Automatic Generation of Research Trails in Web History

Elin Rønby Pedersen
Google, Inc.
Mountain View, CA 94035
USA
elinp@google.com

Karl Gyllstrom
Dept. of Computer Science,
University of N. Carolina,
Chapel Hill, NC 27599, USA
karl@cs.unc.edu

Shengyin Gu
Dept. of Computer Science,
University of California,
Davis, CA 95616, USA
gus@cs.ucdavis.edu

Peter Jin Hong
Google, Inc.
Mountain View, CA 94045
USA
peterjinhong@google.com

ABSTRACT
We propose the concept of research trails to help web users create and reestablish context across fragmented research processes without requiring them to explicitly structure and organize the material. A research trail is an ordered sequence of web pages that were accessed as part of a larger investigation; they are automatically constructed by filtering and organizing users' activity history, using a combination of semantic and activity based criteria for grouping similar visited web pages. The design was informed by an ethnographic study of ordinary people doing research on the web, emphasizing a need to support research processes that are fragmented and where the research question is still in formation. This paper motivates and describes our algorithms for generating research trails.

Research trails can be applied in several situations: as the underlying mechanism for a research task browser, or as feed to an ambient display of history information while searching. A prototype was built to assess the utility of the first option, a research trail browser.

Author Keywords
Web history, automatic clustering, semantic clustering, activity based computing, task browser, ethnography.

ACM Classification Keywords
H.3.3. Information Search and Retrieval, clustering

General Terms
Algorithms, Design, Experimentation, Human Factors.

INTRODUCTION
We recently conducted an ethnographic study, which indicated that ordinary people are engaged in extensive research and investigations on the web, but conduct this activity in ways that are distinct from traditional models of scholarly and investigative research [12]. Thus, the study confirmed findings widely reported in other studies of information work such as the importance of context in the user experience and the difficulty establishing and maintaining it: people typically ask, "Where is all the stuff I just worked on?" or "Where was I?" [7]. But the study also highlighted important differences in both goals and quality criteria from traditional scholarly and investigative research leading to the concept of *early research* as an important but underserved area. According to the study, early research is characterized by the following four properties.

For personal consumption: It is done for own consumption, done to get an answer or understand an issue, and finished as soon as the answer is found or the researcher abandons the task for more important or more enjoyable pursuits. Material is collected but is minimally processed or organized.

Fragmented process: Substantial work effort may go into a task but it is done in small installments, possibly spread over long time with many other activities interspersed. This leads to some time wasted in finding where to pick up from the previous round.

Topic sliding: Substantial user effort is invested in a single thematic exploration, though the theme may change slightly during the research process as the researcher learns more about the domain. It is often difficult to determine a specific task for a given user activity, even when we were able to ask the user right there in the moment.

Premature structure: Early researchers are typically working with vague or very open questions, and they only gradually build sufficient understanding of the domain; while they might be tempted to apply their normal organizing techniques (putting into folder, devising labeling schemes, etc.) they quickly realize their effort is wasted and sometimes even counter-productive as their organizational scheme may reflect an outdated understanding.

Based on these findings, we suggest that people would greatly benefit from tools that would allow them to browse through previous research sessions to effectively and efficiently provide a context for their work and enabling them to easily pick up where they last left off. Research trails are one such tool; they group together events that the user perceive as belonging to the same research task and represent them as temporally ordered lists of segments.

RELATED WORK
Revisitation of previously viewed web page is common, with one study reporting that it accounts for 81% [3] of web visits. Unfortunately, support for re-finding web pages is poor. Bookmarks are simple tools for keeping references to pages, but require that users immediately recognize the value of a page, and are rarely used [3]. Most web browsers retain the

IUI'10, February 7–10, 2010, Hong Kong, China.

users' browsing history, and in addition to client-based history tools the user can also use server-based tools (like Google Web History). These enable users to search for entries in their web history using text queries, akin to web searches. Though useful, existing browsing history tools are limited by their simplicity: with no intuitive abstractions built upon them, managing them is cumbersome, and finding information within them grows more difficult with size. Users often elect to re-find information by issuing new web queries rather than search their history [7].

There have been attempts to improve the usefulness of web history through better visualization. This often takes the form of page thumbnails displayed with some meaningful structure, including path-based [6], hub-and-spoke [4], and 3D [15]. LeeTiernan et al. showed clustering of pages by URL similarity and temporal proximity to be effective visualization tools [9]. Won et al. studied users' problems with using web history as a tool to re-find pages, and used the findings to inform the design of a contextual history search tool [14]. This tool provides more flexibility with filtering by date ranges, and gives contextual cues such as thumbnails. Eyebrowse records and displays users' web page visits, computes aggregated statistics and visualizes the information for users [11]. Ideally we should look for a combination of making the history easily browsable and also reducing the amounts of data by filtering only stuff that the users have invested some minimal amount of effort in.

Personalized and task-based search have been applied with some success. Umea [8] and TaskTracer [5] improve the process of keeping files organized according to task, but require users to predefine tasks, which is not easy or obvious in early-stage research. Some personalized search assists users in re-finding previously viewed information, first by building semantic profiles from terms appearing in pages from their web history or PC, then applying these profiles to add or rank results (e.g., [2, 10, 13]). Research trails are similar to personalized search in that we use data from users' web history to help re-find information, although our focus is separating history according to users' tasks, as a usable abstraction. This can also be used to re-establish working contexts, rather than simply find single web pages.

DESIGN OF RESEARCH TRAILS

We propose the concept of research trails to help the web researchers create and reestablish context across fragmented work processes without requiring them to explicitly structure and organize the material.

Supporting context: The researcher is helped by showing where he or she is in the ongoing activity. While it might seem obvious to think of overview techniques, like bird's eye view, this may be of limited practical value due to sheer complexity. Looking at the user data, we can isolate two situations where contextual assistance would be desired: "what did I leave unfinished?" and "where did I leave off last time I worked on this?" So we choose to anchor the representation in the "now", showing the researcher how he or she got there. Consequently, the research trails are one-dimensional strings of visited pages, starting from the most recent and going back in time.

Combining semantic and activity based signals: We extract both activity-based and semantic information from available sources of user activity, each type of information potentially being noisy and error-prone, for instance, using semantic analysis to determine similarity between visited pages (limited to those that lend themselves to semantic analysis). Careful combination of the two approaches allows us to mitigate flaws, e.g., using timely proximity (segment membership) to compensate for lack of semantic specificity of pages: they are tentatively assumed to be related to all topics in the same segment.

Designing with room for ambiguity: The ability to handle ambiguity – though an essential aspect of human capability – often gets left out when computing tools are designed. Research trails accommodate ambiguity in at least two areas. First, while each research trail is about strongly related work, we allow topic sliding since we only require local relatedness; thus, the first and the last segment of the trail can potentially be quite different, reflecting the development of insight the researcher went through. Second, relatedness is perceived at many different levels, and not only the theme or topic of the work can be important but also the timely proximity of events, e.g. a researcher would talk about "the work I was doing when I got the email from my sister about sitting in the Paris café"). While our semantic algorithms would allow us to split timely clustered page visits that cover different tasks or themes, we chose to support a so-called "virtual split" (explained later) thereby providing some level of contextual richness by timely affinity.

ALGORITHMS FOR TRAIL CONSTRUCTION

We apply two different perspectives on a user's web history. An *activity-based* perspective focuses on how a user interacts with data, e.g., how long a page is viewed; and a *semantic* perspective focuses on the material the user worked on, e.g., how data are related to each other by text contents. We also define three main entities: *event*, *segment* and *topic*. An event is a page visit from a user's activity history. A segment is a temporal clustering of events. A topic is a semantic descriptor obtained from a suitable statistical/linguistic technique.

Activity Analysis of Events

Events are temporally clustered into distinct periods of activity, denoted as segments. When more than M (e.g., M=5) minutes transpire between two consecutive events, a segment boundary is produced. Each segment includes the events within its two boundaries. Besides providing a first rough segmentation of work periods, the activity-based segmentation is also used to cluster events with little text content such as events dominated by images. These events are often related to their temporal neighbors and may "borrow" trail membership from them.

Semantic Analysis of Events

Semantic analysis of events provides each event with *topic vector*, which is a list of its coverage of an automatically determined set of topics. Some events have substantial contents that can be subject to semantic analysis for detection of relationship between events. Exceptions are pages with a lot of visuals and very little textual material to feed to the analysis, and pages that cannot be retrieved for analysis. We use the value *unknown* as algorithmically different from a zero value, and try to engage secondary methods for determining topical relations.

We generate a *segment topic vector* simply by averaging the topic vectors of each event in the segment.

Coherence within Segments

We are interested in segment coherence, i.e., the extent to which events within a segment have topical similarity. Topical similarity between two events is computed as the cosine similarity between their topic vectors. As we know that many segments reflect multi-tasking, we are interested in also capturing bi-focal work. Thus, we calculate *segment coherence* by combining two qualities called *average coherence* (*AC*) and *maximum coherence* (*MC*). Average coherence is calculated by averaging the similarity for all pairs of events within a segment. To compute maximum coherence we determine the maximum similarity that each event *e* in segment *S* shares with any other event in *S*, and computing the average of each of these maximums. High *AC* and *MC* indicate a mono-focal segment; high *MC* but low *AC* suggest multi-focal segment; while low *AC* and *MC* indicate a diffuse segment.

Virtual Segment Split

A multi-focal segment can be potentially split into *virtual sub-segments* to achieve better coherence within virtual sub-segments. However, we do not physically split the segments, we merely identify the sub-segments, assign topic vectors for them, and compute similarities using sub-segments when building trails.

We use a brute-force algorithm for virtual splitting, beginning by randomly divide the segment into two equal sized sub-segments. For each event in the segment, we attempt to move it over to the other sub-segment. The attempt is successful, if after the operation, it improves the average coherence for both sub-segments. The algorithm terminates when all events in the segment are processed. The process can be iterated.

Provisions to Handle Topical Slide

We tailored the trail creation method to allow for topic sliding, requiring only strong local semantic similarity among consecutive segments. This would allow a research trail to have little or no semantic similarity between the first and the last segment, provided similarity remains strong within subsequences of the trail.

More specifically, trails are created in the following way. Each segment *S,* which is not already in a previous trail, can start a trail. We add *S* to its trail. Then for all segments after *S* up to a time limit, we check if the next segment *N* should become part of the trail. *N* will be added to the trail, if *N* is not previously consumed and is similar to any of the last *W* segments in the trail, subject to a similarity threshold.

In the case of virtual split of multi-focal segments, we compute similarity against each sub-segment. If one of these similarities is above a certain threshold, the entire segment will be included in the trail, but only the sub-segment will be considered in the subsequent trail building. In this way, a multi-focal segment can belong to several trails. This leads to an operational definition of research:

- Research happens when we have trails of at least length *L* (e.g., *L*=3) with an overall net duration of at least *T* (e.g., *T*=60 min); net duration is the sum of segment durations.
- Research with significant topical sliding is characterized as *early research*. And conversely, *mature research* is research with little topical sliding

IMPLEMENTATION OF A RESEARCH TRAIL BROWSER

We have implemented and done preliminary evaluation of a research trail browser that the users can invoke from the new tab page. The implementation consists of two modules: an initialization module that builds the semantic model based on the user's Google web history and a trail browser module, consisting of a *user interface*, and a *model server* that handles the background processing for the user interface.

Initialization Module

The initialization module captures users' activity history, detects linguistic topics, and translates temporal segmentation and topic clusters into research trails. We used Google history data in this prototyping effort providing event types like query, query-click, and page visit, and derived user activity data from time stamps attached to them. History data is used to recreate the corresponding web content (caveat: pages might have changed since they were originally visited), and essential content is extracted for subsequent processing by a topic detection algorithm.

We apply the Latent Dirichlet Allocation [1] topic-detecting algorithm to the extracted web page content. This produces a list of *μ-topics* (a μ-topic is a list of prominent words ordered according to their importance to the topic), and for each page, a vector of real values between zero and one, reflecting the relative strength of semantic relationship between the page and the μ-topics.

Trail Browser Module

User Interface: The interface is fitted to the New Tab page that exists in many browsers; in addition to the usual services, such as most recently visited or most visited pages, the user sees a list of the most recent research trails; other research trails can be shown as well on request. The user can view the trails, their segments, as well as all the events (visited paged).

Model Server: When a user makes a request to see the trails, the server gets the request and queries the database. To be efficient, the trails are computed only once and stored in the server side database. The server sends the trails in XML format for the user interface module to process and display.

Preliminary Assessment

During the development of the prototype, the algorithms were continually assessed and checked towards web histories, which had been volunteered by colleagues. We conducted a preliminary assessment of the research trail method in an end-to-end experiment involving on three users. In this experiment, we presented the segments and the research trails. For segments, we also attached associated attributes such as its time stamps, duration, topic summary, coherence values, etc. Users could explore trails and segments, and inspect the corresponding web pages.

We found that the segment definition naturally captures the concept of a work session, as perceived by the user. Majority of the segments reflected a single task session, where both average coherence and maximum coherence were high. In cases of multi-tasking, we observed that maximum coherence was high and average coherence was low which matched our expectation. This version of experiment did not include the virtual segment split.

We also found that segments in trails are mostly related and coherent locally, that is, within the "trail windows". Topic sliding was observed in some cases and seems well supported. Sometimes we observed unrelated segments in trails (false positive), and some related segments were not included (false negative). The latter would sometimes be grouped with another trail of very similar research tasks. One direction of improvement is to merge similar trails and adjust similarity threshold in the trail construction algorithm so we get fewer trails but they would map better to users' perceived research tasks.

SUMMARY AND FUTURE WORK

Our ethnographic study described *early research* as a common activity that is not well supported by current tools. The study informed our design of *research trailing*, a method that automatically filters and reorganizes users' activity history (browsing as well as general interaction history) into trails of related work. The trailing method is robust against gradual shifts in research direction.

An extensive evaluation of the prototype and the algorithms is in progress, and the results will guide further design. Ideally the research trails would be visualized with screen snapshots to facilitate fast browsing. Manipulation of trails by criteria, including time, duration, recency etc., is also desirable. While such features are essential to a fulfilling user experience, they are not crucial for demonstrating the research trail concept.

Other potential future designs include: graceful degradation when less user activity detail is captured; capturing richer activity data, e.g., user activity on visited page, by moving the trail building from server to client; "mixed initiative" approaches that would let user further process the research trails, clean them up, name them and categorize them; and finally, implementation of incremental update of the topic model.

REFERENCES

1. Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent Dirichlet allocation. In Journal of Machine Learning Research 3, 2003
2. Chirita, P., Firan, C., and Nejdl, W. Personalized query expansion for the web. In Proc. SIGIR '07, ACM, 2007.
3. Cockburn, A., and Mckenzie, B. What do web users do? An empirical analysis of web use. In International Journal of Human-Computer Studies, 54, 2000.
4. Cockburn, A., Greenberg, S., Mckenzie, B., Jasonsmith, M., and Kaasten, S. Webview: A graphical aid for revisiting web pages. In Proc. OZCHI '99, 1999.
5. Dragunov, A.N., Dietterich, T.G., Johnsrude, K., McLaughlin, M., Li, L., and Herlocker, J. TaskTracer: A Desktop Environment to Support Multi-tasking Knowledge Workers. In Proc. IUI'05. ACM 2005
6. Hightower, R. et al. Graphical Multiscale Web Histories: A Study of PadPrints, Proc. Hypertext, 1998.
7. Jones, W., Dumais, S., and Bruce, H. Once found, what then? A study of "keeping" behaviors in the personal use of web information. In ASIST, 39(1), 2002.
8. Kaptelinin, V. UMEA: translating interaction histories into project contexts. In Proc. CHI '03. ACM, 2003.
9. LeeTiernan, S., Farnham, S., and Cheng, L. Two methods for auto-organizing personal web history. In Proc. CHI '03, 2003. ACM.
10. Luxenburger, J., Elbassuoni, S., and Weikum, G. Matching task profiles and user needs in personalized web search. In Proc. CIKM 2008, ACM, 2008.
11. Moore, B., Van Kleek, M., and Karger, D. Eyebrowse. http://eyebrowse.csail.mit.edu/
12. Pedersen, E.R. Habits of Ordinary Web Researchers. Under review (expected 2010).
13. Qiu, F. and Cho, J. Automatic identification of user interest for personalized search. In Proc. WWW '06, ACM, 2006.
14. Won, S.S., Jin, J., and Hong, J.J. Contextual web history: using visual and contextual cues to improve web browser history. In Proc. CHI '09. ACM, 2009
15. Yamaguchi, T., Hattori, H., Ito, T, and Shintani, T. On a web browsing support system with 3d visualization. In Proc. WWW Alt. '04, ACM, 2004