A Dynamic Navigation Guide for Webpages
Jawed Karim, Ioannis Antonellis, Varun Ganapathi, Hector Garcia-Molina
Department of Computer Science
Stanford University
{jawed, antonell, varung, hector}@cs.stanford.edu

ABSTRACT
Navigating websites is often a frustrating process: Website visitors, despite their widely varying and individual information-seeking needs, must contend with static, general-purpose link structures that have been set in place by website owners. Because many visitors tend to browse for the same content, they are individually repeating the same navigation activity. Visitors would benefit from being able to take advantage of the collective search and discovery work that has already been performed by other visitors. Although many attempts have been made to improve website navigation by tapping into the “wisdom of the crowds”, the currently available approaches suffer from maintenance, usability, and user interface integration issues. We present a navigation guide for websites that provides visitors with helpful suggestions based on their browsing activity and the browsing activity of prior, similar visitors. Our navigation guide does not require any downloads, can be easily added to websites by website owners, and automatically remains up-to-date.

AUTHOR KEYWORDS
Web navigation, Recommendation Systems, Adaptive User Interfaces

ACM Classification Keywords
H3.3 [Information Storage and Retrieval]: Information Search and Retrieval, H.5.2 [Information Interfaces and Presentation]: User Interfaces

INTRODUCTION
The navigation interfaces of most websites are static and are designed to appeal to the “average” visitor, giving equal weight to a wide range of possible information-seeking scenarios. The individual website visitor, who has a very specific need, is forced to adapt her navigation path to the arbitrary link structures set in place by website owners. As a result, the information discovery process on websites is often time-consuming, circular, and repetitive. If a visitor's browsing path spans across several websites, her task is further complicated by having to adapt to different website link structures, layouts, and designs along the way. A university student seeking the gym’s opening hours, for example, might have to navigate three entirely different websites, each with a distinct organization: First the university's main website, followed by the athletics department, followed by the facilities website. Each time, a visitor has to perform a context switch to reorient himself within a new and unfamiliar navigation structure.

CURRENT METHODS
Faced with these limitations, visitors have increasingly relied on external website navigation aids, such as search engines. Instead of finding the Stanford gym hours by starting at www.stanford.edu and traversing the link path “Life on Campus → Recreation and Fitness → Facilities → Recreation Hours”, visitors might instead perform a Google search for “stanford gym hours”, or “gym hours site:stanford.edu” (to limit search results to local webpages on the stanford.edu domain), in order to arrive on the desired page directly.

In a response to this trend, many large websites (including most university websites) display a search form on their front webpage that performs a domain-limited search of the respective university website. Although using search engines for navigation can lead to shortcuts in some cases, it does not always lead to satisfactory results, as search engines must rely on the proper pages being indexed and the proper search keywords being associated with those pages. Using our example, searching for “stanford gym hours” on Google does not lead to any useful result pages.

In the same way that visitors have resorted to using search engines to search for specific content on other websites (i.e., content that they already knew existed), they have also relied on 3rd party social link aggregation websites, like digg.com and del.icio.us, to discover interesting content on other websites (i.e., content that they didn't previously know existed). These link aggregation websites enable millions of users to collect, tag, and share their favorite bookmarks, providing a valuable service for making visitors aware of newly discovered and popular content. For example, a search for “Stanford” on del.icio.us reveals that del.icio.us users consider Stanford's iTunes website and Stanford's iPhone programming class website, both of
which offer free video lectures, to be among the most popular websites on the stanford.edu domain. Yet despite the popularity of these websites, most visitors to Stanford's main webpage and computer science webpage will remain unaware of this popular content, unless they first reference an external 3rd party social link aggregation websites.

While social link aggregation websites have proven useful in these isolated cases, they are not practical as persistent navigation tools: They can only cover a small subset of the Web and may be out of date. From a user interface integration standpoint, the fact that they are external websites results in a disjointed browsing experience for visitors. This issue is partially addressed by browser plugins offered by link aggregation websites (for example the FireFox add-on “Delicious Bookmarks”). But browser plugins have generally not seen widespread adoption because they are browser-specific and because of the high usability barrier imposed by having to download, install, and configure them.

Given these limitations of current navigation methods, visitors spend much time trying to locate content on websites, often settling for sub-optimal content or missing critical information entirely.

**THE “WISDOM OF THE CROWDS”**

It has been shown that the average opinion of a group of individuals can be used as an effective evaluation method. In 1907, Francis Galton demonstrated that by averaging the independent estimates of the weight of an ox from many observers at a country fair, a more accurate value could be obtained than from individual experts [1]. On the web, the wisdom of the crowds can be applied to guide visitors in their navigation of webpages. Much available data suggests that Web use follows a power law distribution: A majority of visitor activity on a website is concentrated around a narrow range of content [2]. It is therefore likely that among the complete set of a website's visitor sessions (the list of pages visited by each visitor), many of the navigation paths for a given information-seeking scenario are frequently repeated. For example, if visitors reach a faculty member's publications webpage from a conference website, they are likely to gravitate towards a small number of publications that relate to the conference topic. By recognizing such a trend, future visitors who follow a similar navigation path to arrive at the faculty member's publications webpage could then be guided towards those publications.

We can build a list of recommendations for any information-seeking scenario by analyzing web traffic data in terms of its three basic components: content (such as links and webpages), visitor actions (such as clicking on a link or visiting a webpage), and visitor attributes (properties that describe visitors, for example their geographical location or previous navigation paths). If we consider every visitor action on a piece of content to be an implicit vote of interest by that visitor for that content, then by aggregating these actions by visitors' attributes, we can build a list of recommended content for future visitors with similar attributes. These recommendations can be used by visitors to more quickly locate the most relevant content on a webpage.

**A DYNAMIC NAVIGATION GUIDE**

In this paper we present a navigation guide that provides visitors with helpful navigation suggestions, based on the actions taken by previous visitors with similar attributes to the current visitor. Our navigation guide addresses the user interface integration, usability, and deployment issues encountered by other methods, and offers the following features:

- The navigation guide is displayed in a floating overlay window at the bottom of the browser, allowing visitors to interact simultaneously with the webpage as well as with the navigation guide.
- The navigation guide is implemented in Javascript and Dynamic HTML and does not require any downloads or installations from visitors.
- The navigation guide can be instantly deployed by website owners on webpages through the insertion of a short snippet of Javascript into the webpage's HTML code.
- The navigation guide automatically remains up-to-date without maintenance or intervention from website owners.
- The navigation guide can dynamically alter HTML elements of the webpage on which it is placed to alert visitors to specific content.

![Figure 1](image-url)

Figure 1. Upon visiting this faculty member's website, visitors are presented with a navigation guide, located at the bottom of the browser window (see Figure 2 for a detailed view).
The navigation guide has several sections for displaying information. The top section displays common subsequent links visited by visitors who reached the current webpage through the same navigation path as the current visitor. For example, in Figures 1 and 2 the navigation guide recognizes that the current visitor reached this faculty member's webpage from a webpage called “Current Members”. Based on this information, the navigation guide displays the most frequently visited subsequent destinations for other visitors who also reached the faculty member's webpage using this navigation path. Next to each link is a bar graph that shows the relative probabilities of each link visit. As can be seen in the navigation guide display, visitors who reach the faculty member's webpage through this path are most interested in the faculty member's FAQ, her publications, and her most senior grad student. This is consistent with what one would expect, as those visitors who reach the faculty member's webpage from a list of research group members (the “Current Members” webpage) are less likely to be familiar with the faculty member, and are seeking to gain a general overview. Like the top section, the bottom section of the navigation guide also shows the most frequently visited outbound links from the current webpage, but does not take into account visitors' prior navigation paths. In combination, the two sections give visitors a quick overview of which information is most important under different information-seeking scenarios.

The navigation guide's interface is displayed within a floating window overlay that scrolls vertically along with the webpage itself, ensuring that all of the webpage's contents remain visible, and enabling visitors to interact with the guide and the webpage simultaneously. The navigation guide can be minimized to the bottom of the browser by clicking on an empty space within the navigation guide window, and the window's background color can be configured to be translucent, so as to show the webpage contents behind it. The navigation guide is also able to dynamically alter elements of the underlying webpage to call attention to certain parts of the content. For example, when a visitor moves the mouse pointer over the links in the navigation guide, the corresponding links in the website are highlighted in order to show the links in their context with respect to the surrounding content.

Having access to these pieces of information is helpful to visitors because it allows them to understand at a glance which information is important on a webpage without having to peruse the entire webpage, or having to visit and evaluate each link individually.

**HOW IT WORKS**

One of the key contributions of our navigation guide implementation is that it integrates the traditionally separate stages of web traffic data collection, data analysis, and presentation into a single step, thereby radically simplifying deployment of the navigation guide. The navigation guide can be deployed by website owners through the simple insertion of a short snippet of Javascript code into a website's HTML code. The snippet, shown in Figure 3, consists of a single line of Javascript code that references a more complex set of scripts (from hereon referred to as the “script”) on a centrally hosted Navigation Guide Server, located at [http://tags.stanford.edu/](http://tags.stanford.edu/). Once a webpage has been instrumented with this snippet, no further maintenance or intervention from website owners or website visitors is required to keep the navigation guide up-to-date, and visitors to the webpage will immediately begin to see navigation suggestions based on previous visitor actions.

When a visitor arrives on an instrumented webpage, the remotely hosted script executes automatically in the visitor's web browser and monitors the visitor's actions on the webpage. Monitored actions include the loading of a webpage (logged when the script loads) and clicks on specific links on a webpage (logged at the time the click occurs). These actions are sent in real-time to the Navigation Guide Server, where they are stored in a centralized database.

```<script src="http://tags.stanford.edu/tags.js" type="text/javascript"></script>
```

**Figure 3. Inclusion of this snippet activates the navigation guide on a webpage.**

The same script that monitors and collects these actions in the visitor's browser also displays the navigation guide in the browser. The contents of the navigation guide are queried by the script in real-time from the Navigation Guide Server. Since the Navigation Guide Server computes these contents on the fly, based on the most recent visitor actions, the guide is updated in real-time to reflect the current visitor's actions.
actions that have been received, the navigation guide is always up-to-date and reflects the latest actions that have occurred on the webpage. For example, if a sudden news event causes a specific link on the webpage to become popular, the change in activity will immediately influence the contents of the navigation guide, which will begin to elevate the significance of that link.

The navigation guide server intercepts link click actions by overriding and adding additional logging functionality to Javascript's event listeners 'mousedown' and 'click'. Capturing visitor action data via Javascript directly at the client from within the browser allows us to record action data at a greater level of detail than if the actions were parsed at the server from a webserver access log. Webservers are only aware of page hits, but not of link clicks. Many browsers mask the "referrer URL" sent to webservers for privacy reasons, preventing webservers from knowing from which link a page hit originated. Our approach ensures that the link that a visitor clicks on is captured, including the link's location relative to the surrounding content within the HTML document, whereas webservers do not have access to this information.

RELATED WORK
The study of mining the behavior of past website visitors for useful patterns is a well established field known as Web usage mining. Pitkow and Sarukkai proposed algorithms that mine historical visitor actions for navigation patterns to predict the actions of future visitors [3] [4]. These methods have been used by tools for website owners to create navigational aids for website visitors. One of the earliest such tools, called “Footprints”, creates a link graph visualization of the paths taken by visitors who have been to a website before, with the links color-coded to show the frequency of use of different paths [5]. Perkowitz proposed a technique for mining webserver access log data to create site maps, or index pages containing collections of links that prior visitors' navigation behavior suggests are related [6]. These navigation aids are difficult for visitors to integrate into their browsing activity. While link graphs and site maps may be useful for simple and small websites, they quickly become unmanageable as the number of webpages and links increases. For website owners, these navigation aids are also difficult to generate, publish, and maintain. Mobasher, who proposed mining historical visitor actions for sequential navigation patterns in order to suggest them to current visitors, breaks down the operation of a usage-based Web recommendation system into three phases: data collection and preparation, pattern discovery, and recommendation [7]. Data collection and preparation is a process that usually requires access to webservice access logs. Pattern discovery involves the parsing and analysis of the visitor activity data contained in logs, and recommendation is the real-time stage of displaying a navigation aid to visitors. These disconnected processes are cumbersome to set up and automate across a large number of websites or webservers, and are likely to lead to inconsistencies and outdated navigation aid outputs.

CONCLUSION & FUTURE WORK
We have presented a navigation guide for website visitors that can be easily integrated into websites by website owners, requires no downloads or installations from website visitors, and remains automatically up-to-date. While the individual components of our work have been applied in other areas, we have integrated them in a way that makes a navigation guide solution feasible and practical for widespread deployment.

One area in which we are focusing future work is to consider more visitor attributes in the navigation guide's recommendation generation. For example, we have found that the search engine keywords used by visitors to arrive on a webpage are highly predictive of their subsequent actions. In an analysis of three months of Web traffic to the Stanford Graduate School of Business's website, we found that visitors who arrived on the webpage http://gsb.stanford.edu/ via the search terms “mba program” behaved very differently from the population that used the search terms “finance phd”, though the behavior within each of the two groups was very consistent. We hope that eventually our concept can be expanded to capture and take into account these and many other visitor attributes to provide increasingly customized recommendations.

REFERENCES
http://doi.acm.org/10.1145/383694.383707