FINDING AND REFINDING WEB PAGES IN CONTEXT
A Tree-based Model of Web History

David Briffa and Chris Staff
Department of Intelligent Computer Systems, University of Malta, Tal-Qroqq, Msida MSD 2080, Malta

Keywords: Web history navigation, Revisiting web pages in context, Automatic query generation, Global reconnaissance, Tree-views, User modelling.

Abstract: A modern challenge for the World Wide Web (Web) is not of just finding information without getting 'lost' in hyperspace, but also re-finding it efficiently. Web Nav is an integrated navigation system that combines both history and page recommendations into one context based tool. Web Nav's framework signifies a paradigm shift in the viewing of history from a linear stack-based system to a hierarchal tree-based system. Web Nav was evaluated qualitatively and quantitatively, analysing 13 users' activity over a seven day period. The results are mixed but there is sufficient evidence to suggest tree-based views of history can be beneficial: to allow users to revisit web pages in context; to show user sessions as trees; and to automatically generate queries based on session contexts to recommend web pages.

1 INTRODUCTION

In this paper, we tackle two major Web Navigation problems: finding and re-finding information. Modern search engines generally do not take the session *context* into account (i.e., a group of web pages related to some task being performed by the user). Two different users searching for the term 'jaguar' will be presented with the same results set, even though in the session contexts, one user has been visiting web pages related to jaguar the animal and the other has been visiting pages related to Jaguar the car. We also tackle the problem of re-finding information. Between 58% and 81% of page visits are page re-visits (c.f. section 2.3), so mechanisms for organizing and easily accessing already found information are important.

In our opinion, representations of history should preserve the contextual structure in which web pages are visited. Context can also help to automatically construct queries to find more relevant information. Web Nav (Briffa, 2010) incorporates context and global reconnaissance into a tree-based model of history.

2 FINDING INFORMATION

Finding information on the Web can be seen as a combination of *searching* and *browsing* (Herder, 2006). Searching involves submitting a query to a search engine, and browsing involves navigating between pages using hyperlinks (Herder, 2004). One of the main problems on the Web is the 'Lost in Hyperspace' problem, where the user starts browsing, and finds herself 'lost' in terms of where she has been and where she intends to go. Adaptation tools can give a sense of direction such as through page recommendations or direct guidance (Brusilovsky, 2001).

In a Web Browser, searching is catered for through search engines and embedded widgets that quickly return results for a query. Search can be improved by building user models from visited pages. *PowerScout* (Lieberman, Fry, & Weitzman, 2001) builds a user model from the pages visited and constructs a query that is submitted to a third party search engine, returning page recommendations.

A Web user must constantly choose between browsing links or initiating a search for pages. This is directly related to Information Foraging Theory (Pirolli & Card, 1999). According to this theory, users try to maximize the ratio of energy spent and information gain. An automated approach may use both neighbouring pages and a search engine to determine which strategy will be more successful.

FollowMyLink (Briffa, 2009) integrates search and browsing by turning user selected text on a Web page into hyperlinks on-the-fly. The user is taken to the top ranking page in the results set following a

Briffa D. and Staff C..
FINDING AND REFINDING WEB PAGES IN CONTEXT - A Tree-based Model of Web History.
DOI: 10.5220/0003401004260429
In *Proceedings of the 7th International Conference on Web Information Systems and Technologies* (WEBIST-2011), pages 426-429
ISBN: 978-989-8425-51-5

query generated from contextual information. FollowMyLink maintains separate user models as a user browses, using page relations and browsing behaviour to decide which user model to update. When a user selects text on a Web page and invokes FollowMyLink, a query is automatically generated from the user model which is updated after each link traversal. Y!Q (Kraft, Maghoul, & Chang, 2005) uses only the context supplied by text surrounding a user selection. Y!Q may take the user to a page the user has visited recently, whereas FollowMyLink takes the user to a previously unseen relevant page.

2.1 Re-finding Information

Apart from searching and browsing, a third element to navigation is *backtracking*. This involves going to a previously visited page. A browser supports a number of tools for both short term and long term backtracking, such as the Back/Forward buttons, history lists, bookmarks, and the History window. Backtracking in Web navigation is important, considering the number of page visits on the Web that are actually re-visits. (Tauscher & Greenberg, 1996) give a figure of 58% in 1996, while (Cockburn & Mckenzie, 2000) calculated a value of 81% in 2000. Herder estimates that 74% of page visits are revisits (Herder, 2006). Reasons for revisiting pages include: checking if information has changed; authoring a page; exploring it further; or the page is on a path to another revisited page (Tauscher & Greenberg, 1996).

Of particular interest are the latter two cases. First, the need to explore pages further may result in a page being bookmarked. Although bookmarking is efficient with regards to space and organization, any context gathered from previous pages is lost from one session to another, and hence when returning there is no context or user model available for the page in question. Second, the mention of a path raises questions on whether bookmarking one page is enough, or whether pages the user followed on the way to the bookmarked page should also be persisted. Empirical evidence shows that the path is not only important, but should also be a factor in history mechanisms (Teevan, 2004), with '*waypoints'* such as page titles and descriptions to be considered as supplementary metadata to the path (Capra & Perez-Quinones, 2003). Tauscher and Greenberg provide a comprehensive analysis of the types of history mechanisms available, and focus particularly on the stack based history list as applied to the Back and Forward buttons. A drawback of this approach is that the resulting history list does not contain all visited pages, due to how all pages above the stack pointer are removed during backtracking (Cockburn & Jones, 2000). Cockburn and Jones also suggest that users themselves have a skewed understanding of how the history list works. Tauscher and Greenberg refer to context sensitive subspace history lists, which suggest the grouping of pages in history into a subspace based on context.

Other systems may give a view of history in the form of maps or views. *WebView* (Cockburn, A., Greenberg, S., McKenzie, B., Jasonsmith, M., & Kaasten, S, 1999) and *WebNet* (Cockburn & Jones, 2000) generate overviews of the users browsing path. As the views are not persisted to memory, they are not useful for long-term backtracking. *CZWeb* (Fisher, B., Agelidis, M., Dill, J., Tan, P., Collaud, G., & Jones, C., 1997) uses a fish-eye view.

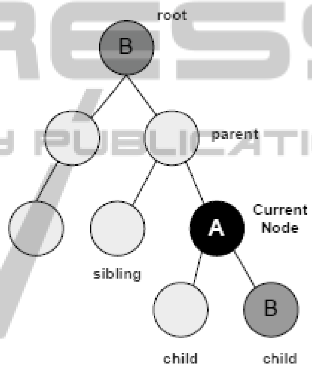

Figure 1: Tree Structure for a hypothetical path.

3 WEB NAV

We built a Mozilla Firefox Extension for the persistence and use of context-based paths. As Firefox uses a stack-based paradigm for its tab history, we designed a separate framework so Web Nav can store its own copy of navigation history as a tree-based structure in an SQLite database. The overall design rationale is to use this framework as a basis for providing both context-based history information and context-based page recommendations so that users are never 'lost' or unable to backtrack to a specific page.

3.1 Web Nav History

Web Nav tracks visited pages in a *Context Model* through tab-based storage using the Mozilla Session Store API. We identify the corresponding node of a page by checking the Session Store. Paths are built by making connections between nodes to reflect the type of traversal performed. Figure 1 shows a typical tree structure, where node A is the current page.

The tree shows the page where a link was clicked to visit page A and other branches in the path. Paths are acyclic. An instance of page B is added as a child to page A, even though it already exists as the root. A page may be seen in multiple contexts, and in multiple instances within the same context.

We distinguish between two movement forms: *browsing* - where a new node is created, and *moving* - where we simply move to a different node in the tree. This can be equated to backtracking in the conventional sense. Each action resulting in a new page-load saves the appropriate information in Session Store so that the subsequent page-load event handler may process the new page correctly and create the necessary links in the database. If the movement is a backtrack, then the nodeID of the target node to move to is stored in Session Store. Otherwise the nodeID of the parent node is stored so that it may be used to create the appropriate connection to any newly created node.

3.2 Interests and Recommendations

Web Nav saves page interests and uses them to provide recommendations. Each ‘node’ saved in a context can have associated *NodeInterests* that are compiled through an indexing procedure. As in FollowMyLink (Briffa, 2009), we collect the relevant text from the corresponding page, and after stemming and a modified TF.IDF calculation we create a set of weighted keywords representing the most relevant terms in the page. Each set of NodeInterests is also used to update a set of *ModelInterests* for the respective model. We save NodeInterests and ModelInterests for the entire model because we use two recommendation algorithms. ModelInterests are used for the algorithm that creates a query based on terms from the entire context (tree) (Briffa, 2009). NodeInterests are used in the algorithm that generates a query based on the ***current branch***, effectively creating an on-the-fly merge of the nodes in the current path. We use this algorithm to provide recommendations that are localized to the path, rather than generalized to the entire tree.

3.3 Adaptations

To support visual adaptations resulting from the approach described in sections 3.1 and 3.2, we implemented both a tab-based interface and a global interface. For tab-based history and recommendations we implemented the Web Nav Popup view (figure 2).

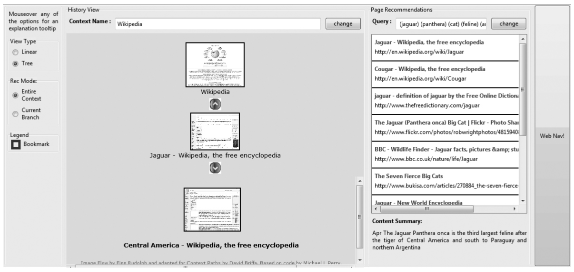

Figure 2: Web Nav Popup View showing options at the far left, history information on the left and page recommendations on the right.

The view is split into two sections. The left side shows the history information for the tab graphically as either a tree or a linearly ordered view. In the tree view, the parent and children of the current page are shown. The user can navigate through the tree to see the entire history for the tab’s context. In the linear view a user may choose to show all the nodes in a context sorted by some criteria, such as recency or frequency.

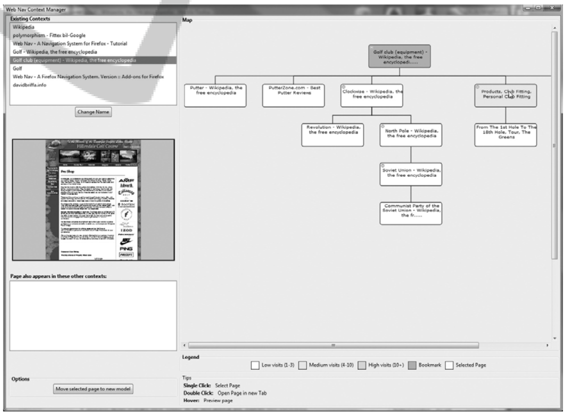

Figure 3: Web Nav Manager showing a list of contexts at the top left, with a preview below. A map of the selected context is shown to the right, where nodes are coloured based on frequency of visits.

A user can return to a previously visited context using the Web Nav Manager (figure 3). The user can see all the persisted contexts, as well as a map view of all the paths saved. Users can jump back into a session, with all context information saved.

4 EVALUATION

We conducted a preliminary evaluation for the general usefulness of Web Nav and the appeal of the paradigm involved in two stages. 14 volunteers used the system in their own home for approximately 7 days, and submitted a qualitative questionnaire about their experience with Web Nav.

The first stage of evaluation used empirical data from action logs and database entities to determine which users exhibited low browsing and low backtracking behaviour. One user's log file was corrupt and another four users were removed as they did not use the browser enough to yield any meaningful data. The second stage concerned a deeper analysis into the empirical data. It showed that the 'Up' and 'Down' buttons were used less than the regular 'Back' and 'Forward' buttons. Qualitative data suggests that the reason might be habitual. Surprisingly, although the usage data was poor, qualitative preference showed that 38.5% of all 13 volunteers (excluding the one with a corrupt log file) preferred Up/Down. The Web Nav popup view was used regularly throughout the evaluation period, and the Web Nav bookmarking feature was used occasionally. However, only a few users actually used recommendations, and even less re-visited recommended pages, possibly due to a construction bug found after evaluation had commenced, especially since 84.6% of users indicated interest in having recommendations provided and 69.2% of users said the recommendations generated were 'somewhat relevant'. Qualitative data suggested that both recommendation methods were equally preferred. The Web Nav Manager, while not used as often as the Web Nav popup view, still showed promise, especially since in many cases the opening of the Web Nav Manager resulted in the backtracking to a node in a previous context session.

The overall experience of users seems to have been positive, with 100% of users agreeing that the concept of paths shown as trees is useful/interesting. Moreover, 69.3% of users expressed interest in possibly using Web Nav in the future.

5 CONCLUSIONS

We have shown an approach to a tree-based navigation system that has yielded fairly promising results in light of the overwhelming shift required to change to a new paradigm. Its main contribution lies in its underlying framework for the persistence of tree-based contexts.

REFERENCES

Briffa, D., 2009. FollowMyLink: An Implementation of User Directed Browsing. (Unpublished report), Dept. of Intelligent Computer Systems, University of Malta.

Briffa, D., 2010, Web Nav: A Context Based Navigation Assistant for the Web. (Unpublished Report), Dept. of Intelligent Computer Systems, University of Malta.

Brusilovsky , P., 2001. Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 11, 87-110, Kluwer Academic Publishers.

Capra, R. G., and Perez-Quinones, M. A., 2003. Re-Finding Found Things: An Exploratory Study of How Users Re-Find Information. Technical Report, Virginia Tech.

Cockburn, and Jones., 2000. Which way now? Analysing and easing inadequacies in WWW navigation. *International Journal of Human-Computer Studies,* 45, 105-129.

Cockburn, A., and Mckenzie, B., 2000. What Do Web Users Do? An Empirical Analysis of Web Use. *International Journal of Human-Computer Studies*, 903-922.

Cockburn, A., Greenberg, S., McKenzie, B., Jasonsmith, M., and Kaasten, S., 1999. WebView: A Graphical Aid for Revisiting Web Pages. *Proceedings of OZCHI'99 Australian Conference on Human Computer Interaction.*

Fisher, B., Agelidis, M., Dill, J., Tan, P., Collaud, G., and Jones, C., 1997. CZWeb: Fish-eye views for visualizing the World Wide Web. M. J. Smith, G. Salvendy & R. J. Koubek *Design of Computing Systems: Social and Ergonomic Considerations*, 2, 719--722.

Herder, E., 2006. Forward, Back and Home Again - Analyzing User Behavior on the Web. (Doctoral dissertation), University of Twente. Amsterdam: F&N Boekservice.

Herder, E., 2004. Sniffing Around For Providing Navigation Assistance. *Proc. of Workshop on Adaptivity and User Modeling in Interactive Systems.* Berlin.

Kraft, R., Maghoul, F., and Chang, C. C., 2005. Y!Q: contextual search at the point of inspiration. *Proceedings of the 14th ACM international conference on Information and knowledge management (CIKM '05)*. ACM, New York, NY, USA, 816-823.

Lieberman, H., Fry, C., and Weitzman, L., 2001. Exploring the Web with Reconnaissance Agents. *Communications of the ACM*, 44(8), 69-75.

Pirolli, P., and Card, K. S., 1999. Information Foraging. *Psychological Review,* 106, 643-675.

Tauscher, L., and Greenberg, S., 1996. Design Guidelines for Effective WWW History Mechanisms. *Workshop on Designing for the Web: Empirical Studies.* Microsoft Corporation, Redmond, WA.

Teevan, J., 2004. *How people re-find Information when the Web changes.* Massachusetts Institute of Technology Computer Science and Artificial Intelligence Laboratory.