Changing How People View Changes on the Web

Jaime Teevan, Susan T. Dumais, Daniel J. Liebling, and Richard L. Hughes
Microsoft Research
Redmond, WA 98052 USA
{teevan, sdumais, danl, richardh}@microsoft.com

ABSTRACT
The Web is a dynamic information environment. Web content changes regularly and people revisit Web pages frequently. But the tools used to access the Web, including browsers and search engines, do little to explicitly support these dynamics. In this paper we present *DiffIE*, a browser plug-in that makes content change explicit in a simple and lightweight manner. DiffIE caches the pages a person visits and highlights how those pages have changed when the person returns to them. We describe how we built a stable, reliable, and usable system, including how we created compact, privacy-preserving page representations to support fast difference detection. Via a longitudinal user study, we explore how DiffIE changed the way people dealt with changing content. We find that much of its benefit came not from exposing expected change, but rather from drawing attention to unexpected change and helping people build a richer understanding of the Web content they frequent.

ACM Classification: H5.2: Information interfaces and presentation: User Interfaces – Graphical user interfaces.

General Terms: Design, Experimentation, Human Factors

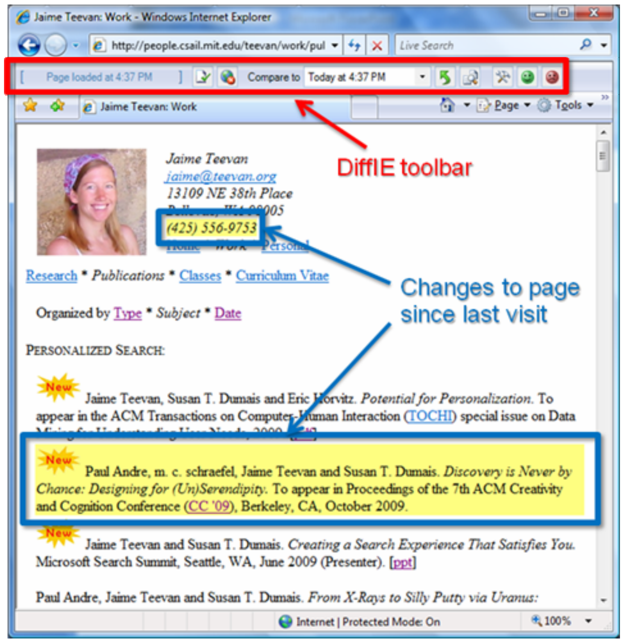
Author Keywords: Web dynamics, change, revisitation, Web browsing, re-finding, dynamic information

INTRODUCTION

When you visit a colleague's Web page, do the new papers she has posted jump out at you? When you visit a conference Web page, is it obvious that the conference schedule has changed? In this paper we describe *DiffIE*, a system that facilitates interactions like these by supporting awareness of Web content change. DiffIE is a Web browser plug-in that caches the Web pages a person visits and highlights any changes to a page when they return to it.

Previous research suggests that people return to content on the Web regularly [2, 22, 23]. Web content also changes regularly [10], and the content people revisit is particularly likely to change [2]. Although changes affect [2], drive [15], and interfere [21, 23] with people's revisits, Web browsers do not support a historical view of Web content. DiffIE has been designed specifically to support awareness of how a revisited page has changed without interfering with the existing Web browsing experience. An example of DiffIE in use can be seen in Figure 1. Through DiffIE the changes to a researcher's publication page since the user's last visit are highlighted, drawing attention to the fact that the researcher has added a new paper. In this example, the author of the Web page also draws attention to new papers using a "new" icon. Although this can be helpful, such annotations are author-centric, not user-centric, and, as in Figure 1, do not always reflect what is new to the user. Additionally, once the change has been seen there is no need to emphasize it again, and different sites expose change differently, if at all. Regardless of whether and how page authors choose to identify new content, DiffIE provides a consistent lens through which to view changes.

In this paper, following an overview of related work, we discuss how DiffIE is implemented and highlight how we addressed the interesting technical issues encountered via prototyping and a large-scale 1-on-1 demonstration. We then discuss how we deployed the plug-in within Microsoft and studied its use during daily Web browsing. We present the primary ways users took advantage of having Web content change exposed, including several unexpected uses that emphasize how DiffIE can help people understand


Jaime Teevan: Work - Windows Internet Explorer
http://people.csail.mit.edu/teevan/work/pul
Live Search
Page loaded at 4:37 PM
Compare to
Today at 4:37 PM
Jaime Teevan: Work
Page
Tools
DiffIE toolbar
Changes to page since last visit
Internet | Protected Mode: On
100%
New
New
New

Jaime Teevan
jaime@teevan.org
13109 NE 38th Place
(425) 556-9753

Research • *Publications* • Classes • Curriculum Vitae

Organized by Type • *Subject* • Date

PERSONALIZED SEARCH:

Jaime Teevan, Susan T. Dumais and Eric Horvitz. *Potential for Personalization.* To appear in the ACM Transactions on Computer-Human Interaction (TOCHI) special issue on Data

Paul Andre, m. c. schraefel, Jaime Teevan and Susan T. Dumais. *Discovery is Never by Chance: Designing for (Un)Serendipity.* To appear in Proceedings of the 7th ACM Creativity and Cognition Conference (CC '09), Berkeley, CA, October 2009.

Jaime Teevan and Susan T. Dumais. *Creating a Search Experience That Satisfies You.* Microsoft Search Summit, Seattle, WA, June 2009 (Presenter). [ppt]

Paul Andre, Jaime Teevan and Susan T. Dumais. *From X-Rays to Silly Putty via Uranus:*

Figure 1. DiffIE in action. Changes to a publication page since the user's previous visit are highlighted.

more about the Web pages they visit than they currently do. We conclude with a discussion of improvements that could be made to the system based on what we have learned.

RELATED WORK

Even though the Web is constantly changing, most Web tools deal only with a single time slice of content. Browsers show only the current version of a page, and search engines use the crawled version as the source for their indices.

Several groups have studied how frequently Web pages change and by how much [2, 7, 10, 20], and found that there are significant amounts of change. Previous research has also revealed that people often revisit Web pages; as many as 50% to 80% of page visits are repeat visits [2, 6, 8, 21, 22]. Adar et al. [2] found that 66% of the pages people revisit change during a 5 week period, and on average 20% of the content changes. The motivation for revisitation is often related to the content change [2]. For example, people sometimes revisit to monitor for new content [15].

Systems like the Internet Archive (archive.org) provide access to historical versions of Web pages. Web search engines also provide a cached version of the pages they index. In each case, a person must explicitly request to see previous versions to understand the page dynamics, and the relationship between the previous versions and the current Web page is not easy to identify. Furthermore, the cached versions are agnostic to users' revisitation patterns.

Kellar et al. [15] describe Web browser enhancements to support monitoring and notification of Web page change. Applications like WebSite-Watcher (aignes.com), Change-Detect (changedetect.com) and the Firefox Update Scanner (updatescanner.mozdev.org) allow people to subscribe to Web pages as they would an RSS feed to be notified of changes. Explicit subscription and notification works well when a user expects change within a fixed set of Web pages, but does not work to identify unanticipated changes. DiffIE focuses more broadly on the identification and visualization of change during normal browsing behavior.

Jatowt et al. [13] discuss several ways historical information about Web pages could be used to enrich Web browsing. They suggest example applications such as providing access to previous snapshots of the page, identifying salient terms in previous versions, and mapping a representation of change onto the current page (e.g., by showing content creation dates or highlighting links to indicate that the user has previously revisited the link). The latter is similar to the approach we take in DiffIE, where we seek to enhance Web browsing by providing *in situ* and lightweight annotations of change that are applicable to all pages.

Change notification has been studied by Borodin et al. [3] to support Web accessibility. The Dynamo system identifies dynamic Web content and makes it accessible to visually impaired users. Using VoiceXML dialogs, users can choose to access only changed content or to jump directly to the changed content. DiffIE visually (v. aurally) displays change on every visit (v. only on refreshes), and thus deals with different UI issues and additional storage and privacy concerns. We also provide a longitudinal study of its use.

The AT&T Difference Engine (AIDE) [9] archives Web pages and supports both notification of page changes and visual display of differences between versions. Display techniques include a side-by-side presentation, a view that shows only differences, and an integrated view that summarizes the common, new, and old material similar to what is done in Microsoft Word's change tracking. This makes changes explicit, but also changes the user experience from browsing to explicitly comparing differences in detail. Liu et al.'s WebCQ system [17, 18] uses similar methods for presenting two different versions of Web pages, but the focus of their work is on developing scalable methods for identifying different types of changes.

Systems have also been developed to support the historical exploration of regions of a Web page. Nadamoto and Tanaka [19] describe a Comparative Web Browser (CWB) which presents two pages concurrently in a way that allows their content to be synchronized. A user selects a region of interest on the current page and pages that have similar content are automatically retrieved. Adar's et al.'s Zeotrope system [1] allows people to view historical data for a Web page gathered from a large scale Web crawl by rewinding pages. They discuss several techniques for specifying regions of interest in a current page (e.g., visual regions, textual matches) and visualizations for presenting the relevant historical information. These sophisticated display techniques could provide interesting extensions to DiffIE to be triggered following the awareness of change.

Site owners have also taken interest in aiding users in identifying changes. Since the early days of the Web, site authors have flagged new content (e.g., with a "new" icon or timestamp). The company 37signals (37signals.com) created a technique by which authors can briefly highlight new content and then have the highlighting fade to temporarily draw attention to novel blocks of content. However, author-flagged content is not necessarily new for the reader. Some sites track changes to content of other sites; for example, one tracks edits to FOX News headlines (thequickbrown.com), and many sites allow subscribing to their changes through email or RSS feeds.

The work reported in this paper expands on previous work in several ways. First, we take a user-centric approach to emphasizing change, rather than content- or author-centric. Whereas prior work focused on when the content changes relative to the page itself (using polling or notifications to retrieve new content), DiffIE shows change from the user's perspective (using user revisitations to guide the storage and analysis of Web pages). Second, the goal of DiffIE is to augment Web browsing rather than to replace it with a different information exploration activity. Thus we show page changes *in situ* as a user is browsing the page using simple highlighting techniques. Finally, we deployed the system, studied how it is used, and iteratively designed features to support both anticipated and unanticipated uses.

OVERVIEW OF DIFFIE

DiffIE is an Internet Explorer browser plug-in that caches the pages users visit and, when a user revisits a page, highlights elements that have changed since the user's previous visit. The goal of DiffIE is to support people in understanding change on the Web in a way that augments their current Web browsing without interfering. We use highlighting to annotate the changes that have occurred since the user last visited the page. For some pages, like news pages or blogs, people may expect change and, in fact, visit the page to see the changes. DiffIE highlighting allows them to quickly see what is new since their last visit. For other pages, such as conference Web pages or publication listings (e.g., Figure 1), people typically revisit to re-find old information rather than to explicitly look for change. In these cases DiffIE can draw people's attention to unanticipated but interesting or important changes.

As can be seen in Figure 2, DiffIE contains three major components: a *cache* where representations of previously visited pages are stored, a *comparison component* that listens and responds to browser events and performs the comparisons of the current page with previous instances, and a *toolbar component* that contains the user interface. All three components are discussed in greater detail below.

The DiffIE system described is the result of an iterative design process. The initial prototype system was used by the authors for their Web browsing for several months. An improved version was demonstrated individually over the course of several days to 300 Microsoft employees, with their reactions, problems encountered, and usage scenarios folded back in to the design. The resulting system was then deployed to eleven people for regular use, with feedback gathered during use and via structured interviews at the end of a two week period. The design decisions made as a result of insights gained during this iterative development process are highlighted in the description of the current system below. In the subsequent section, we summarize the ways we observed people used DiffIE in practice.

DiffIE Cache

In order to highlight how a page has changed since it was last visited, DiffIE needs access to previous versions of the page. For this reason, a major component of DiffIE is a Web page cache. In this section we discuss how pages are represented in the cache, and explain key design decisions.

Web Page Representation

DiffIE identifies changes to text-based Web content at the Document Object Model (DOM) level, including hidden elements such as drop down menus that may be exposed on click or mouse events. Pages are represented internally as a tree of hash values to support this DOM-level comparison of text across pages. The text nodes of a Web page are typically the leaves of the DOM tree. The content of these nodes are hashed using the MD5 algorithm, and the information is propagated up the tree with each parent node assigned a *subtree hash value* corresponding to the MD5 hash of all the hashes of its children, in order. This enables fast comparison across complicated DOM structures.

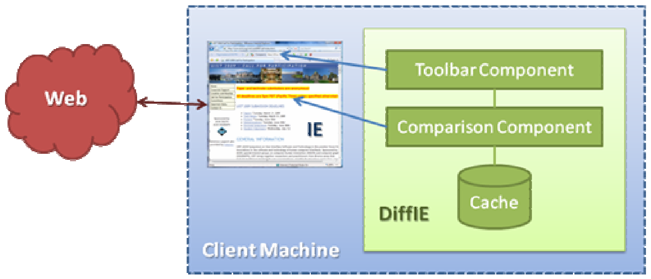


Figure 2. The DiffIE architecture. DiffIE is a Web browser plug-in that resides on the client machine. It consists of three parts: a cache, a toolbar component, a comparison component.

Hashing the text means that if a single word changes, it is possible to identify the DOM node where the change happened, but not which word has changed. Additional information could be stored to highlight word-level changes, but this would require a larger storage footprint and potentially compromise the user's privacy. Since Web page designers tend to chunk related content, DOM-level analysis works well in practice to group related words for highlighting, even when only one of the words in a node has changed. Changes to images or content other than text are not currently identified; the semantics and presentation of image changes are more difficult to interpret.

Users can choose to persist a copy of the rendered page. This enables them to return to previous versions if desired, but can create significant additional space overhead.

Key Cache Design Decisions

Representations of the Web pages a user visits are stored locally on the client machine. Although browsers contain caching mechanisms that save previously viewed content, they exist for performance reasons and not to enhance the user's view of the Web page. We create a parallel cache for DiffIE so that the application can control cache expiration and store multiple versions of the same page. Multiple versions allow users to view the changes not just since their last visit but also since other earlier visits.

Representations are tied to a particular URL and timestamp via a file naming scheme. Each representation is stored in a file named with a hash of the URL followed by the date. A person's notion of a Web page does not necessarily map directly to a unique URL; multiple URLs may map to the same content, or the same URL may be used to access very different content. There has been research into identifying mirrored content [4] and stripping form elements from URLs that produce the same content [12], and such functionally could potentially improve the user experience.

The maximum amount of disk space used by the cache is configurable. When the cache reaches capacity, the oldest pages are dumped to make room for new content. By default capacity is set to 500 MB. The maximum number

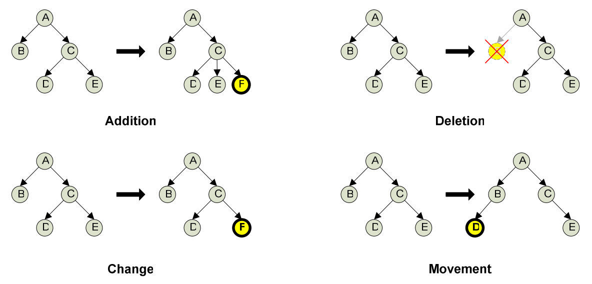


Figure 3. An illustration of the types of changes that can occur at the DOM level of a Web page.

of versions of the same Web page that are cached is also configurable, and set to five by default. Only 6% of all pages are visited more than five times, but not storing more than five visits per page can reduce the number of elements in the cache by 25% [22]. Cache size is further reduced by not storing duplicate versions of the same page when no change has occurred. Older versions of duplicate pages are stored as *pointer files* that indicate they contain the same content as the next version of the page. Including pointer files, the average size of an element in the DiffIE cache is 150KB. Given people visit on average from 14 to 97 pages per day [6, 14, 22, 24], the 500MB cache can support on the order of two or three months visitation history.

Previous research suggests that some classes of pages are visited several times in a single browsing session, and then not returned to again, while others are visited only once in a session but are very likely to be returned to later [2]. DiffIE could potentially increase its ability to cache pages that will be useful by biasing its cache policy towards those pages that are most likely to be revisited.

By default, secure pages (https://) are not cached, but this is primarily for the users' comfort. Storing hashed versions of page content locally does not pose a significant privacy risk. However, several users mentioned they appreciated not having secure pages stored, and one user turned the DiffIE system off when visiting sensitive content even though she knew no identifiable content was being stored.

Comparison Component

When a person visits a page for which there is cached content, DiffIE loads the most recent previous version of the page while the live content is downloaded, and compares and displays the differences between the live version and cached version when the download is complete. The current version can also be compared with earlier versions on demand. The comparison component of DiffIE is responsible for detecting and highlighting the changes.

Researchers have explored many different algorithms for detecting differences in Web pages [3, 9, 17, 18]. While any of these algorithms would work for DiffIE, we chose an approach with a small storage requirement that is fast at identifying changes at the expense of being able to identify fine-grained changes like node translation and transposition. Page comparisons can run in O(n) time, compared to algorithms such as Dynamo's [3], which runs in O(n^2) time.

Detecting Differences

The comparison component compares two pages by looking at how the current tree of hashes differs from the previous tree of hashes in a depth-first manner. Starting at the root node, DiffIE compares the pre-computed subtree hash of the live version and the cached version. If at any point the subtree hashes of the two pages are the same, DiffIE terminates comparison of the corresponding subtree, since identical hashes implies the content must not have changed.

For those subtrees that do not match, traversal continues down the branch. For any node, the following types of differences are identified:

- ***Addition.*** The hash of the text content of a node does not appear in the previous version of the page, and the node's parent in the current version of the page has more children than the previous version.
- ***Change.*** The hash of the text does not appear in the previous version of the page, and the node's parent in the current version has the same number of children.
- ***Deletion.*** The number of child nodes the node has in the current page version is less than the number of child nodes the node had in the previous version.
- ***Movement.*** The hash of the text content of a node in the current page appears in the previous version of the page, but with a different parent.

These four types of change are illustrated in Figure 3.

By default, only additions and changes are displayed to the user by DiffIE. Deletions are ignored because highlighting deletions would necessitate adding additional content to the current page and we did not want to interfere with the user's existing browsing experience.

Movement is ignored by DiffIE because it can be difficult to identify without semantic knowledge of the page. For example, for blogs or news feeds, where all of the content moves down when something is inserted at the top of the list, only the element that has been inserted conceptually changes. However, when the semantics of the position are important (e.g., a list of current best sellers), the movement of all of the elements in the list can be important even when no element has actually changed. Displaying additions captures the semantics of the news feed example, but not the best sellers example. The category could be broken down into more fine grained changes such as list insertion or reordering to capture these differences.

Highlighting Differences

Differences are highlighted via manipulation of the page's DOM. The background color of the changed node's parent is set by adjusting the node's style attribute. We worked with a designer to find a suitable highlighting color choice, and identified several important constraints. The color must be salient but not annoying (e.g., people found colors that were too bright distracting), culturally appropriate (e.g., red is often used to indicate a warning in the United States), and generally distinguishable from Web page background

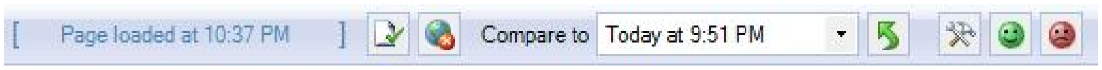


Figure 4. The DiffIE Web browser toolbar.

elements (e.g., pages may have blue elements, and white text, making blue highlighting hard to see). Most importantly, however, highlighted text must be easily legible. As the majority of Web pages use white as the dominant background color [5], most text is dark or black. However, text can also be white or light. Very little text is in the mid-range, so we selected highlighting colors in this range. Nonetheless, there will be occasions where the highlighting color is similar to the color of the highlighted text. These cases can be identified and text color inverted.

Performance
In practice we find that loading a cached page and computing and highlighting the differences takes on the order of tens to hundreds of milliseconds, depending on DOM complexity. Pages with a very complex DOM structure (more than 20,000 nodes) are ignored because of increased resources necessary for the computation and storage of such pages, but these pages are rare: a recent study found an average of 281 HTML tags on a pseudorandom sample of pages [16].

Although DiffIE is very fast, the comparison component does not trigger until the page has completed loading and the *load* event has fired. We observed while demonstrating DiffIE that this can make the application appear slow. Users often begin interacting with a page before all of the elements have loaded, and on occasion some page elements may not load for a very long time. DiffIE does not fire earlier is because it is important to have a stable DOM tree for comparison and stability is not guaranteed earlier. In fact, JavaScript and other non-HTML content may continue to modify the DOM even after the document is loaded.

Although waiting for the page to finish loading can delay the appearance of the highlighting, DiffIE does not interfere with the user's interactions prior to firing. The highlighting is merely a supplemental feature that appears when all of the content has loaded. We found that adding a status notification to DiffIE toolbar explaining the delay helped people understand what was happening and why.

Toolbar Component
The toolbar component of DiffIE is the portion of the application with which the user interacts. It can be seen in Figure 4. From left to right, the toolbar elements are:

Status Area [Page loaded at 12:13 AM]
The status area shows what DiffIE is currently doing, or what the current state of the page is. The status area also displays additional useful information like whether the page has been visited before, and if it has, when.

Highlighting Toggle Button
The highlighting toggle button turns highlighting on or off for the loaded Web page. The button has two states: 1) show what has changed in the page (green tick), or 2) hide what has changed in a page (red X). When the changes are hidden, the area reports the number of hidden changes.

This button was added to the toolbar because DiffIE's highlighting can sometimes be distracting. But several people reported using the toggle button for additional purposes. One user liked to turn the highlighting on and off to draw his attention to what had changed. Others used it when the highlighting interfered with their ability to use the page, most often when text color matched the highlighting.

Ignore Site Button
For some pages, DiffIE's highlighting is always distracting. In this case we allow the user to blacklist sites on a wildcard basis. By default, we blacklist any sites using the HTTPS protocol (secure sites). We chose not to blacklist intranet sites by default. Blacklisted pages are added to a list of ignored sites in the setting dialog, and can be removed from the blacklist from there.

Compare-To List Today at 12:08 AM
This list displays the visitation history for the page, and allows the user to select which previous version of the page the current version should be compared to. By default the most recent version is the comparison point.

Load Page Button
The page to which the current version is being compared can be loaded into a new tab for further inspection. As mentioned earlier, for space reasons page content caching is turned off by default and must be turned on via the settings dialog for this functionality to work.

Settings Button
There is also a button on the toolbar to open the settings dialog. The settings dialog allows the selection of highlighting color, the ability to turn DiffIE off, and control over the blacklisted sites and the size of the cache.

Feedback Buttons
The feedback buttons provide a mechanism for people to let us know when they have had a positive (green smiling face) or negative (red frowning face) experience with DiffIE. When a feedback button is clicked, the system generates an email message with a screenshot of the current page and information about the user's current settings. The user can modify the email to include additional content or remove private content. For privacy reasons, no other information than what was explicitly sent to us was collected.

While the DiffIE toolbar contains a lot of functionality for experimental purposes, we believe that the most important functionality can be easily encapsulated in a single button similar to the highlighting toggle button. The enhanced button would include an iconographic representation for the page loading, and provide access to a drop down menu for access to the additional features.

UNDERSTANDING DIFFIE

As mentioned earlier, we designed and deployed DiffIE in three stages. During the first stage, an initial prototype was used by the authors for their daily Web browsing for several months. This phase was used to identify bugs, problems with robustness, and performance issues. Stability and efficiency are very important when deploying a prototype that operates on all Web pages viewed in a browser.

Second, an improved version of DiffIE was demonstrated to over 300 Microsoft employees individually over a two day period, with a focus on the user experience. We observed during the second phase that people experienced confusion when nothing was highlighted until a Web page finished loading, and added a status message to the toolbar to explain what was happening in response. We also saw people became overwhelmed when too much content was highlighted, and thus surfaced in the toolbar the ability to easily turn highlighting off on a per-page basis.

Finally, to better understand how an awareness of Web content change might affect normal Web interactions, we asked eleven people to use DiffIE as part of their daily Web browsing activity for an extended period of time, and to share their experiences with us. This section focuses on results and insights from this phase. We discuss the study methodology, summarize some of the interesting and unexpected ways DiffIE was used, and describe several areas for improvement.

Study Methodology

Eleven people (5 women, 6 men) installed DiffIE on their primary work computers. All participants were Microsoft employees. Three were developers and the rest researchers.

Participants were encouraged to use the feedback buttons on the DiffIE toolbar to submit their positive and negative experiences. In total, we collected 51 pieces of feedback relating to the DiffIE user experience, 26 of which were positive, 23 negative, and 2 unlabeled. We did not log users' interactions for privacy reasons, and focus primarily on users' subjective experiences in our analysis.

After people had used the system for at least two weeks, we conducted semi-structured interviews with nine participants to get an in depth picture of how DiffIE was being used. In the interviews, we asked participants about their general experience with DiffIE, their understanding of the various user interface elements, and how often they felt the encountered the following different situations during the course of their use:

- Pages with nothing highlighted.
- Pages with too much content highlighted.
- Pages where highlighting drew attention to something unexpected they would not have noticed otherwise.

Of those pages that drew attention unexpectedly, we further asked whether the content attended to was important, interesting, or distracting. The answers participants gave to these questions can be found in Figure 5. The majority of people indicated that they never or rarely saw pages with nothing highlighted, that there was sometimes or often too much highlighted, and that there was often or always some unexpected highlighting (although it was rarely distracting).

During the interview, we also asked participants to revisit a self-selected sampling of pages from their browser history, five which they had visited on the same day as the interview, and five from a previous day. For each page we collected a screenshot and asked participants to answer the following questions:

- What was their intent when they last visited the page?
- Did they expect change since their last visit?
- Was their experience with DiffIE on loading the page during the interview positive, negative, or neutral?

In this way we observed people's interactions with a sample of 76 pages. Although during the general questioning most people indicated they never or rarely encountered pages with nothing highlighted (see Figure 5), 43 of the pages visited during the interviews had nothing highlighted. This is consistent with the rate of change found in previous research [2], and may suggest that highlighted instances were particularly memorable. Of the 33 pages where content had changed, people reported having a positive experience with DiffIE 20 times, a neutral experience six times, and a negative experience seven times. The different breakdown between positive and negative experiences observed during experience sampling compared to what was collected via the feedback mechanism suggests participants were more likely to send negative feedback or that the pages that participants chose to visit during our interviews were more likely to have interesting change.

We now summarize the ways that DiffIE enhances people's browsing experience, and explore how the DiffIE experience could be further improved.

How DiffIE Was Used

When people are first introduced to DiffIE, the two most common uses that come to mind are to find new content on sites known to change regularly (e.g., news sites or blogs) and to draw attention to changing content on sites that people monitor (e.g., stock quotes or sports scores). These two scenarios are probably popular because they represent instances where people actively seek out dynamic content.

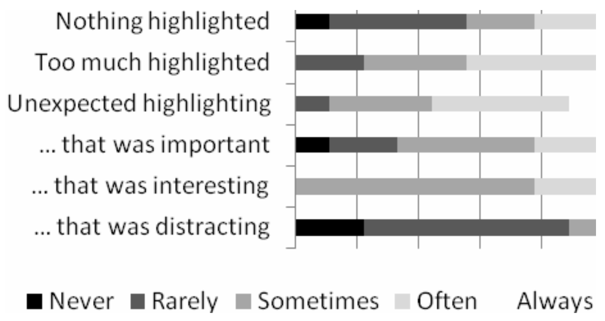


Figure 5. The frequency at which participants reported experiencing different scenarios while using DiffIE.

Based on what we observed during our study, however, it does not appear these scenarios are the most common use cases for DiffIE, nor does it appear that DiffIE is even always useful in these cases. Instead, DiffIE seemed to be most valuable to our participants when it revealed things about the pages they were visiting that they might not have otherwise been seen or understood. In this way, DiffIE seems to have the potential to expand the experiences that people are able to have with the Web.

We first discuss the two obvious use cases (*monitoring* and *finding expected new content*) in greater detail, and then describe several other ways that DiffIE was used to help participants see more content or better understand the pages they visit (including *finding unexpected important new content*, *serendipitous encountering*, *understanding a Web page*, *attending to activity*, and *editing*). Examples of these different uses are shown in Figure 6. We conclude with a discussion of how people said they might use the archived page content once they knew that the content had changed.

Monitoring

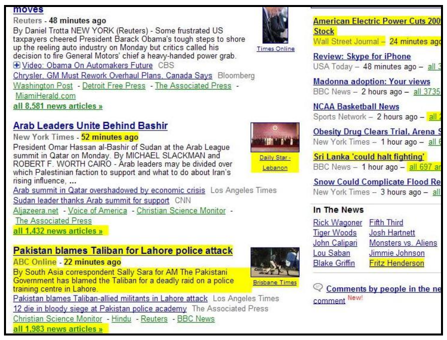
A common activity on the Web is monitoring a page for change [15]. For example, people may monitor a financial site to keep track of the latest stock prices, a message board while waiting for new postings, or online sports scores. Twenty percent of the pages discussed with us during the structured interviews were visited with a monitoring intent.

DiffIE provides value in such scenarios by highlighting and making it easy to quickly focus on the monitored content when it changes. For example, one participant reported enjoying using DiffIE to monitor the scores of several tennis matches because it drew her attention to those matches with updated scores. Another participant had monitored the search result page for the query "spock actor" to see if any of the sites contained information about Zachary Quinto, the actor who played Spock the recent *Star Trek* movie. He was thrilled when he pulled up the search result page during our interview and saw the tenth result highlighted and about Quinto, as he would not have noticed it otherwise even though he had been actively looking for it.

Interestingly, DiffIE did not always enhance the monitoring experience. Some pages are highly targeted to supporting monitoring. For example, many finance sites are specifically designed to show changing stock prices. Several people reported that DiffIE was annoying in these cases because it added clutter to a page already designed to draw attention to the changing value. The true value of DiffIE in monitoring situations appears to be for sites that are not explicitly (or not well) designed around the scenario.

Finding Expected New Content

Another popular use for DiffIE was to identify new content on sites that post new content regularly. Sixteen percent of the pages visited during the interviews had been previously visited to find new content that people expected to change during the day. For example, many people described how DiffIE drew their attention to recently posted content on news sites, blogs, or portals.



moves
Reuters - 48 minutes ago
By Daniel Trotta NEW YORK (Reuters) - Some frustrated US taxpayers cheered President Barack Obama's tough steps to shore up the reeling auto industry on Monday but critics called his decision to fire General Motors' chief a heavy-handed power grab.
Video: Obama On Automakers Future CBS
Chrysler, GM Must Rework Overhaul Plans, Canada Says Bloomberg
Washington Post - Detroit Free Press - The Associated Press - MiamiHerald.com
all 8,581 news articles »

Arab Leaders Unite Behind Bashir
New York Times - 52 minutes ago
President Omar Hassan al-Bashir of Sudan at the Arab League summit in Qatar on Monday. By MICHAEL SLACKMAN and ROBERT F. WORTH CAIRO - Arab leaders may be divided over which Palestinian faction to support and what to do about Iran's rising influence, ...
Arab summit in Qatar overshadowed by economic crisis Los Angeles Times
Sudan leader thanks Arab summit for support CNN
Aljazeera.net - Voice of America - Christian Science Monitor - The Associated Press
all 1,432 news articles »

Pakistan blames Taliban for Lahore police attack
ABC Online - 22 minutes ago
By South Asia correspondent Sally Sara for AM The Pakistani Government has blamed the Taliban for a deadly raid on a police training centre in Lahore.
Pakistan blames Taliban-allied militants in Lahore attack Los Angeles Times
12 die in bloody siege at Pakistan police academy The Associated Press
Christian Science Monitor - Hindu - Reuters - BBC News
all 1,983 news articles »

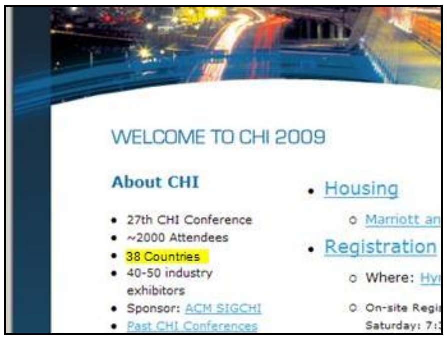
Times Online
Daily Star - Lebanon
Brisbane Times

American Electric Power Cuts 200
Stock
Wall Street Journal – 24 minutes ago
Review: Skype for iPhone
USA Today – 48 minutes ago – all 3
Madonna adoption: Your views
BBC News – 2 hours ago – all 3735
NCAA Basketball News
Sports Network – 2 hours ago – all 2
Obesity Drug Clears Trial, Arena S
New York Times – 1 hour ago – all 6
Sri Lanka 'could halt fighting'
BBC News – 1 hour ago – all 697 ar
Snow Could Complicate Flood Re
New York Times – 3 hours ago – all

In The News
Rick Wagoner Fifth Third
Tiger Woods Josh Hartnett
John Calipari Monsters vs. Aliens
Lou Saban Jimmie Johnson
Blake Griffin Fritz Henderson

Comments by people in the ne
comment New!

Monitoring / Finding Expected New Content

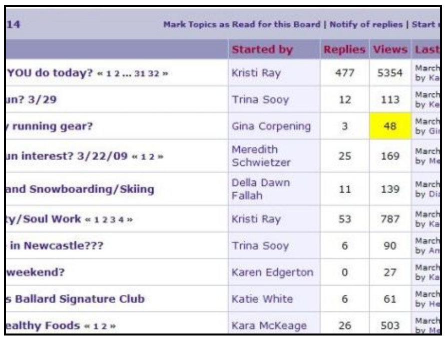


WELCOME TO CHI 2009

About CHI
- 27th CHI Conference
- ~2000 Attendees
- 38 Countries
- 40-50 industry exhibitors
- Sponsor: ACM SIGCHI
- Past CHI Conferences

- Housing
  - Marriott an
- Registration
  - Where: Hy
  - On-site Regi Saturday: 7:

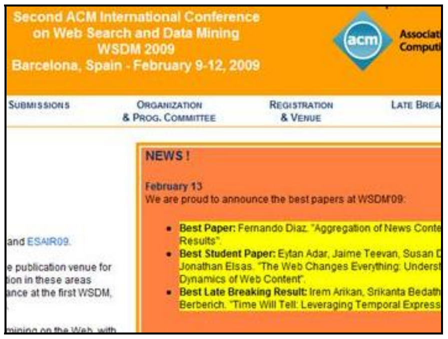
Serendipitous Encountering



14 Mark Topics as Read for this Board | Notify of replies | Start

	Started by	Replies	Views	Last
YOU do today? « 1 2 ... 31 32 »	Kristi Ray	477	5354	March by Ka
un? 3/29	Trina Sooy	12	113	March by Ke
running gear?	Gina Corpening	3	48	March by Gi
un interest? 3/22/09 « 1 2 »	Meredith Schwietzer	25	169	March by Me
and Snowboarding/Skiing	Della Dawn Fallah	11	139	March by Di
ty/Soul Work « 1 2 3 4 »	Kristi Ray	53	787	March by Ka
in Newcastle???	Trina Sooy	6	90	March by An
weekend?	Karen Edgerton	0	27	March by Ka
s Ballard Signature Club	Katie White	6	61	March by He
ealthy Foods « 1 2 »	Kara McKeage	26	503	March by Me

Attending to Activity



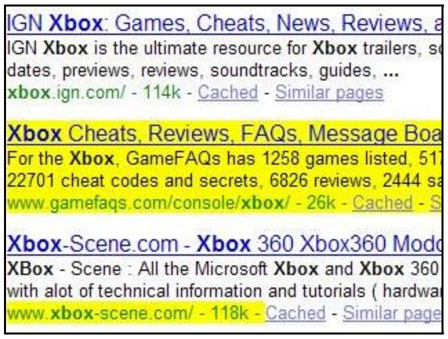
Second ACM International Conference on Web Search and Data Mining
WSDM 2009
Barcelona, Spain - February 9-12, 2009

Submissions | Organization & Prog. Committee | Registration & Venue | Late Brea

and ESAIR09.
e publication venue for
ion in these areas
ance at the first WSDM,
mining on the Web, with

NEWS !
February 13
We are proud to announce the best papers at WSDM'09:
- Best Paper: Fernando Diaz. "Aggregation of News Conte Results".
- Best Student Paper: Eytan Adar, Jaime Teevan, Susan D Jonathan Elsas. "The Web Changes Everything: Underst Dynamics of Web Content".
- Best Late Breaking Result: Irem Arikan, Srikanta Bedath Berberich. "Time Will Tell: Leveraging Temporal Express

Finding Unexpected Important Content

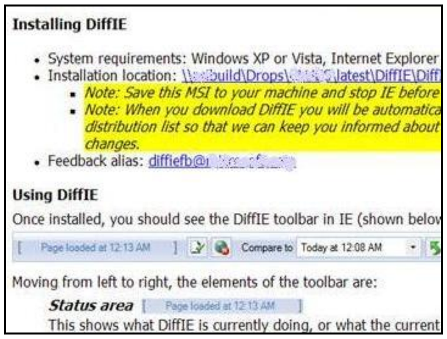


IGN Xbox: Games, Cheats, News, Reviews, a
IGN Xbox is the ultimate resource for Xbox trailers, s dates, previews, reviews, soundtracks, guides, ...
xbox.ign.com/ - 114k - Cached - Similar pages

Xbox Cheats, Reviews, FAQs, Message Boa
For the Xbox, GameFAQs has 1258 games listed, 51 22701 cheat codes and secrets, 6826 reviews, 2444 sa
www.gamefaqs.com/console/xbox/ - 26k - Cached - S

Xbox-Scene.com - Xbox 360 Xbox360 Mod
XBox - Scene : All the Microsoft Xbox and Xbox 360 with alot of technical information and tutorials (hardwa
www.xbox-scene.com/ - 118k - Cached - Similar page

Understanding a Web Page



Installing DiffIE
- System requirements: Windows XP or Vista, Internet Explorer
- Installation location: \\[redacted]build\Drops\[redacted]\latest\DiffIE\Diff
  - *Note: Save this MSI to your machine and stop IE before*
  - *Note: When you download DiffIE you will be automatica distribution list so that we can keep you informed about changes.*
- Feedback alias: diffiefb@[redacted]

Using DiffIE
Once installed, you should see the DiffIE toolbar in IE (shown below
[Page loaded at 12:13 AM] Compare to Today at 12:08 AM
Moving from left to right, the elements of the toolbar are:
Status area [Page loaded at 12:13 AM]
This shows what DiffIE is currently doing, or what the current

Editing

Figure 6. Examples of different ways DiffIE was used. The system was not always useful for scenarios involving the monitoring or finding of expected new content because sites are often designed to support these scenarios. Instead, DiffIE appeared particularly useful for scenarios where people do not currently expect change.

However, many participants also reported that DiffIE often highlighted too much information (as shown in the second row in Figure 5). This was most common for pages that people view periodically, but had not visited in a while. One participant reported dreading her first visit to an online news site in the morning because she knew everything would be highlighted, but enjoying subsequent visits. Another participant explicitly visited a news page first thing in the morning, without intending to read it then, so that DiffIE would be more useful on subsequent visits. For highly dynamic sites like news sites, DiffIE appeared to be most useful when the sites were monitored for change rather than visited to find new content.

Although we did not observe many cases of people finding new expected content on pages that change less frequently, a number of participants said they expect to find that DiffIE will be useful in the future to identify new content for pages they visit less frequently. For example, one participant visited a colleague's academic publication listing during his interview, and while there were no changes highlighted at the time, he expressed the expectation of being able to return in several months and find new papers of interest.

Finding Unexpected Important New Content

A less common scenario particularly delighted users was when DiffIE drew attention to important but unexpected new content. As shown in Figure 5, people reported often having their attention drawn to unexpected content, and some of this time they found this content to be important.

An example is shown in Figure 1, where the phone number listed on the researcher's home page has changed. Another example in Figure 6 highlights the best paper awards on the WSDM conference page. One of our participants reported that DiffIE enabled him to notice the price of an airline ticket had risen, and another learned of an event she might want to attend when it was posted on an online calendar.

Serendipitous Encountering

Most of the time when people's attention was drawn to unexpected content, that content was not actually important. However, participants did not appear to find having their attention drawn to unimportant content distracting, and instead reported that it was often interesting (see Figure 5). In fact, several people even said it was interesting to be made aware of the existence of advertising in locations they would not have expected.

In this way, DiffIE appeared to support serendipitous information encounters [11]. For example, one subject followed a highlighted link during his interview because he thought it sounded interesting, but he his motivation in visiting the page was not to find the information and he would not have noticed it otherwise.

Understanding a Web Page

These unimportant and unexpected changes were often interesting to people because they provide new way of understanding the Web pages. When asked if they expected a page to change when visiting it, participants sometimes answered that they previously did not expect change on that particular page, but now did as a result of DiffIE.

For example, many people were surprised by how often search results changed (as shown in Figure 6). One participant used a search engine that provided search history functionality, but was not aware of the functionality until DiffIE started highlighting his changing history. That same participant said he used to think a block of content with different cells for different sports teams contained information that all changed at the same time, but that he now recognized that the information about teams changed at different rates. Instead of viewing content as a block, he began to view it as a collection of separate entities. Another was intrigued to notice that new postings to an audio software news site that he visited about once a week were not always listed chronologically.

Attending to Activity

DiffIE also made some Web content was not thought to be useful by itself to become useful in context. One way in particular that we observed this happening was that it enabled people to attend to the activity by other users on a Web page. Many Web pages are structured so that visitors leave footprints as they interact with them. Pages may have counters that increment on page loads, comment posts, or thread visits, or lists of names that show who is logged on, visiting, or reading content. While this information provides some value when static, it appears particularly valuable when users can see that it has changed because it enables people to understand what others are doing.

For example, one participant reported using DiffIE to monitor activity on a message board she frequented. She found that when the counter representing how often a thread was visited would increment, she would think about the thread again even if there had not been a new post to the thread (see Figure 6). Another participant used DiffIE to know when a postponed tennis match she was interested in monitoring had started updated again.

Editing

A number of participants reported DiffIE was particularly useful when they were editing Web content they were responsible for publishing. Although in these cases participants knew exactly what changes they expected to see highlighted following an edit, they found DiffIE was very useful in drawing their attention to the changes they had made so they could confirm they appeared as expected. One participant mentioned he was trying to edit two pages in a similar manner at the same time, and that being able to quickly find the content in both without doing a visual seek was particularly useful.

People also found DiffIE very useful when they were in charge of producing the content of a page but not in control of actually changing it. Two participants had used DiffIE to easily see if the person who was tasked with making the requested edits had actually done so.

Using the Archived Content

Once they were made aware of the existence of change, there were a number of ways people suggested the archived content could be of additional value. Several people mentioned it could be used to understand the type of change (e.g., is the content new or different?) or direction of change (e.g., did the price go up or down?). It could also be used to support the re-finding of lost content (e.g., what was the news story that was posted here yesterday?). The archived page could also be useful to view the changes made while editing a page or to revert to an earlier version.

Areas for Improvement

Based on our observations of how people used DiffIE, there are three main ways that the application can be improved. One is to better expose the changes that occur to a page, another is to explore the exposition of different types of change beyond simple additions and changes, and the third is to allow people to see change they have not personally experienced. All three are discussed in greater detail below.

Ways of Exposing Change

With DiffIE, we chose a very simple way to notify the user that the content of the page had changed: We highlighted all of the changed content. However, this simple approach has its flaws. Many participants reported finding too much highlighting, either because too much content had changed, or because content very high in the DOM structure had changed and the highlighting of that high-level element lead to significant highlighting.

One way to address this would be to not highlight changes in these cases (for example, if more than fifty percent of the page is going to be highlighted). However, this would make it difficult for users to distinguish between pages where nothing has changed and too much has changed. This could be addressed by briefly presenting and then fading highlighting, or by showing the amount of new content iconographically in the DiffIE toolbar.

Another way to reduce the appearance of over-highlighting would be to only show changes that are likely to be of value to the user. Adar et al. [2] found that people's revisitation rates can be used to help identify DOM elements that are likely to be interesting because they change at a similar rate. However, the number of people who enjoyed having their attention drawn to unexpected changing content suggests there may be something lost in the experience by filtering which changes are shown even if interesting change can be accurately identified.

Many people also wanted more information about changed content than merely that certain DOM elements had changed since the last visit. Additional information requested included details about much the page differed from older versions. In order to get a longer term historical picture of how a page changes, two people suggested the notion of change decay. The live version of a page could not be merely compared to the last version stored in the cache, but to all previous versions. Older changes could be highlighted faintly and fresh changes highlighted brightly. This would help prevent the change context from entirely shifting each time a person re-loads the page.

There are a number of ways the details of how a page differed from the previous version could be exposed (see also [9]). The most straightforward, which is currently supported, is to allow the user to load the previous version and make the comparison themselves. However, it can be easy to miss changes when viewing pages side by side. Changes could also be exposed via Microsoft Word's change tracking, by displaying the earlier content on hover, or by building on ideas from related research (e.g., highlighting and rewinding areas, as done in Zoetrope [1]).

Types of Change Exposed

One of the reasons people wanted to view the previous content is that although we highlight additions and changes, it is not always apparent how the content differs from the previous version; has the content changed, or is it new? Several participants mentioned wondering about this, and one actually changed the highlighting colors for the two types of changes so that they would be highlighted differently. She suggested it might be interesting to highlight new content in a very salient color, and highlight changed content in a less salient version of the same color.

Several participants asked for additional types of changes (specifically moves and deletions) to be displayed. The challenge here is how to represent them. One participant suggested that deletions could be represented through a small amount of highlighting to empty content, with the removed content visible upon hover.

People also expressed interest in seeing how other types of content, and in particular images, change. This could be done by earmarking or outlining images that have changed. We have explored rendering different versions of the same page as images and performing a pixel-wise comparison of the old and new versions of the page to show how the page has changed as is visible to the user. However, while this captures all manner of change, it is not particularly robust to small movements.

It might also be interesting to capture a semantic notion of change. For example, numbers that increase or decrease could be highlighted have an arrow drawn next to them representing the direction of the shift, much in the way stock tickers do. Conceivably even changes to facts could be captured and represented using the Semantic Web.

Exposing Unseen Change

We observed some people purposely visit pages to establish a baseline version in their cache. In these cases, their visits were not driven by the current version of the page, but by their interest in the future differences. This suggests DiffIE could potentially benefit from a richer cache than one that merely contains copies of the pages a user has visited.

One way DiffIE could avoid the cold start problem would be to pre-populate the user's cache with pages from the user's browser history during installation. In addition to enabling future scenarios, this would provide immediate gratification as the new user first explores the application.

We could also show changes not just between content the user has viewed, but also with other versions of the page. This could be done by proactively crawling explicitly requested, previously visited, or globally popular pages, or by using versions stored in an archive service like the Way Back Machine or a search engine Web cache. DiffIE could also support social scenarios where users and organizations share peer-to-peer or proxy caches to enable awareness of changes relative to what the group has collectively seen. Cached content can be used retrospectively, to provide the user with an understanding of how the pages change over time, as well as prospectively, to identify for the user when interesting change occurs to a page of interest.

As we have seen, there are many ways DiffIE might better help people better understand changing Web content. It is notable, however, that even though there is significant room for improvement, all participants chose to continue using the system following the completion of the study.

CONCLUSION

In this paper, we have looked at how Web browsers can better support the fact that people work in a dynamic information environment. We presented *DiffIE*, a browser plug-in that caches the pages a user visits and then highlights the way those page have changed when the user returns to them. We described challenges to building such a system, such as identifying change types and presenting them in a salient but non-distracting way. By deploying DiffIE and observing its use, we found a number of unexpected ways that DiffIE helps people have a richer experience with the Web content they interact with, and we explored several interesting areas for improvement.

REFERENCES

1. Adar, E., M. Dontcheva, J. Fogarty, and D. S. Weld. Zoetrope: Interacting with the ephemeral Web. UIST '08, 2008, 239-248.
2. Adar, E., J. Teevan, and S. T. Dumais. Resonance on the Web: Web dynamics and revisitation patterns. CHI '09, 2009, 1381-1390.
3. Borodin, Y., J. P. Bigham, R. Raman and I. V. Ramakrishnan. What's new? – Making Web page updates accessible. ASSETS '08, 2008, 145-152.
4. Bharat, K. and A. Broder. Mirror, mirror on the Web: A study of host pairs with replicated content. WWW '99, 1999, 1579-1590.
5. Bucy, E., A. Lang, R. Potter, and M. Grabe. Formal features of cyberspace: Relationships between Web page complexity and site traffic. *JASIS,* 50(13):1246-1256, 1999.
6. Catledge, L. D. and J. E. Pitkow. Characterizing browsing strategies in the World-Wide Web. WWW '95, 1995, 1065-1073.
7. Cho, J. and H. Garcia-Molina. The evolution of the Web and implications for an incremental crawler. VLDB '00, 2000, 200-209.
8. Cockburn, A. and B. McKenzie. What do Web users do? An empirical analysis of Web use. *International Journal of Human-Computer Studies*, 54(6):903-922, 2001.
9. Douglis, F., A. Feldmann, and B. Krishnamurthy. Rate of change and other metrics: A live study of the World Wide Web. USENIX Symposium on Internet Technology and Systems, 1997.
10. Fetterly, D., M. Manasse, M. Najork, and Wiener, J. A large-scale study of the evolution of Web pages. WWW '03, 2003, 669-678.
11. Foster, A. and N. Ford. Serendipity and information seeking: An empirical study, *Journal of Documentation*, 59(3): 321-340, 2003.
12. Hupp, D. and R. Miller. Smart Bookmarks: Automatic retroactive macro recording on the Web. UIST '07, 2007, 81-90.
13. Jatowt, A., Y. Kawai, H. Ohshima, and K. Tanaka. What can history tell us? Towards different models of interaction with document histories. HT '08, 2008, 5-14.
14. Jones, R. and Fain, D. C. Query word deletion prediction. SIGIR '03, 2003, 435-436.
15. Kellar, M., C. Watters, and K. M. Inkpen. An exploration of Web-based monitoring: Implications for design. CHI '07, 2007, 377-386.
16. Levering, R. and Cutler, M. The portrait of a common HTML Web page. DocEng '06, 2006, 198-204.
17. Liu, L., C. Pu, and W. Tang. WebCQ: Detecting and delivering information changes on the Web. CIKM '00, 2000, 512-519.
18. Liu, L., W. Tang, D. Buttler, and C. Pu, Information monitoring on the Web: A scalable solution. WWW '02, 2002, 263-304.
19. Nadamoto, A. and K. Tanaka. A Comparative Web Browser (CWB) for browsing and comparing Web pages. WWW '03, 2003, 727-735.
20. Ntoulas, A., Cho, J., and Olston, C. What's new on the Web? The evolution of the Web from a search engine perspective. WWW '04, 2004, 1-12.
21. Obendorf, Hartmut, H, Weinreich, E. Herder, and M. Mayer. Web page revisitation revisited: Implications of a long-term click-stream study of browser usage. CHI '07, 2007, 597-606.
22. Tauscher, L. and S. Greenberg. How people revisit Web pages: Empirical findings and implications for the design of history systems. *International Journal of Human-Computer Studies*, 47(1):97-137, 1997.
23. Teevan, J., E. Adar, R. Jones, and M. A. Potts. Information re-retrieval: Repeat queries in Yahoo's logs. SIGIR '07, 2007, 151-158.
24. Weinreich, H., Obendorf, H, Herder, E., and Mayer, M. Not quite the average: An empirical study of Web use. *TWEB*, 2(1), 2008.