Changes in Webpage Structure over Time

Mira Dontcheva[1] Steven M. Drucker[3] David Salesin[1,2] Michael F. Cohen[4]
[1]Computer Science & Engineering
University of Washington
Seattle, WA 98105-4615
{mirad, salesin}@cs.washington.edu
[2]Adobe Systems
801 N. 34th Street
Seattle, WA 98103
salesin@adobe.com
[3]Microsoft LiveLabs, [4]Microsoft Research
One Microsoft Way
Redmond, WA 98052-6399
{sdrucker, mcohen}@microsoft.com

UW CSE Technical Report 2007-04-02

ABSTRACT

We present an analysis of the prevalence and nature of structural changes of websites. We study the evolution of some 12,000 webpages from 20 different websites over a period of five months. The websites cover a wide spectrum in both types of content and volume of traffic. We find that the structure of webpages from lower-volume sites changes very little, while webpages from high-volume sites change in mostly minor ways. Some of these sites go through drastic structural changes, but only on the order of once every couple of months. We discuss the implications of these observed changes for the design of structure-based extraction algorithms and how they can evolve over time. Our analysis leads us to the conclusion that structural extraction algorithms can play an important role in future applications for aggregating and summarizing Web content.

Categories and Subject Descriptors

H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia; H.3.m [**Information Storage and Retrieval**]: Miscellaneous

General Terms

Experimentation, Measurement

Keywords

webpage evolution, structural analysis, data extraction, wrapper induction, templates

1. INTRODUCTION

The Web changes daily. And, for the most part, its content is completely unstructured: anyone can create a website for any type of content with any structure or appearance. In addition, more and more frequently, we view the Web through websites that aggregate content from multiple sources, such as `kayak.com` or `farecast.com`. An enormous challenge for the people who maintain such websites is collecting the desired information in a reliable way from such a highly unstructured and constantly evolving data source as the Web. In this paper, we examine the prevalence and nature of these changes, through a study of the evolution of some 12,000 webpages from 20 different websites over a period of five months.

WWW2007, May 8–12, 2007, Banff, Canada.

Gibson et al. [3] find that 40-50% of Web content consists of webpage templates and that templatization is growing approximately 6-8% per year. Despite this growth in uniformity the task of reliably collecting structured content from the Web remains difficult. Websites that perform content aggregation today either forge business relationships with the companies providing the content or must constantly update their algorithms to reliably extract the desired information. RSS feeds provide a structured mechanism for delivering semantic content directly to users, but since RSS feeds have to be specified by the content providers they only include information specified by the hosts. Visions of the Semantic Web allow great benefits for end-users, but often require costly investments by companies with no immediate profit incentives.

There are numerous research efforts in the area of wrapper induction with the goal of automatically extracting and categorizing data from webpages (see Laender et al. [8] for a survey), including NLP algorithms, constructing models, and using ontologies. Although very encouraging, these approaches can be brittle and are typically specific to the type of content. Recently, a set of new approaches [1, 4, 5, 14] has emerged that use the structure of webpages to extract content. Structure-based approaches evaluate the structure of the document object model (DOM) behind a webpage to build a hierarchy of webpage elements. This hierarchy can then be used to find common patterns present in the structure, such as the items in a list or the cells in a table. The advantages of structure-based approaches are that they typically require very few examples and tend to be fast since the structure of a webpage is usually much smaller than the content of a webpage. One drawback is that structure-based approaches are, as the name implies, sensitive to any structural changes.

The goal of the work presented here is to begin to understand, categorize, and quantify the kinds of structural changes that take place within webpages. Our expectation is that this study will help guide the design and implementation of systems to help users gather and organize information. Towards this end, we have analyzed the structure of a set of webpages over a period of five months. To the best of our knowledge, our work is the first to study the structural evolution of webpages. We report on the type, amount, and rate of structural change of webpages from websites ranging in both size and type of content. We find that the structure

of most webpages tends to change regularly in small ways with occasional drastic changes every couple of months.

After we report on our findings, in Section 4 we discuss the implications for the design of structure-based algorithms and how they can evolve over time. We also propose that temporal data about the structure of a webpage can be used for automatic segmentation of the content in a webpage.

1.1 Related work

Previous analyses of changes on the Web were motivated by search, caching, and indexing applications and as such focus on changes of the content of webpages rather than their structure. Ntoulas et al. [12] collected weekly snapshots of 150 websites for one year and measured the evolution of content and link structure. Fetterly et al. [2] collected weekly snapshots of 150 million webpages for 11 weeks and measured degree of change across different classes of webpages. Ntoulas et al. found that most webpages change very little, and that those that do change do so frequently. Their results indicate that creation of new webpages is a much more significant source of change on the Web than the changes of existing pages. Fetterly et al. found that large pages tend to undergo many more changes than smaller ones and that past changes are a good predictor of future changes. In addition to degree of change, Lim et al. [10] analyze the clustering of changes in webpages to find that document changes are typically small and clustered.

Gibson et al. [3] studied the volume and evolution of webpage templates and found that 40–50% of content on the Web is template content and that the amount of template content grows on average by 6–8% per year. Gibson et al. used the structural content of the HTML tags to help identify template regions of a webpage, but they did not analyze how the structure of the templates changes over time. The growth in template content implies that structural approaches for extracting content may become more and more prevalent, and so we use our analysis of structural changes over time to discuss a set of implications for extraction algorithms and how they can adapt to changes in webpage structure.

The Internet Archive [7] has been collecting data for the last ten years, and when we first started this study we considered using this archive for our purposes. However, we found it did not include all of the types of webpages we wanted to analyze. The Internet Archive does not include websites that request not to be archived, such as `amazon.com` or `hotels.com`. Further, the Internet Archive does not always record pages daily and as a result does not include a fine granularity of data.

2. EXPERIMENTAL SETUP

To collect Web history data for our study, we selected a set of 100 webpages from 24 "popular" websites. We expected to see different types of changes in websites of different size and with different types of content. To categorize the types of changes that arise in websites, we chose to cover a wide variety of websites. We targeted aggregation applications and thus selected a number of catalog websites, which list products, usually for sale. We used `alexa.com` to select websites that varied in the amount of traffic volume. We selected some high-traffic-volume websites such as `amazon.com` and `citysearch.com` and some lower-traffic-volume, more specialized websites such as `giantrobot.com` and `borders.com`. We also selected websites with dynamic content, such as `nytimes.com` and `flickr.com`. Table 1 shows the list of websites we logged and their traffic rankings as of November 2006.

website	ranking	website	ranking
yahoo.com	1	epicurious.com	2,910
amazon.com	19	bbc.com	4,245
imdb.com	35	concierge.com	8,308
flickr.com	38	michaels.com	9,247
nytimes.com	95	tennis.com	27,072
weather.com	124	nyc.com	28,692
target.com	278	photoblogs.com	30,670
nih.gov	309	stevemadden.com	31,648
tripadvisor.com	459	giantrobot.com	143,718
travelocity.com	468	theparamount.com	317,540
citysearch.com	519	borders.com	358,950
hotels.com	1,011	buymusic.com	n/a

Table 1: We logged webpages from this set of websites for 5 months and collected over 15,000 webpages.

For our study, we chose one to five specific webpages from each website. The specific pages were chosen by hand to sample some breadth in the individual websites. We wanted to use more than one page to confirm the changing behavior for a website. However, for most blog sites such as `photoblogs.com` and certain other cases, there is only one dominant webpage, which we used. Since in this study we are interested in changes in the layout templates, or structure, of webpages, not in the number of layout templates employed by different websites, a few webpages from each website were sufficient for our analysis. Prior work on the evolution of the Web has used a much larger set of webpages. We selected a smaller set, but still one of considerable size, so that we could delve deeper with our analysis. This smaller data set allowed us to move beyond analyzing just magnitude and rate of change but also categorize the types of structural changes that occur on common websites. Prior analysis on the number of layout templates used within one website [12] points to the use of only a few templates; thus, we can expect similar types of changes in all the similar pages of a website — e.g. product webpages will change in similar ways to one another but not necessarily in a similar way to the front page.

We chose to log only English data websites, but we expect similar patterns of change to occur in foreign language sites as well. While language differences might not cause any structural differences, there might be cultural differences that could affect how often the overall structure of a site changes. That analysis is beyond the scope of this paper.

2.1 Collecting the data

We downloaded the 100 webpages every day for a period of five months from June 1st to November 15th, 2006. As is to be expected, all of the targeted webpages were not always available; however, the pages failed to download fairly infrequently. The only website that was unavailable more then 50% of the time was `imdb.com`, and so we omit those results from our analysis. For two weeks in August we were unable to collect data due to technical difficulties.

To collect the data we used the GNU Wget software package [11] and registered a process using the `cron` utility to

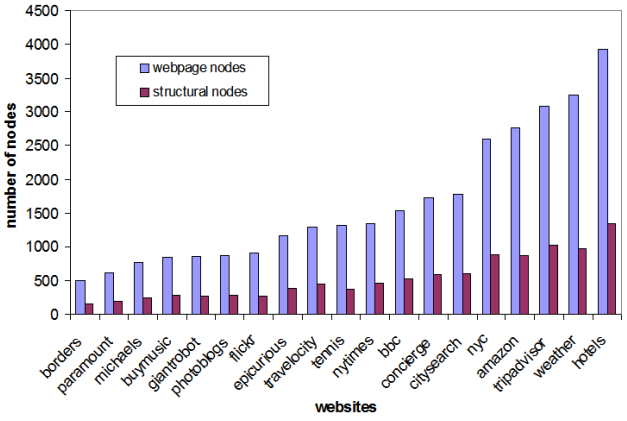


Figure 1: The webpages of the websites we logged vary greatly in size. The average number of nodes is 1600, and structural nodes make up approximately 33% of the total number of nodes.

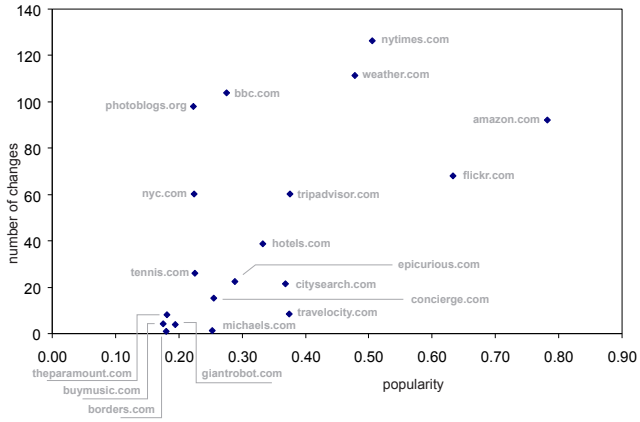


Figure 2: We consider the relationship between website popularity and average number of changes. To compute popularity we use traffic volume data from `alexa.com`. We find that generally more popular websites have a higher number of changes. Also, dynamic webpages, such as `photoblogs.org` and `bbc.com` also tend to exhibit more changes, which is in part due to the dynamic nature of the content.

collect data at the same time every day. In the end we collected over 15,000 webpages, which when compressed takes 1.5GB of space. Of those we were unable to use 3,000 of the downloads due to bad or missing data. The data set we analyze includes 12,000 webpages from 20 different websites.

2.2 Processing the data

We analyze the structural changes in webpages by examining the Document Object Model (DOM). The DOM is an interface for specifying the content, structure, and style of Web documents. With the DOM, every element in a webpage, represented with tags such as `<body>`, `<table>`, or `<p>`, becomes a node in the DOM hierarchy, with the `<html>` tag at the root of the hierarchy. We use Python for our analysis and employ the standard Python DOM implementation.

In order to create the DOM, a webpage must be transformed from HTML to XHTML. HTML is not always well formed because in its definition it does not require that all tags are closed. For example the `<p>` tag and `<img>` tag need not be closed in order to display an HTML document in a browser. To clean the HTML and convert it to well-formed XML, we use the TidyHTMLTreeBuilder Python library, which uses a library version of the HTML Tidy utility. The HTML Tidy utility first appeared in 1996 and has been improving ever since. It is available for many platforms and has wrappers for many languages.

To track changes in webpage structure over time, we examine the types of tags present in the HTML webpages. HTML includes over 50 types of tags, some of which are mainly concerned with style such as the size and look of text (`<b>` or `<h1>`), and others with structure, for example to specify divisions or tables (`<div>` or `<table>`). To analyze the structure of a webpage we could ignore all tags but a small set, but as there are a number of different ways to create webpages and form their layout, we instead chose to remove only elements that are definitely not structural and perform the analysis over the remaining hierarchy. We removed all script tags, which include `<script>` and `<noscript>`. We removed all text and image nodes, as we are not interested in how the webpage content is changing. And finally we removed style tags, because the style tags are closely related to the content and would affect our results for webpages that are highly dynamic and change content frequently. The HTML tags we considered stylistic are `<font>`, `<basefont>`, `<b>`, `<i>`,`<h1>`,...,`<h6>`, `<style>`,`<em>`, `<strong>`,`<u>`, and `<s>`.

Note that today many websites are turning to AJAX as an approach for building rich user experiences. As a result, there is a large amount of Javascript embedded in the webpages. Scripts can often add or remove content from the webpage to dynamically alter the structure. For this analysis we ignore this behavior as these type of effects tend to be local to a part of the page and do not drastically alter the whole structure. For the rest of this paper we will use the term "structural nodes" to refer to the remaining nodes after the removal of the script, style, and content nodes.

2.3 Measuring change

We measure change in the structure of the webpage using the number of structure nodes in the tree. We track if the number of nodes changes from day to day, and compare the trees from consecutive days to find changes in the tree structure. The comparison algorithm take as input two trees to be compared and recursively traverses the trees. At each node in the tree the algorithm compares the node's tag name and number of children. If the node name and number of children are the same, the algorithm continues; otherwise, we employ one of two strategies. Whenever there is a mismatch in the number of children, the *strict strategy* gives up and counts all the children and their subtrees as changed. The *flexible strategy* attempts to find a mapping between the different number of children and follows the subtrees as far as possible. We use a simplified version of tree alignment, which is employed by some structure-based extraction approaches [4, 14]. In our version, we only account for node insertions at the end of the list of children. Note that our comparison approach gives higher weight to structural changes closer to the root of the tree as these types of changes tend to be more significant.

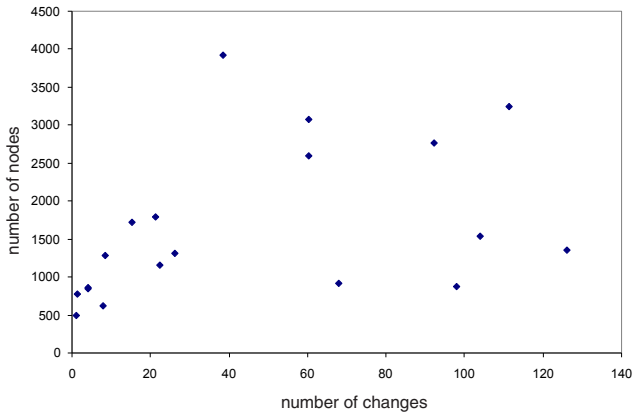


Figure 3: In this plot we consider the relationship between webpage tree size and number of changes. Previous research finds that large webpages go through more changes than small ones. We do not observe this correlation in our data set.

We track the location of the changes in the tree with two metrics: the depth of the change from the root of the tree, and the maximum distance of the change from the leaves of the tree. We also perform some qualitative analysis of the type of changes that are occurring, and discuss their impact on extraction algorithms in the next section.

3. RESULTS

First, we describe the general patterns of the data and how changes vary between low-volume and high-volume websites and between websites with relatively static and relatively dynamic content. We then describe some examples of the patterns in more detail and finally show some anecdotal evidence for the types of changes that are occurring.

Figures 1-3 show a high level summary of the data. Figure 1 shows the distribution of total number of nodes across the different websites. Structure nodes make up approximately 30% of the nodes for a particular webpage. Our findings on changes in the structure of webpages agree with Ntoulas et al.'s conclusions that many webpages do not change very much. Figure 2 shows the relationship between popularity and structural changes. We compute popularity using traffic information from `alexa.com`. A low traffic ranking means that the website is very popular. For the purposes of the figure we compute popularity as $1/log(\text{traffic ranking})$. The figure shows that popular websites tend to change more frequently than less popular websites. There are a few exceptions to this general trend. The `bbc.co.uk` and `photoblogs.com` experience a high number of changes as compared to other cites of similar popularity. This is due to the dynamic nature of their content. Although `nyc.com` has fairly static content for the hotel webpages we logged, during this period the website periodically updated the type of options available to the user, which accounts for a high number of changes. Also, note that traffic ranking is an absolute metric that ignores regional effects. The BBC website is more popular in some parts of the world than others. We can expect similar regional effects for `nyc.com`. Lim et al. studied the proximity of content changes in webpages and found that content changes tend to be clustered. Similarly, we find that most changes occur at the leaves of the DOM tree. And finally, Figure 3 shows how the amount of structural change varies with the number of nodes in a webpage. Unlike Fetterly et al., who found a strong correlation between the size of a webpage and the frequency and degree of change, our results show little correlation between the number of nodes and the amount of change.

As a general trend, the number of structural changes increases both with the amount of dynamic content and with the traffic volume of the website. We expect that this is the case because some of the changes are due to advertisements and dynamically added content, which is much more common on high-volume websites. Most changes to the structure happen very close to the leaves of the webpage, which implies either content changes or minor adjustments of the layout. Changes higher up in the tree are responsible for larger mismatches in the comparisons and are detrimental to automated extraction algorithms that rely on a consistent structure. These types of large-scale changes are not very frequent — on the order of every couple of months. The flexible comparison strategy matches a superset of webpages relative to the strict comparison strategy. On average, the flexible strategy matched 94% of the tree structure, while the strict comparison strategy matched 91% of the tree structure.

For a more thorough analysis of the changes, we discuss four examples.

3.1 A low-volume static website

An example of a low-volume static website is the Easy Street Records website, `buymusic.com`. Figure 4 shows four plots of the data for one product webpage. Plot (a) shows five curves for the number of nodes in four categories of nodes — structure, content, style, and script — as a function of time. It also shows the total number of nodes on the webpage. The script and style curves are close to zero because the layout for this website is fairly basic. This plot shows that the number of nodes for this particular webpage has not changed significantly over the last five months. Plot (b) shows the number of structure nodes and the number of nodes that matched using the flexible comparison strategy. Remember that we only compare adjacent samples. So, for samples from day 1, 2, 3, and 4 we compare day 1 and day 2, day 2 and day 3, and day 3 and day 4. The graph labeled "matching" shows how many of the structural nodes could be matched from day to day using the flexible comparison strategy. This plot shows that there are a very small number of changes, four in total over a period of five months. Of the four changes, two are small and two are significant, as shown by the sharp spikes for day 15 and 115. In this plot, we also show a "cumulative" comparison for the flexible strategy. So for samples from day 1, 2, 3, and 4, we compare how much of the tree remains the same between day 1 and day 2, day 1 and day 3, and day 1 and day 4. Plot (b) shows that the layout stays mostly static and that any changes to the structure are fairly permanent. Plot (c) and (d) show the location of the changes with respect to depth from the root and distance from the leaves. Two of the changes are close to the leaves, which is why they do not affect the total number of matched nodes shown in (b) significantly. The remaining two differences are caused by changes high in tree.

The structural change patterns at Easy Street Records are

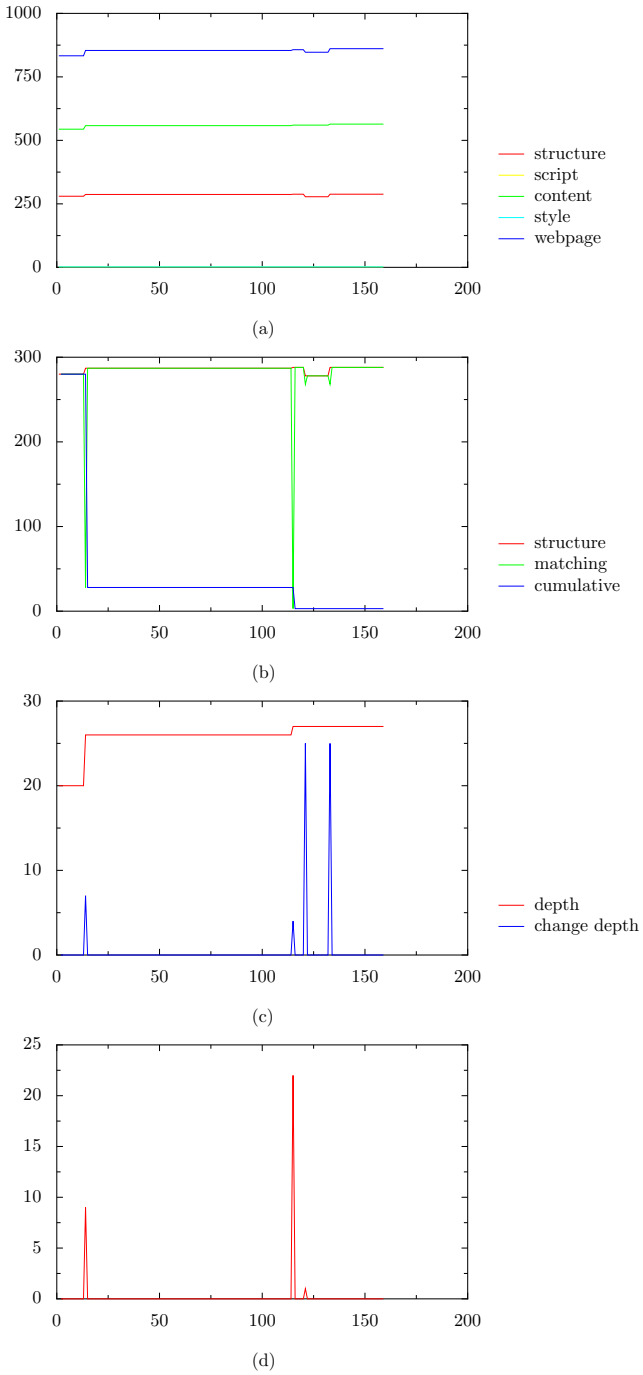


Figure 4: Changes in the structure of `buymusic.com`, a low-volume static website. Plot (a) plots the number of structure, style, script, content, and total webpage DOM nodes. Plot (b) graphs the number of structure nodes, and the results of the consecutive and cumulative comparisons. Plot (c) shows the average depth of the tree and the average change depth from the root of the tree. A change depth of zero implies that there aren't any changes. Plot (d) graphs the average maximum distance of the changes from a leaf in the tree. Note that all the plots are aligned to relate the location, magnitude, and type of change. Together these plots show that `buymusic.com` changes infrequently, but when it does the change can be significant.

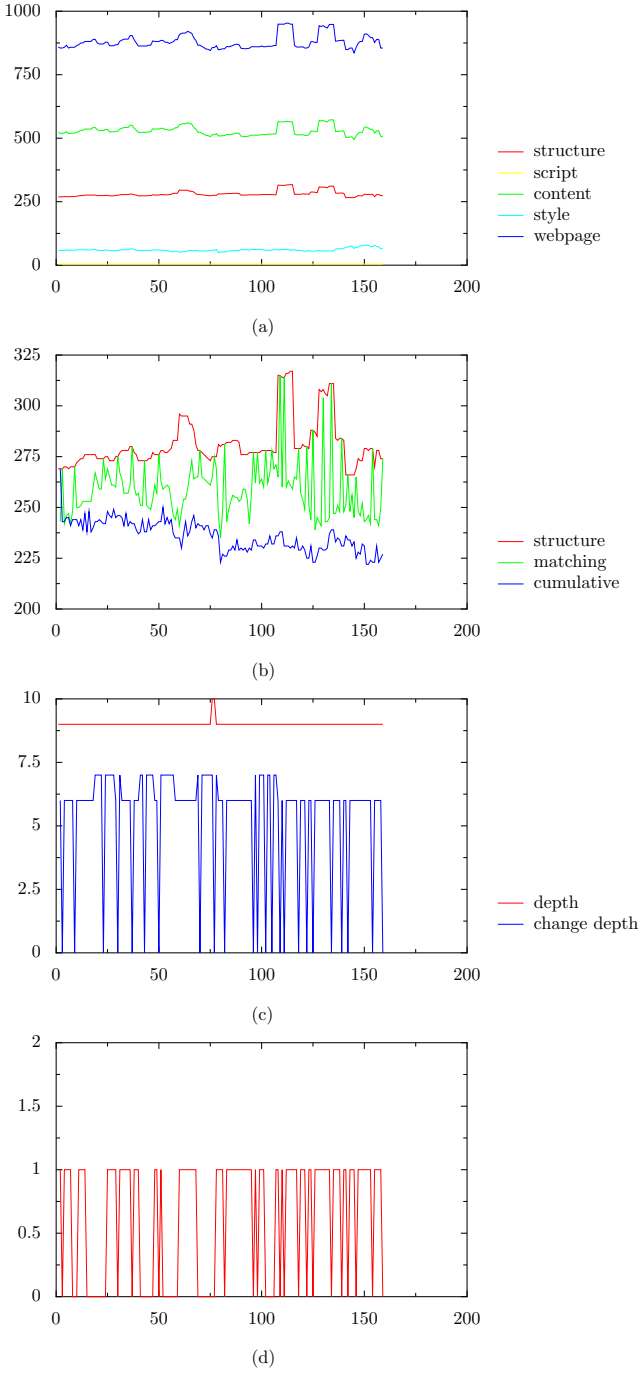


Figure 5: Changes in the structure of `photoblogs.com`, a low-volume dynamic website. Please see Figure 4 for a description of the characteristics of the four plots. Together these plots show that `photoblogs.com` changes frequently due to the dynamic nature of its content. Approximately 200 of the structure nodes remain static over the five month period showing no drastic structural changes. The structural changes due to content changes are at approximately the same depth of the tree and are always one link away from a leaf node.

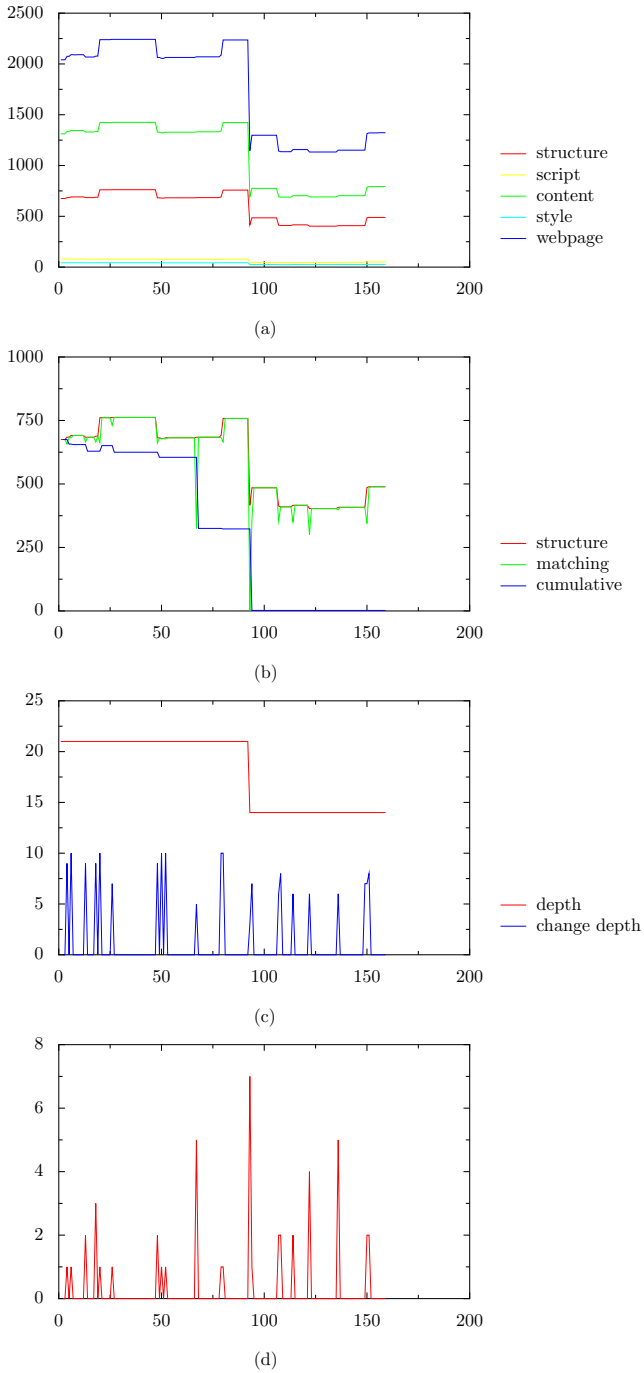


Figure 6: Changes in the structure of `citysearch.com`, a high-volume static website. Please see Figure 4 for a description of the characteristics of the four plots. Together these plots show that `citysearch.com` changes regularly, on the order of once every 15 days. Most of the changes are close to the leaves and affect the webpage structure in minor ways; however, there are two drastic structural changes at days 68 and 93. Structural extraction algorithms can accommodate the minor changes, but they may have difficulties with the drastic changes.

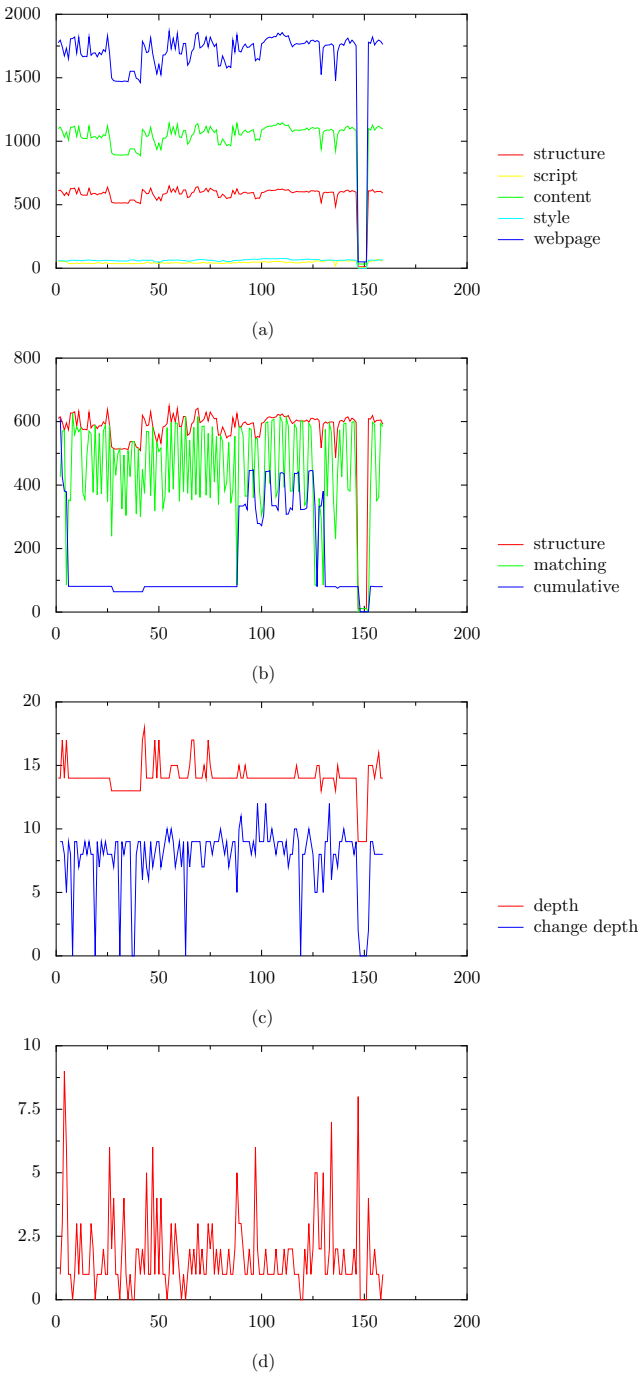


Figure 7: Changes in the structure of `nytimes.com`, a high-volume dynamic website. Please see Figure 4 for a description of the characteristics of the four plots. Together these plots show that `nytimes.com` changes on a daily basis due to its dynamic content. The magnitude of these changes is approximately 200 nodes and as these changes are primarily at the leaves, they are due to content changes. As Plot (d) shows, however, there are periodic drastic changes. We suspect that the New York Times website has a hierarchy of layout templates corresponding to different sections for different types of content.

very similar to the other low-volume static websites. These websites change the structure of their webpages infrequently, but when they do, they tend to change the structure significantly. Extraction algorithms can adapt to this without too much trouble because the content remains static. When there is a major change, the most commonly used tree matching algorithms may be successful; however, a more efficient approach might be to first re-find the content of interest by searching through the webpage and then rebuilding the structural extraction wrappers. Other websites that exhibit similar change patterns include `borders.com`, `theparamount.com`, `giantrobot.com`, and `michaels.com`.

3.2 A low-volume dynamic website

To see how low-volume dynamic webpages change structure over time, we examine `photoblogs.com`. Figure 5 shows how its structure changes. To illustrate the differences between the different categories we use the same plot characteristics as in the previous example. Plot (a) shows that the number of nodes in the whole website changes frequently as is to be expected from a blog with frequent new posts. There is a correlation between the content and structure changes, which implies that some structure is defined by the content; however, the magnitude of changes in the structure is much smaller. Plot (b) shows how the structure is changing and how well the flexible matching strategy performs. Since some of the structural changes are due to content change, fewer of the total structural nodes on the webpage remain static over time. The cumulative comparison graph reveals that up to 200 nodes remain static. Since the total number of structural nodes varies by up to 50 nodes, we can expect that while overall the structure is changing, it is only changing near the leaves. Plot (d) confirms this behavior as it shows all changes occurring one link away from a leaf. Plots (c) shows that most changes occur at depth 6.

The structural patterns at `photoblogs.com` are similar to other blog website we logged and the frequency of change at these low-volume dynamic websites is similar to high-volume dynamic websites. Extraction algorithms for these types of websites must be aware of and adapt to the dynamic nature of the content; however, they can rely on a fairly static webpage layout. The performance of these algorithms on such websites will largely depend on the type of data they are collecting. If, for example, the algorithm is extracting the most recent blog post or the main headline, even a simple extraction algorithm will be appropriate. If, however, the algorithm is extracting details within posts, a more complex algorithm that employs more than structure will be necessary. Other websites that exhibit similar change patterns include `flickr.com` and and `bbc.com`.

3.3 A high-volume static website

Figure 6 shows the evolution of the structure of one restaurant webpage from `citysearch.com`. Plot (a) shows how the webpage changes overall. This type of website changes more frequently than the low-volume static websites but much less frequently than the dynamic websites. Similar to the `photoblogs.com` plots, plot (a) shows a correlation between the overall change and structural change, which implies that new content is added and removed from the webpage. This type of content is likely new reviews, promotions, or advertisements. On day 93, the total number of nodes decreased dramatically from around 2300 nodes to about 1300 nodes. The total number of structural and content nodes decreased proportionally. This signals a redesign of the layout of the webpage. Plot (b) shows the effects of the changes. Most changes have small effects on the overall matching because they are close to the leaves of the tree. The webpage redesign on day 93 causes a complete mismatch. Plots (c) and (d) show the location of the changes in the depth of the tree. The changes at day 68, 93, 122, and 136 are closer to the root of the tree; however, only changes on days 68 and 93 cause significant mismatches.

Extraction patterns for these types of websites must adapt to periodic addition or removal of content. We expect that most of these modifications can be accommodated by a tree alignment algorithm that adapts to predictable changes in the content, such as the addition of reviews or promotions. Thresher [4] uses this type of algorithm to find data records in one webpage, but this algorithm could be extended for analyzing the same webpage over time. For significant changes, such as the re-factoring of the websites, tree matching algorithms will not be effective. To adjust to such changes algorithms could employ domain knowledge. Other websites that exhibit similar patterns of change include `nyc.com`, `hotels.com`, `tripadvisor.com`, and `tennis.com`.

3.4 A high-volume dynamic website

Figure 7 shows how the front page of the "Style" section of the New York Times website evolved over five months. Similar to the low-volume dynamic website, the webpage tree fluctuates in size, and the structural changes correlate with the content changes. Plot (b) shows daily changes in the structure. The effects of those changes remains fairly constant with the flexible matching algorithm matching approximately 60% of the structure nodes. The cumulative matching graph shows that unlike low-volume dynamic websites, high-volume dynamic websites experience structural changes in addition to those due to content changes. Initially the graph drops to 100 nodes, but for about 30 days it hovers around 400 nodes. This pattern is due to the fact that the New York Times website uses several levels of templating. The top level layout template is expressed with 100 nodes including the navigation bars. Each section uses separate layout templates. Plot (c) and (d) show that while most of the daily changes are fairly close to the leaves of the tree, corresponding to news stories, there are a significant number of changes, 9-10 in plot (d), closer to the root.

High-volume dynamic websites go through the highest amount of structural changes, and structural extraction algorithms may not be appropriate for extracting content. As with the low-volume dynamic websites, there is a website layout template, but for high-volume websites that template is limited to just the navigation bars. Layout templates for the rest of the webpage often change with respect to the content. Other websites that exhibit similar patterns include `weather.com` and `amazon.com`.

4. DISCUSSION

We also performed a qualitative analysis of the types of changes in webpages in order to get a better understanding for how extraction algorithms can adapt to those changes. There are two types of structural changes, minor changes and major changes. The minor changes are usually close to the leaves of the tree, while major changes usually occur

closer to the root and affect a large percentage of the tree.

4.1 Minor changes

Most of the changes we observed in all of the webpages are minor changes. We define *minor structural changes* as changes that are either close to the leaves of the tree or involve inserting or deleting a small number of nodes. In such situations, tree matching algorithms can usually determine a mapping between the existing nodes and the new ones. There are several flavors of tree-matching algorithms. The simplest one performs tree matching by using the Levenshtein string matching algorithm [9] to align the children of every two nodes that are compared. More complex ones compare the tree structures and compute mappings between them [13].

From our qualitative analysis, we hypothesize that the simple tree alignment algorithm will be effective for most minor changes. For example, `amazon.com` changes throughout the year to accommodate new events. Figure 8 shows the difference in layout for two consecutive days. The only change in the structure is the addition of the Father's Day promotion. A different type of minor change appears at `buymusic.com`. On day 115, the website changed its structure by adding another level in the hierarchy very high in the tree. Because our comparison algorithms does not account for this, we see a large mismatch in the comparison in Figure 4. Structural matching algorithms should be able to account for such changes, and in this situation a full tree matching algorithm will be necessary. Figure 9 shows the types of changes common in high-volume websites such as `tripadvisor`. New reviews are added but the overall structure of the webpage does not change.

4.2 Major changes

While most structural changes are minor, 74% of the websites we logged exhibited at least one major change. We define *major structural changes* as those that alter the structure of the webpage by modifications close to the root of the tree or removal or reorganization of sections of the webpage. In Figure 6 we saw a complete refactoring of the CitySearch website. This type of change is not frequent as it requires significant human effort. What is more frequent is the modification of a part of the page. For example, we observed `borders.com` move its navigation bar outside of the table that includes the main content. Although this modification was not visible to people visiting the site, the structural algorithm that was extracting content from website would have observed a different tree. Figure 10 demonstrates a similar pattern. Prior to June 29th, the advertisements on the Travelocity website were inside of the table that contains content. Following June 29th, they appear in a column adjacent to the table that contains the main content. Structurally there is little similarity between the Travelocity website on June 28th and on June 29th.

Structural algorithms have little hope of adapting to major changes using tree matching algorithms. In such situations, the algorithms can rely on domain knowledge to extract content through one of the wrapper induction algorithms or they can be assisted by a user. Prior research [4, 6, 1], has shown that users can easily assist algorithms in computing extraction wrappers. Users can share their expertise in a community, helping algorithms to adapt to major structural changes. As soon as one person goes to a website and points at the relevant content, all subsequent extractions will be successful.

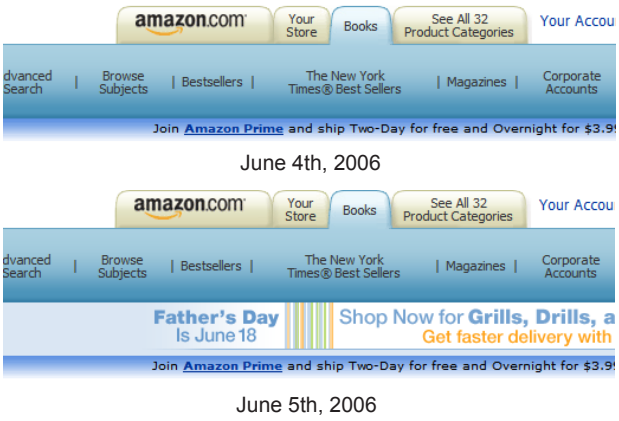


Figure 8: On June 5th the structure of the Amazon webpages we logged changed because of a promotion for the upcoming holiday. This type of change is fairly common and can be overcome by structural extraction algorithms because the change is localized and does not affect the rest of the webpage. Further, since it will eventually be removed, algorithms can learn to adapt to these types of modifications over time.

5. CONCLUSIONS AND FUTURE WORK

We have presented the first in-depth analysis of the structural evolution of webpages. Although the webpages in our data set do not represent all websites, we have sampled a specific space that is representative of the types of websites targeted by extraction algorithms. Our analysis leads us to the firm conclusion that structural extraction algorithms have a role in future applications for aggregating and summarizing web content. While they are not appropriate for all types of changes, they can be effective for most.

This analysis has raised a number of interesting questions. First, the growing amount of AJAX and Flash applications will change the way we study the evolution of the Web. Today, webpages are often customized to help the user with context, and thus the same webpage may look very different depending on the previous webpage viewed by the user. Further, websites such as `gap.com` completely break the familiar hyperlink/webpage paradigm and are impossible to log because the content is only presented when the user interacts with the webpage.

Our analysis of webpage changes leads us to question whether we can automatically segment a page into foreground and background pieces. A foreground piece may include all of the "relevant" information on a webpage, such as the name, price, and reviews of a book, and the "background" may include the navigation bars and all the advertisements and promotions. Gibson et al. [3] show a visualization of the amount of layout templates in webpages and report that every website has only a few layout templates. Perhaps we can do a space and time analysis of a website and automatically extract the most static and most dynamic pieces of relevant content.

Further, we believe the user has a role to play in spec-

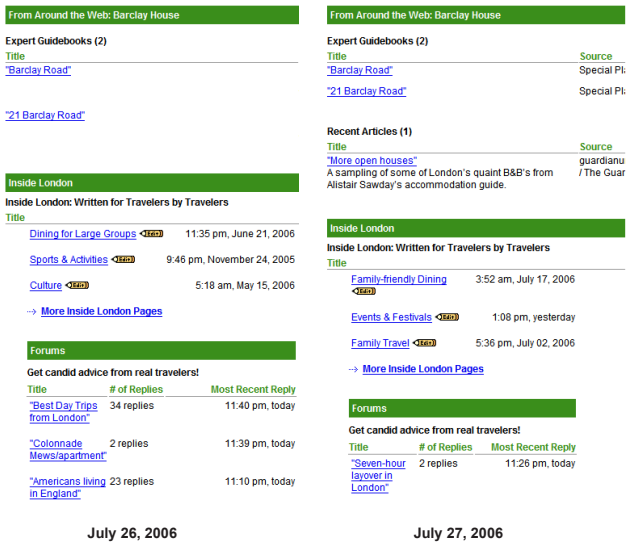


Figure 9: A common type of change for all websites is the addition of new reviews. On July 27th, TripAdvisor updated the reviews and advice they display for the Barclay House Hotel. The addition or removal of content does not significantly affect the structure of the webpage, as the changes happen at the leaves of the tree.

ifying the type of content that should be aggregated and summarized by applications. In conclusion, we hope that this analysis of the changes in webpage structure helps further development of content extraction algorithms as they will be a critical part of future applications for aggregating and summarizing Web content.

6. REFERENCES

[1] M. Dontcheva, S. M. Drucker, G. Wade, D. Salesin, and M. F. Cohen. Summarizing personal web browsing sessions. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 115–124, New York, NY, USA, 2006. ACM Press.

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

[3] D. Gibson, K. Punera, and A. Tomkins. The volume and evolution of web page templates. In *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 830–839, New York, NY, USA, 2005. ACM Press.

[4] A. Hogue and D. Karger. Thresher: automating the unwrapping of semantic content from the world wide web. In *WWW '05: Proc. of the 14th international conference on World Wide Web*, pages 86–95, New York, NY, USA, 2005. ACM Press.

[5] D. F. Huynh, R. C. Miller, and D. R. Karger. Enabling web browsers to augment web sites' filtering and sorting functionalities. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 125–134, New York, NY, USA, 2006. ACM Press.

[6] U. Irmak and T. Suel. Interactive wrapper generation with minimal user effort. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 553–563, New York, NY, USA, 2006. ACM Press.

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

[8] A. Laender, B. Ribeiro-Neto, A. Silva, and J. Teixeira. A brief survey of web data extraction tools. *SIGMOD Record*, 31(2), June 2002.

[9] V. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics-Doklady*, 10:707–710, 1966.

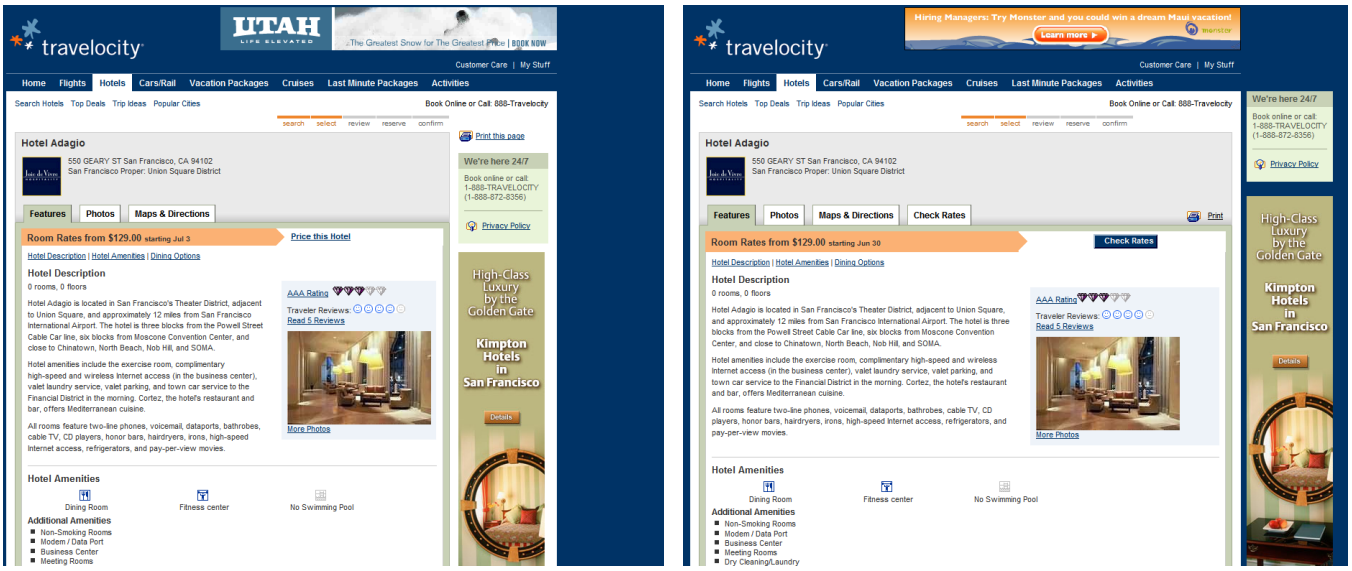
[10] L. Lim, M. Wang, S. Padmanabhan, J. S. Vitter, and R. C. Agarwal. Characterizing web document change. In *WAIM '01: Proceedings of the Second International Conference on Advances in Web-Age Information Management*, pages 133–144, London, UK, 2001. Springer-Verlag.

[11] H. Nikšić. Gnu wget. `http://www.gnu.org/software/wget/`.

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

[13] K.-C. Tai. The tree-to-tree correction problem. *J. ACM*, 26(3):422–433, 1979.

[14] Y. Zhai and B. Liu. Web data extraction based on partial tree alignment. In *WWW '05: Proc. of the 14th international conference on World Wide Web*, pages 76–85, New York, NY, USA, 2005. ACM Press.



June 28th, 2006 **June 29th, 2006**

Figure 10: Periodically, although not very often, websites will redesign their layout templates and drastically affect the structure of the webpages. Travelocity changed the structure of their webpages drastically on June 29th. They moved advertisements outside of the main region of content, added new functionality for printing, and created a new tab for checking rates. Structure-based extraction algorithms are sensitive to these types of drastic structural changes.