree learning to remove "nepotistic" links, which are not present for a valid navigational purpose, and Kushmerick [7] introduces AdEater, a browsing assistant that learns to automatically removes banner advertisements from pages.

There are various approaches to understanding the relative merits of different parts of web pages that address problems raised by the presence of templates and related phenomena. Kao et al. [6] propose a scheme based on information entropy to focus on the links and pages that are most information-rich, hopefully downgrading template material in the process. Song et al. [10] carefully decompose web pages using features of the layout into blocks, and then judge the quality and salience of the blocks in order to rate their importance.

Finally, our approach to detection of templates is related to string matching. The techniques of Broder et al. [2] cluster together documents with sigificant overlap. Edit distance [8] and related string matching operations such as the longest common subsequence of multiple documents may be taken as motivational algorithms for our proposed techniques for syntactic matching of text templates.

3. ALGORITHMS

The algorithms presented in this section have been developed in aid of our measurement efforts; they have been tested to determine that they accurately discover templates across pages on a site, but it is not our intention to provide a formal comparison of template detection algorithms. For validation, we do report results comparing the algorithms on a sample of pages.

We consider two algorithms, one based on the DOM structure of the web page, and the other based on syntactic sequences of characters. DOM-based algorithms provide efficient representations (as a typical page may contain 10-20K of content but only around 100 DOM nodes), and perform well on hierarchical templating schemes using table layouts. Text-based algorithms, on the other hand, are amenable to a class of probabilistic speedups, and perform well in jsp-style templates, as the material in the template need not correspond strictly to the DOM tree.

3.1 DOM-based algorithm

This algorithm uses the DOM structure of the pages on a website by searching for nodes of the DOM tree that are repeated across multiple pages on the website. It is based on the work of Rajagopalan and Bar-Yossef [1], and Yi and Liu [11], but contains simplifications from those techniques.

Construction of the DOM tree for a page requires that the page first be cleaned. This is a substantial problem on the Web due to the diverse set of languages, authors, and tools; and also due to the excellent efforts of web browsers to render badly-formed HTML correctly. We modified an existing HTML parsing and cleaning library called HyParSuite [3] to address this problem, maintaining offsets to nodes in the original unclean page so that the links and text inside and outside templates may be extracted later. The algorithm then operates in two passes.

First pass: The first pass iterates over all the pages in the website and dumps information about all the DOM nodes in a page. This information consists of the hash of the content of the node (template-hash) and the start and end offsets into the original file. The template-hash is calculated using the HTML content within the node's start and end tags and DOM node's name, attributes, and their values. For example, consider the following HTML substructure:

```
<td><a href='...'>Click here</a> to visit ...</td>
```

This structure consists of four HTML nodes. The topmost node is the `<td>` node. The template-hash of this node will be computed from the entire HTML string. The `<a>` tag is a child of the `<td>` node and its template-hash will be calculated using the the contents between the `<a>` and `</a>` tags inclusive of the tags. Text nodes are constructed for stretches of text in HTML files and the above example consists of two text nodes.

Thus the template-hash is a compressed representation of the HTML tag and its contents. Counting the number of times a template-hash is encountered in a website tells us the number of times a specific HTML node is seen. Hence, the first pass keeps track of the number of times each template-hash has been seen in the website and passes this information to the second pass.

Second pass: The second pass then scans this information and computes a set of template-nodes for each page. A HTML node in a particular page is said to be a template-node if the following conditions are met: first, the occurrence count of the node's template-hash is within a specified threshold; and second, the node is not a child of any other template-node.

Sibling template nodes are then coalesced to produce the templates on a page. The coalescing process permits small gaps of changing content in the final templates produced. This is useful for templates with dynamic content, where small portions of the template content changes while the essential HTML and text structure remains the same.

Parameter settings: The DOM-based algorithm is parameterized by the upper and lower thresholds on the number of occurrences of template-nodes. A lower-threshold value of 1 will cause the entire web page to be regarded as a single template, as the root of the page always occurs at least once. The upper-threshold parameter prevents the algorithm from detecting extremely small HTML constructs like `<BR>` as templates just because they are fairly common in HTML files. Other than removing small commonly-occurring HTML nodes from consideration, the upper-threshold does not have significant impact on the quality of templates detected.

For the experiments reported below, the lower threshold is set to 10% of the number of pages scanned on each site, while the upper threshold is set conservatively to the full number of pages scanned, since the volume of small templates detected does not contribute significantly to the overall proportions. 200 pages were scanned per site. The processing runs at an average of 17.5 seconds per site on a 2.4GHz Pentium IV machine.

3.2 Text-based algorithm

The text-based algorithm does not make use of HTML structural information. The page is pre-processed to remove all HTML tags, comments, and text within `<script>` tags. The resulting *detagged content* is typically 2-3 times smaller than the original HTML. The algorithm operates henceforth on this representation.

The algorithm detects templates using a two-pass sliding-window controlled by four parameters: a window size W, a fragment frequency threshold F, a sampling density D, and a page sample size P. All are described below in more detail.

First pass: In the first pass, P pages are sampled uniformly at random from the crawled pages of the site[1] and

[1] Note that a uniform sample is critical here; if we were instead to crawl only the first few levels of a site, for example, significant biases could be introduced.

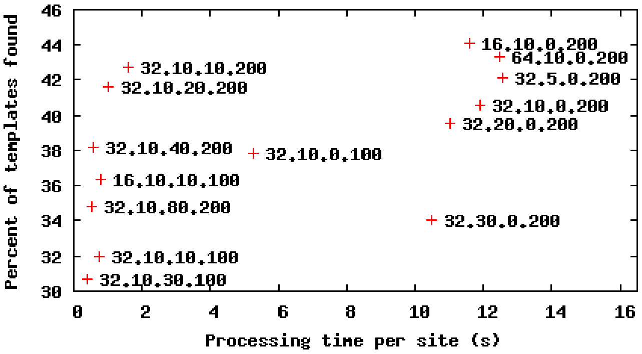

Figure 1: Running time and aggregate detection performance for a variety of parameters. Each point is labeled with the parameters W.F.D.P

a window of size W is slid over the text of those pages. At each offset, a counter is incremented for the fragment contained in the window. Those fragments which occur at least F times in the sample are passed to the second pass.

For efficiency, we introduce the sampling density parameter D in the first pass. A counter for a fragment is only kept if the hash of the fragment is zero modulo D. Thus, only 1 in every D fragments will be considered, but the downsampling is performed such that if a certain fragment is counted on one page, it will be counted on all pages. Other downsampling mechanisms, such as retaining every D^{th} fragment, do not have this essential property. We choose $D \approx W$ in order to increase the likelihood that after the filtering process concludes, consecutive fragments are contiguous. A coalescing process in the second pass ensures that the total volume of template text is counted correctly. A value of $D = 0$ in the experiments means all fragments are used.

Second pass: In the second pass, each page is scanned for these frequent fragments, and overlapping or contiguous fragments are coalesced into a single template.

At the end of the second pass, we have a set of template hashes which are either individual or coalesced fragments. These hashes are stored in a hash table, so that a new page can be broken into fragments and scanned quickly for templates.

Parameter settings:

Figure 1 shows the performance of this algorithm for various values of the parameters. These studies were performed on a 2.4GHz Pentium IV machine: the running time varies from 0.4 to 12.5 seconds per site, compared to 17.5 seconds per site for the DOM-based algorithm.

The `32.10.0.200` data point represents the algorithm with no downsampling of the number of available templates. Increasing or decreasing W results in a greater proportion found. However, D can now be set to achieve equivalent performance, with much improved running time. With $D = 40$ we achieve a similar proportion detected, with a running time of 0.59 seconds per site, or 3 ms per page, achieving a speedup of 20 times over the non-randomized approach.

Note that if P is set to a smaller value, the detection accuracy changes. As the number of pages sampled is decreased, F must decrease too, in order to detect the same fragments. With very small values of F, however, there is a risk of detecting greater numbers of spurious fragments.

In our experiments, we apply the algorithm with parameter settings 32.10.0.200.

4. TEMPLATES ON TODAY'S WEB

This section covers a family of experiments performed on a recent snapshot of the web.

4.1 Methodology

Our concern is to analyze the prevalance and nature of templates across the entire web without introducing unnecessary biases towards a particular subset. To begin, we make use of the IBM WebFountain data set, a large crawl containing over two billion pages and fifty million sites. From this set, we select uniformly at random a subset of two hundred sites, each containing at least two hundred pages.[2] The scale of the initial collection provides a broad underlying sample space from which to resample. We then manually classify these sites into categories and report results of template behavior for each category.

In addition to studying the amount of templated content on the web, we also study how templating behavior varies across seven site categories determined by inspection of the two hundred sites. These categories are intended to reflect various genres or modes of content that occur on the web, without regard to the nature of the content. Each has implications for the kinds of formatting and quantities of information that occur on each page. The categories are:

- Brochure. The online presence of a company or organization, typically containing events, reviews, press releases and diverse other information.
- Catalog. Listings of products, usually for sale.
- Community. Sites with content submitted and managed by a large number of individuals.
- Documents. Sites containing reference material. Many academic and government sites fall into this category.
- News. Sites which contain regular and editorially controlled updates on some range of topics. Most often this is local news or news devoted to specific topics.
- Personal. Homepage of a single individual, irregularly updated and containing a mix of content.
- Portal. Links to contents elsewhere. Often these are local portals, for a particular city or region.

A dating site, for example, falls most naturally into the "Catalog" category, even though the "products" are not really for sale. If the site also contained a chat forum, it would also fall into the "Community" category; thus, multiple assignments are allowed in our categorization.

Of the 200 sites, 109 were labeled, and the remainder were either pornographic (about 3%), no longer existent (about 15%), or not in a language understandable by the authors (about 30%); see below for a discussion of the biases introduced by this labelling. The number of sites in each category are shown in the following table.

[2] The requirement that each site contain at least two hundred pages introduces a bias; we discuss the nature of this and other biases below.

News	5
Personal	8
Community	14
Documents	14
Catalog	40
Brochure	42
Portal	16

Roughly 5% of sites are news sites, much fewer than the number of community and personal sites. The dominance of the commercial sector of the web is clear from the number of catalog and brochure sites.

Summary of Biases: The following biases exist in our sample. First, we consider only sites with at least two hundred pages in our crawl. Pages that lie on smaller sites represent approximately 20% of the overall crawl, and thus represent a non-negligible fraction; nonetheless, for technical reasons, we focus on the 80% of pages that belong to larger sites. Second, we consider non-pornographic sites only; we thus report results for the non-pornographic region of the web. Third, our classification results apply only to sites in English, but all other results apply to sites in all languages. This bias is difficult to overcome without enlisting the skills of many assistants. Finally, the crawling of sites is performed by a commercial crawler, which encodes many design decisions that may influence its behavior for or against a particular site. Overall, however, we believe the scope of the underlying dataset makes the results reasonably representative.

4.2 Results: proportions of template text

We ran both the DOM-based and the text-based algorithms over this sample set. The text-based algorithm reports the fraction of text content within templates on each page. The DOM-based algorithm reports the fraction of template versus non-template HTML content on the page, and then through post-processing of the resulting templates, also reports the fractions of links and text that appear within a template.

The two algorithms should report similar values for the fraction of text content that appears in templates. An examination of the results shows that the reported fractions of template content on average differ by only 7%, and show a similar level of agreement for each individual category. Given the extremely different approaches taken by these two appraoches, we find the measures of fraction of template content to be fairly stable across these approaches.

The results are shown in Figure 2. The figure shows a significant difference between the volume of templates across the different categories. Overall, the amount of template text on a page is around 50%, but this is significantly lower for News sites, and significantly higher for Personal sites. The types of text found in templates also varies across categories: for example, there are noticeably more links in templates in the Documents category.

4.3 Results: Counting and aggregating template bytes

Having considered the raw counts of template bytes across categories, we now turn to a more detailed view of the nature of these templated regions. We will refer to a contiguous sequence of bytes discovered by one of our algorithms as a "template hash." All analyses reported in this section are performed using the template hashes returned by the text-based algorithm, over the 109 sample sites. The algorithm found 64K occurrences of 3K distinct templates in this collection.

	Text- Based	DOM HTML	DOM Text	DOM Links
Brochure	56	59	53	55
Catalog	66	59	57	51
Community	64	51	50	53
Documents	35	57	26	58
News	12	15	8	12
Personal	67	68	77	52
Portal	44	48	39	43
OVERALL	53	53	46	49

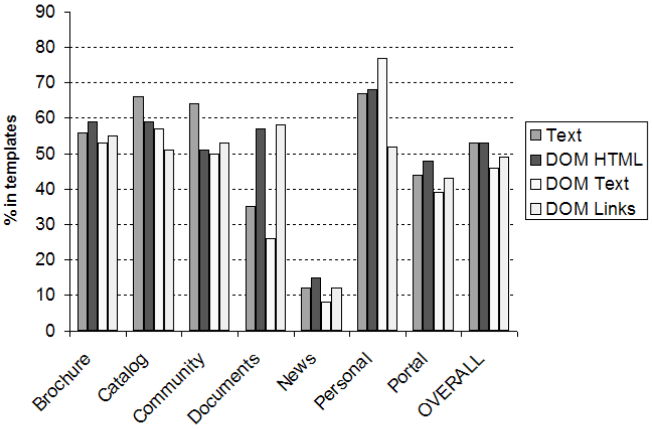

Figure 2: Proportions of templated content for all categories

First, we consider occurrences of template hashes across sites. For the set of sites we study, the amount of cross-site template duplication is extremely small. Over the entire set of three thousand distinct template hashes, exactly three distinct templates occur on more than one site. One instance is a message that a page has moved, occurring twenty times on one site and four on another; the second contains HTML header material that appears accidentally in the body of the page, occurring twenty times on one site and fifteen times on another; and the third contains part of an error message, occurring ninety-two times on one site and three times on another. Overall, same-site duplication is the dominant cause of the significant duplication shown in our results.

We now study the distribution of frequency with which each template byte occurs. Figure 3 shows a plot of rank of template hash (ordered by number of occurrences) versus number of occurrences, in log-log space. For comparison, the power law with exponent 0.8 is also plotted. The distribution is not a clean power law. Surprisingly, the majority of distinct template hashes in our data occur in the heavy region between $x = 200$ and $x = 2000$. The first two hundred hashes represent 40% of the number of template occurrences, and the dense region with $x \in [200, 2000]$ represents the next 55% of occurrences. Thus, a small template cache could reduce the amount of template traffic required by at least 50% (holding only three small template hashes per site on average), but the additional captured mass would grow more slowly until the cache reaches ten times that size.

Next, observe that our algorithms find contiguous regions of duplicated text (template hashes), but in fact these re-

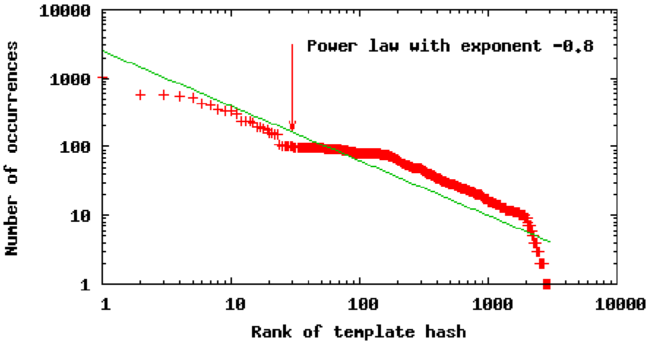

Figure 3: Re-use across templates.

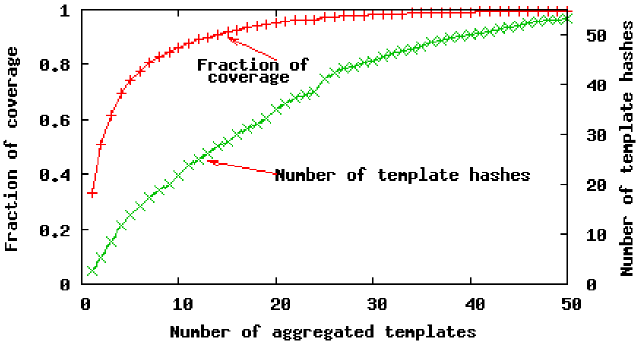

Figure 4: Re-use across templates.

gions may be coalesced into entire templates representing, for example, a header, a sidebar, and a footer. The three parts of such a template will not be contiguous on a page, and thus will represent multiple non-contiguous template hashes according to our algorithms. We now consider an algorithm to group these template hashes back together into full templates. Such an algorithm should, for example, identify that the sequence "1,2,3" of template hashes might be a template if a series of pages contain template hashes 1, 2 and 3 in that order, even if each page in the series contains other information of varying lengths in between these template hashes. Further, pages in the series might also contain other templates; for example, "1,2,3" might be a site-level template with a uniform header, sidebar and footer; but there might also be a distinct template for a particular part of the site, such as the "world news" section, which adds some headlines on the right side of the page. The algorithm should capture both of these occurrences. Finally, the algorithm should be extremely efficient, given the number of web sites across which it should run.

We propose the following simple greedy algorithm. During each phase, the algorithm finds the template hash which occurs most frequently in the entire set of pages, breaking ties by choosing the minimum average offset of the template in the page. Next, it advances a per-page counter to the location of the most-frequent template hash. It then finds the template hash which occurs most frequently to the right of the per-page counter, adds this hash to the current template, advances the per-page counters, and continues. At each stage, the template will have a certain length, and will occur on a certain number of pages. The final template output by this pass of the algorithm will be the one that maximizes the count times the length (representing the number of distinct template hash occurrences capture by the template). After finding this template, all occurrences of the template are greedily removed from all the pages in the collection, and the algorithm begins again.

Figure 4 shows the results of applying this algorithm to our collection; the numbers have been scaled to show per-site averages, so that we can meaningfully discuss a certain number of aggregated templates per site. The top curve in the figure shows for each number of aggregated templates the average over all sites of the fraction of template bytes on the site which are covered by that number of aggregated templates, based on the scale on the left axis. The lower curve shows for each number of aggregated templates the total number of template hashes needed to represent that many aggregated templates, based on the scale on the right axis.

The first one or two aggregated templates require about 3–5 template hashes per site, and capture around 40% of the total template bytes; over all sites, this corresponds to the early region of Figure 3 up to around two to three hundred templates across the entire collection of sites. Coverage grows to about 90% of total reuse based on around fifteen aggregated templates per site; this corresponds to a cache of approximately 50% of the total number of distinct template hashes. The slope of the lower curve shows that aggregated templates average about two template hashes, and hence around 30–60 bytes of actual site content. Of course, some templates are much longer.

5. VISUALIZING RICH TEMPLATE STRUCTURE

Templating behavior across a site is rich, complex, and hard to capture in a single numerical analysis. For example, a site may manifest a single uniform template, or entirely different templates for different regions of the site, and the templates may exhibit recursive structure, such as a common footer with many different headers. In this section, we introduce a pictorial representation of template behavior across a site to provide a complementary and more visceral view of the nature of templates on that site. In many cases, this view will provide insights into the more complex template structure of the site.

In order to display the pattern of template occurrences across a site in a compact but accessible way, we dispense with displaying text entirely and represent each template as a bar of color. To choose a color for each template hash, it is not feasible to assign each distinct template hash a unique color, since there may be very many templates on a site, and the colors would rapidly become difficult to distinguish. Rather, we assign a few broad classes of colors, based upon the frequency of occurrence of the template hash. Template hashes which occur fewer than 10 times over the site are colored red, those occurring 10-100 times are orange, and so on, through the rainbow scale of yellow, green, blue and finally purple indicating the most frequent templates. Since templates of similar kinds are likely to be used with similar frequency, this groups similar templates visually.

Each page, then, is represented as a single thin horizontal bar of gray, read from left to right, where the templates are placed along the bar according to their position on the page,

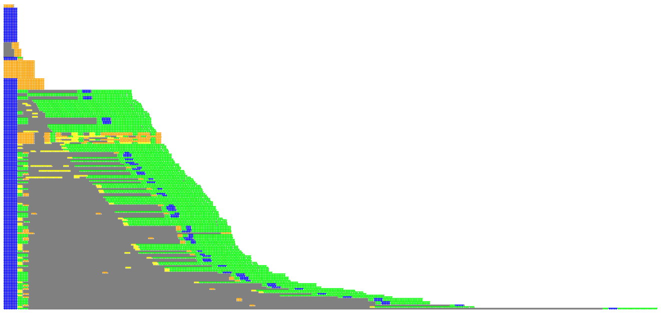

Figure 5: allsiouxlandhomes.com templates. This is a Catalog and Brochure site

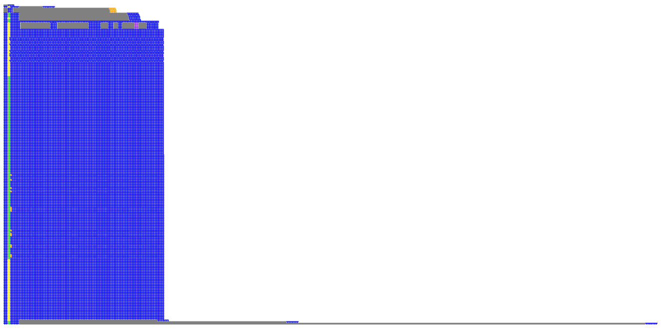

Figure 6: instrumentexchange.com templates. This is a Catalog site

and with length proportional to their size. Each horizontal bar is thus a miniaturized sketch of the page. The overall length of the bar is proportional to the size of the page. The plot is scaled so that the largest page fits into the maximum width allowed.

Pages must then be arranged in some order which is likely to group together pages with similar template structure. Our solution is simply to sort pages in order of increasing length. Intuitively, we expect that similar pages have similar lengths, and our plots indicate that this intuition usually provides a good ordering. Additionally, sorting by length makes the right-hand edge of the plot into a smooth curve, which gives a clear indication of the page size distribution on the site.

Figure 5 is a real estate site, which has been classified as a Catalog and a Brochure site. It is clear that there are a few categories of short pages, and two dominant categories of longer pages differing primarily by the length of the footer template.

5.1 Observations

From the examples presented in this paper, and across the range of sites in our sample, we can draw the following conclusions.

First, template structure is fairly simple. In the large majority of cases, pages contain only a header and a footer template. In the extremes of some News sites, this forms a negligible fraction of the page. In some cases, notably for some Catalog sites, the templates dominate the portion of the page which contains varying content. Figure 9 is presented as a rare counterexample, in which there is a large templated region in the middle of the text of each page.

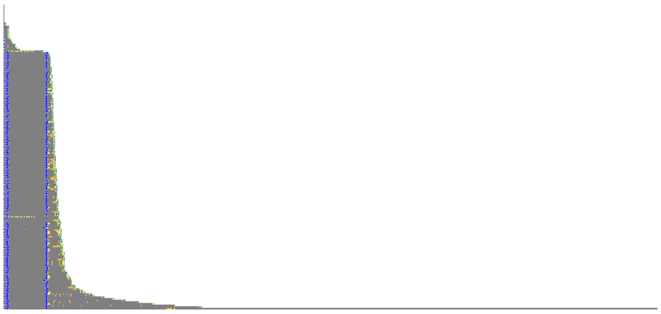

Figure 7: www.ade.az.gov (Arizona Department of Education) templates. This is a Documents and Brochure site

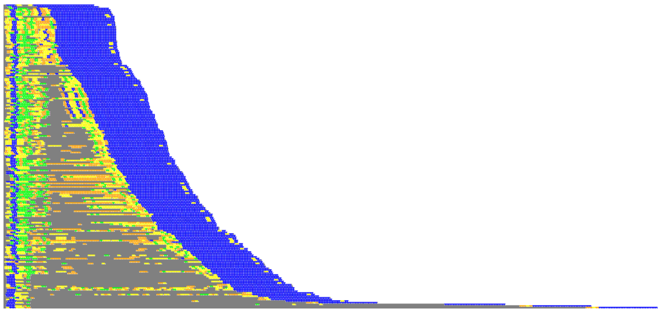

Figure 8: findbestcasinos.com templates. This is a Portal site

In the case of Portal sites, such as Figure 8, the header and footer pattern is clear, but there is also a lot of text which is repeated across the body of many portal pages, since portal sites typically lay out many small fragments of content onto a page. This results in a dust-like pattern of small templates in the plot.

We may consider the pattern of templates across a page as the “template schema” for the page. There are relatively few template schemata that are used by any particular site. The examples shown are representative in that most of the pages on a site will typically follow the same schema. The exceptions tend to be the shorter pages, which are often navigation pages, and redirect (302) and error (404) pages.

This relative paucity of distinct template schema types is largely because sites tend to have a single focus, such as selling products. In some cases we have found that when more than one form of template schema does exist, the schemata separate by age: older pages which have not been updated to a newer template schema are still available.

These visualizations suggest that an exploratory tool can be built, which can very easily present the pages on a site, grouped according to their template schema, and possibly perform different forms of text analytics based on the schema: indexing only content-rich pages and using link-rich templates to find key areas of the site, for example. Unusual pages, outliers in the clustering by schema, are also easy to present.

6. TEMPLATE EVOLUTION

In this section, we describe a set of experiments studying the evolution of templates from 1996 to 2004, based on a

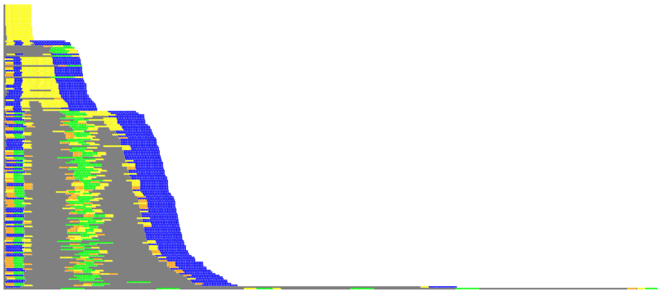

Figure 9: www.travellerpoint.com templates. This is a Community and Catalog site

	Unbiased	Popular
Non-empty Websites	78	105
Total pages	32K	42K
Avg snapshots/site	5	8

	Unbiased	Popular
Year of Snapshot	#Snapshots	#Snapshots
1996	2	19
1997	5	51
1998	10	46
1999	15	64
2000	50	162
2001	60	165
2002	98	178
2003	194	198
2004	24	39
Total	458	922

Table 1: Internet Archive data volumes for Unbiased and Popular collections of websites.

crawl of pages stored in the Internet Archive [5]. We study two sets of sites. The first set is the familiar collection of 109 unbiased sites introduced in Section 4. We will refer to this data set as the "unbiased" set. Of the 109 sites in the set, we found at least one snapshot for 78 of them.

Our second evolutionary dataset covers more popular web sites, and is better represented in the archive's historical database. Ntoulas et al. [9] used a set of 157 sites in order to study changes over time. While this set may be less representative of the web at large, it is perhaps more representative of the types of content that people typically browse, and it has been extensively studied by Ntoulas and his co-authors, allowing us to place our results in context. We found at least one snapshot for 105 of the 157 sites in the set.

We successfully crawled approximately 72K pages from the Internet Archive from these two datasets representing 1380 snapshots of a website at a particular time. Some details about these data sets are shown in Table 1.

For each snapshot, we identified templates using the DOM based detection method, and considered six regions on each page: links, text, and HTML within and outside templates.

6.1 Fraction of template content over time

Our first set of evolutionary results covers the fraction of content that appears inside versus outside templates as a function of time. The results and trends are similar for popular and unbiased sites, so we report only results for popular sites as the number of snapshots is larger. Figure 10 is a scatter plot in which each point represents a website from our popular dataset at a particular point in time (i.e., one of the snapshots of Table 1). The x axis represents the time of the snapshot. The y axis is the fraction of links on the page that occur inside a template. While coverage for sites in the 1990s is more sparse, it is clear that snapshots from 2002 and 2003 show a significantly larger proportion of sites with more links in templates. The best fit trend line shows a growth of 8% per year in the fraction of links that are inside a template.

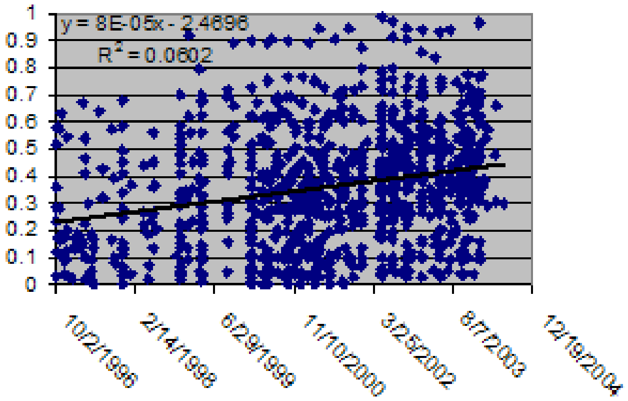

Figure 10: Fraction of links inside versus outside templates as a function of time. Collection: popular sites.

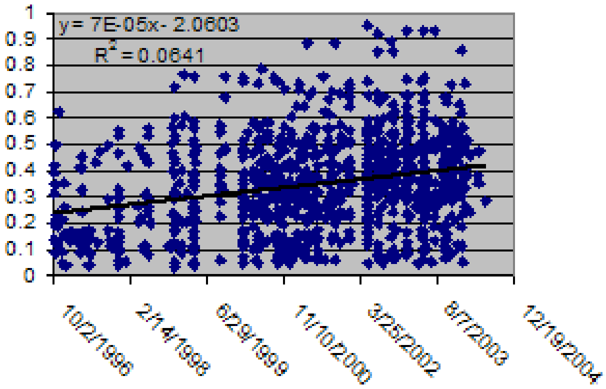

Figure 11: Fraction of HTML content inside versus outside templates as a function of time. Collection: popular sites.

Figures 11 and 12 show the same type of scatter plot for the fraction of the bytes of HTML, and the bytes of detagged content, that appear within templates. The best fit growth rates are about 7% and 6% respectively. Total bytes of HTML again shows a mass of more heavily-templated pages in more recent years. While many recent pages have more than 70% of their links in templates, this is not true for total HTML content, supporting the intuition that pages may contain menus, headers, footers, and sidebars with a large number of navigational links, but will still contain some reasonable amount of non-template content.

Table 2 shows summary information for these figures. The

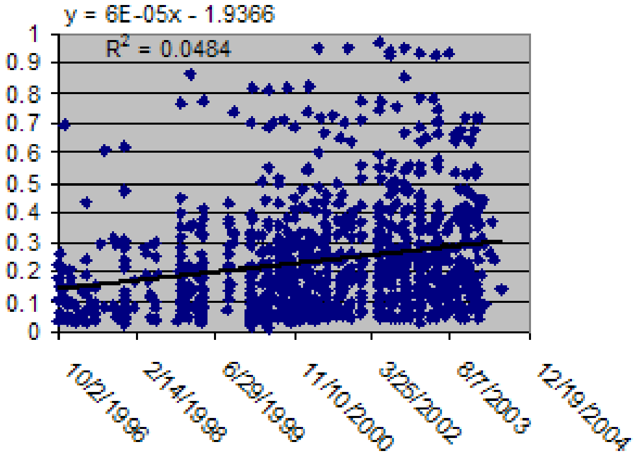

Figure 12: Fraction of text content inside versus outside templates as a function of time. Collection: popular sites.

	Unbiased Sites			Popular Sites		
Category	96–01	02–04	All	96–01	02–04	All
Links	44%	55%	52%	32%	42%	36%
HTML	39%	46%	44%	32%	40%	35%
Text	28%	38%	35%	21%	28%	24%

Table 2: Fraction of links, HTML, and text that appears in templates by data collection and date range.

popular sites show less overall template activity than the unbiased sites, though with similar trends. The unbiased sites from 2002 onwards show 38% of their text, 46% of their HTML, and fully 55% of their links in templates. Combining this aggregate information with the trend lines, we see that a large and rapidly-growing fraction of links appear in templates, suggesting that template-based navigation continues to increase in popularity. The aggregate results shown in this table are normalized for site size and number of internet archive crawls per site. Thus, the results should be taken as representative of the "average" page in the given collection.

6.2 Change rates

Ntoulas and his co-authors crawled each site of the popular set weekly, and performed experiments to capture the amount of change noted each week; this amount was found to be very small for most changes. We conducted a similar experiment to check whether the amount of change would be higher if we first removed templates from these pages. The Internet Archive crawls pages much less frequently than once per week, so the change on each visit will be much larger in our case. However, from our data we can estimate changes that occur less frequently than every hundred days.

We perform the following experiment. Consider a series of n snapshots of a web page, and let $x_1, \ldots, x_n$ be the value of the templated region of the page at each timestep. Let t_i be the time of the i^{th} snapshot. We will apply exactly the same approach to the *detemplated region* of the page; that is, all content on the page other than the templates. In this analysis, we consider the text content rather than the HTML or links. Consider the i^{th} snapshot, x_i. If $x_1 \neq x_i \neq x_n$ then we say that the value x_i is *bracketed*, meaning that we saw the page before this template appeared, and thus we have some estimate of the date when it appeared; and we saw the page after the template had disappeared, and thus we have an estimate of the date when it disappeared. For any bracketed value x_i, we define the *first* value $f(x_i)$ as the index i' at which $x_{i'} = x_i$, but $x_{i'-1} \neq x_i$. Likewise, the *last* value $\ell(x_i)$ is the index i' such that $x_{i'} = x_i$ but $x_{i'+1} \neq x_i$. The beginning $B(x_i)$, the time at which the template appeared, is then estimated to be $(t_{f(x_i)} - t_{f(x_i)-1})/2$. Likewise, the end $E(x_i)$ is estimated to be $(t_{\ell(x_i)+1} - t_{\ell(x_i)})/2$. Notice that these times must all exist if x_i is bracketed. Finally, the duration $D(x_i)$ is taken to be $E(x_i) - B(x_i)$.

For any time t, we say the active templates at t are all the templates such that $B(x_i) \leq t \leq E(x_i)$. Notice that the active templates at time t are all the templates that both exist on some page at time t and are bracketed (so that we can estimate their duration). The average duration at time t is then the average of the duration of all templates that are active at time t. Figure 13 shows the average duration as a function of time. The figure also shows a second curve in which the value of x_i is not the templated region of the page, but is the remainder of the page (that is, the detemplated region). In both cases, the average duration of a template can be seen to shrink dramatically over time, implying that the rate at which both content and templates are changing is shrinking. The average duration of templates is slightly larger than that of detemplated text, but the difference does not appear to be significant.

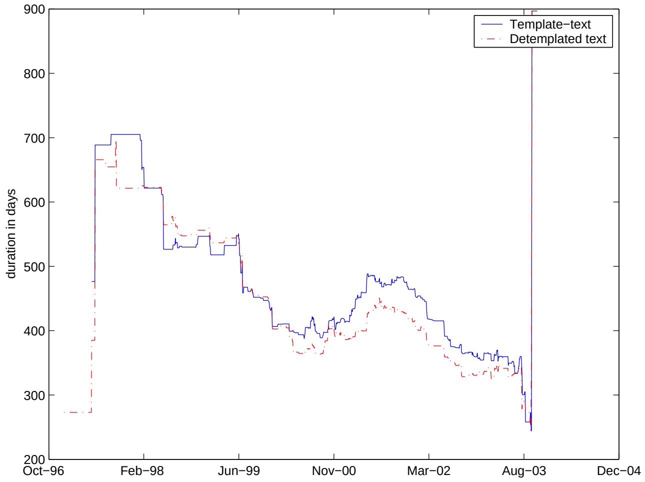

Figure 13: Average duration of all templates and detemplated pages existing at each point in time.

Figure 14 shows the histogram over the entire timeframe of the study of the average durations of templates. Due to the refresh rate of the Internet Archive, we do not have detailed information for content that changes more frequently than every hundred days. However, the figure demonstrates that both templates and detemplated content typically last for between fifty and three hundred days, with perhaps five percent remaining for two years or more.

6.3 Change rate versus change magnitude

Figure 15 shows an analysis of changes in the text content

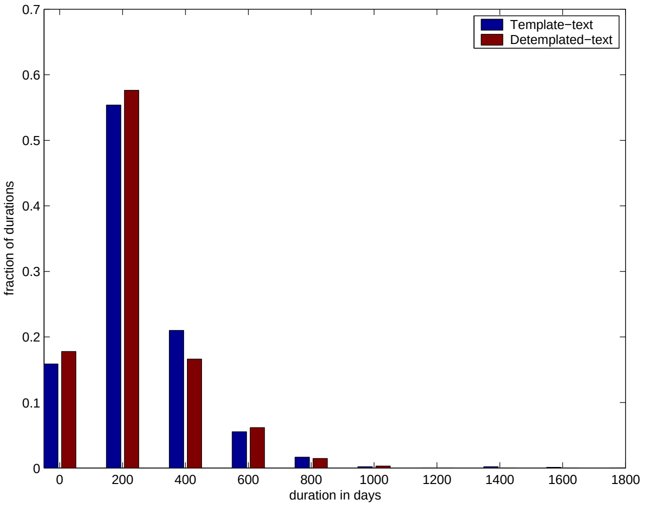

Figure 14: Histogram of durations of templates and detemplated content over all pages.

of pages from one version of a page to another. For two documents with word sets A and B, the magnitude of the change is taken to be: $1 - 2\frac{|A \cap B|}{|A|+|B|}$. The figure shows the distribution of the magnitude of change for the detemplated region of the page and for the entire page. Changes of magnitude 65% or larger are about twice as likely in the detemplated text, suggesting that results on large changes may be biased by the presence of a significant and unchanged template. Overall, however, the results in this figure are very similar to those of Ntoulas et al.

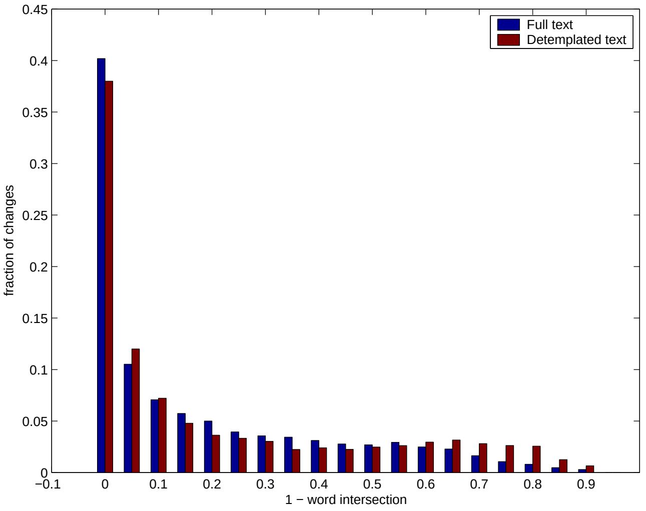

Figure 15: Distribution of magnitude of change in full text and detemplated content.

7. CONCLUSION

Templates represent 40–50% of the total bytes on the web, and this fraction continues to grow at a rate of approximately 6% per year. Similarly, the fraction of visible words, and the fraction of hyperlinks appearing in templates is extremely high. This finding implies that: (1) the graph structure of the web is increasingly dominated by boilerplate, suggesting that link analysis algorithms require understanding of templates; (2) with increased bandwidth, site creators are spending an increasing fraction of their resources on conveying information that has little raw content value, suggesting that improved caching and delivery mechanisms are needed.

8. REFERENCES

[1] Z. Bar-Yossef and S. Rajagopalan. Template detection via data mining and its applications. In *Proceedings of the Eleventh International Conference on World Wide Web*, pages 580–591. ACM Press, 2002.

[2] A. Z. Broder, S. Glassman, M. Manasse, and G. Zweig. Syntactic clustering of the web. *WWW6/Computer Networks*, 29(8-13):1157–1166, 1997.

[3] S. Chakrabarti. HyParSuite, `http://www.cse.iitb.ac.in/~soumen/download/`.

[4] B. Davison. *Recognizing nepotistic links on the Web*, pages 23–28. AAAI Press, 2000.

[5] B. Kahle. The internet archive, `http://www.archive.org`.

[6] H.-Y. Kao, M.-S. Chen, S.-H. Lin, and J.-M. Ho. Entropy-based link analysis for mining web informative structures. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, pages 574–581. ACM Press, 2002.

[7] N. Kushmerick. Learning to remove internet advertisement. In O. Etzioni, J. P. Müller, and J. M. Bradshaw, editors, *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*, pages 175–181, Seattle, WA, USA, 1999. ACM Press.

[8] V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1996.

[9] A. Ntoulas, J. Cho, and C. Olston. What's new on the web? the evolution of the web from a search engine perspective. In *Proceedings of the World-Wide Web Conference (WWW)*, 2004.

[10] R. Song, H. Liu, J.-R. Wen, and W.-Y. Ma. Learning block importance models for web pages. In *Proceedings of the 13th International Conference on World Wide Web*, pages 203–211. ACM Press, 2004.

[11] L. Yi and B. Liu. Web page cleaning for web mining through feature weighting. In *Proceedings of Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, 2003.

[12] L. Yi, B. Liu, and X. Li. Eliminating noisy information in web pages for data mining. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 296–305. ACM Press, 2003.

9. APPENDIX: SITES USED

Site	Categories	Size	Text-Based	DOM HTML	DOM Text	DOM Links
searchenginebase.com	Portal	200	42	45	28	36
www.sportsandmarineparafunalia.com	Cat	205	68	62	68	90
reven01.uboot.com	Pers	206	63	56	59	52
www.notliveyet.com	Cat	206	59	53	58	42
mirror2.www.umb.edu	Doc	207	28	35	21	29
www.naddak.com	Cat	207	42	59	46	63
www.worldmediafestival.de	Bro	207	56	34	23	11
www.reverbmag.com	Portal	208	32	60	49	76
www.asianyachting.com	Portal	210	4	11	2	15
www.bfranklin.edu	Bro	211	11	8	5	3
www.osemicro.com	Bro	211	68	46	21	71
midnight.moulen.org	Pers	216	80	38	56	57
bt.tradenow.ru	Cat	222	89	81	81	83
www.recticel.com	Bro	229	19	20	17	7
www.r28.org	Pers	232	25	12	7	12
www.asktooltalk.com	Doc	235	20	32	13	43
alsfree4u.freeblvd.com	Cat	239	87	35	73	27
www.hndlink.org	Bro	240	8	36	8	67
stonecipher.joeuser.com	Comm	242	100	100	100	100
qaborrower.lendingtree.com	Cat	245	42	63	25	74
www.gordie.ca	Pers	245	12	23	13	38
www.oldmonterey.com	Cat/Bro	245	41	22	31	9
www.mobig.co.kr	Cat	259	18	44	16	47
dreamhomeanloans.com	Cat	268	54	30	46	34
www.kuni-net.com	Pers/Comm	269	16	15	12	35
www.npa.uiuc.edu	Doc	275	9	38	12	68
alfie-uk.com	Pers/Bro	276	30	50	23	70
www.atltest.com.cn	Cat	284	11	22	12	43
allsiouxlandhomes.com	Cat/Bro	286	39	66	37	26
www.mv-o.de	Comm/Bro	286	8	19	23	10
findbestcasinos.com	Portal	292	60	52	51	76
www.auto-evasion.com	Cat	292	63	65	47	72
www.safetytechnology.com	Cat	298	17	17	9	16
www.rejis.org	Bro	303	54	22	18	23
www.ojc.cccoes.edu	Bro	310	18	9	14	0
www.aoc.ca	News/Bro	311	75	73	62	76
www.worldofppc.de	News/Comm	311	3	8	3	9
www.kinderklinik-bochum.de	Bro	313	76	52	54	67
www.loewe.dk	Cat	313	0	30	7	27
dev.ediware.net	Bro	318	8	15	10	6
homework-hotline.org	Doc	320	18	28	18	15
www.prowebdevelopers.com	Bro	324	51	73	69	70
www.stala.com	Cat	329	12	18	12	28
dlhp.berlios.de	Doc	336	2	2	1	0
www.allpme.com	Doc	337	97	92	92	96
www.lomejoraqui.com	Cat	351	62	71	64	85
www.austrian-mint.com	Doc	353	1	8	1	11
seniorstore.com	Cat	367	72	69	73	69
www.actualrh.com	Bro	368	43	25	22	23
www.art-tees.com	Cat	369	84	68	73	81
www.innisfileconomic.on.ca	Portal	376	40	93	88	51
www.umfa.utah.edu	Bro	377	17	57	14	60
www.oipartners.net	Bro	381	26	70	18	76
www.travellerpoint.com	Comm/Cat	386	52	67	56	64
www.motionvillage.com	Doc/Bro	395	13	38	8	56
www.sjc-den.org	Bro	409	19	43	20	68
www.clicksoft.co.kr	Bro	428	11	28	17	40
www.christianlove.net	Comm	433	65	43	51	37
www.covertford.com	Cat	461	81	54	52	58
software.qa.engineer.jobs.helpwantedsite.com	Cat	477	87	70	83	67
www.theedgerocks.com	Comm	485	72	64	59	83
www.aarons.com	Cat	512	86	67	60	91
www.partnergsm.co.il	Cat	513	23	35	29	6
www.rudystoys.com	Cat	536	78	55	75	75
energystar.gov	Doc/Cat	556	26	36	20	47
www.desmoinesdt.com	Portal	556	88	54	77	35
www.us.kernel.org	Doc	571	10	4	1	0
www.single.de	Comm/Cat	575	35	41	28	55
www.montagne.ch	Portal	576	22	47	19	54
www.eplus-online.de	Cat	655	62	34	53	0
www.ade.az.gov	Doc/Bro	656	71	77	49	81
www.dehs.umn.edu	Doc	660	10	22	6	38
www.diekuecheonline.de	Cat	693	82	93	79	97
www.nhldepot.net	News/Comm	696	51	22	34	5
www.enterprisewirelesstechnology.com	Bro	756	33	52	34	50
www.jscjewelry.com	Cat	775	45	30	36	35
www.dreamingheart.de	Comm	776	78	66	76	76
www.entelchile.net	Portal	803	7	33	8	35
www.vanloese.dk	Bro/Portal	816	3	12	7	7
www.corvetteimagegallery.com	Cat	828	95	64	71	53
www.berktek.com	Doc/Cat/Bro	844	30	62	21	35
www.gugapunk182.fotologs.com.br	Pers	868	75	66	70	52
caroig.com	Portal	890	36	45	29	47
www.assemblyplus.com	Cat	930	74	43	38	79
www.sh-online.de	Portal	999	83	76	70	71
www.brondby-bibliotekerne.dk	Bro	1009	33	40	31	38
www.thecoadletter.com	News	1054	12	20	7	22
cisab-do-stuff-matching-game-1.adult-friend-finder-online-dating-free-chat.com	Portal	1082	7	5	4	14
batsbaseball.com	Bro	1083	20	35	24	74
www.a-z.x0.com	Portal	1214	66	44	51	53
www.rental-living.com	Cat	1216	77	54	69	72
cpculoman.cpcusociety.org	Bro	1227	89	25	14	18
www.maggievalley.com	Portal	1328	25	15	8	6
www.buensoft.com	Cat/Bro	1458	51	41	48	43
www.buy-valtrex-online-a-i.biz	Portal	1601	10	9	8	15
instrumentexchange.com	Cat	1717	93	80	87	75
www.gemeinden-web.ch	Bro	1949	50	62	29	68
vancouverisle.film-can.com	Doc/Bro	2105	55	66	53	63
www.usubscribe.com	Cat	2196	49	54	46	30
www.dbndiscos.com	Cat	2645	37	51	51	56
www.sibiuonline.com	Comm	2779	76	78	69	83
uol.com.br	News	2905	5	6	3	0
www.schweinepriester.de	Comm	3324	51	28	28	14
www.rvrepos.com	Cat	3982	97	66	75	64
www.foxcreekleather.com	Cat	4593	52	23	33	6
www.lexicity.net	Comm	4625	87	38	40	41
www.snapoo.com	Cat/Portal	4639	77	64	69	50
www.jpmaroney.com	Pers/Bro	4944	82	75	87	52
kahncentral.net	Comm	5289	48	71	83	90