Watching Through the Web: Building Personal Activity and Context-Aware Interfaces using Web Activity Streams

Max Van Kleek
MIT CSAIL
32 Vassar Street
Cambridge, MA 02139
USA
+1-617-669-3864
emax@csail.mit.edu

David R. Karger
MIT CSAIL
32 Vassar Street
Cambridge, MA 02139
USA
+1 (617) 258-6167
karger@csail.mit.edu

mc schraefel
School of Electronics and Computer Science
University of Southampton
SO17 1BJ, United Kingdom
mc+sigir@ecs.soton.ac.uk

ABSTRACT

This paper proposes the use of the increasing numbers of Web-based user activity and personal information sources to enable the creation of more personal, adaptive, and activity-sensitive information tools. We describe our initial steps at investigating this idea, including challenges surrounding integrating information from heterogeneous web data sources. This paper contributes an implementation of an in-browser framework called PRUNE that derives an internal world model consisting of an entity database and event chronology based on heterogeneous RSS/ATOM feeds, Web APIs and other web-based data sources. Finally, we apply this model in an application called Notes that Float, that automatically learns associations between notes and a user's other activities to enable context-aware implicit reminding.

Keywords

User modeling, life-logging, personalization, and personal information management

1. Introduction

The wealth of instantaneous information brought to us by the Web, e-mail, mobile phones, social networking web sites and ubiquitous network access has begun to dramatically change how we manage our everyday work and leisure activities. In particular, the sheer volume of information has exceeded our ability to consume it, while at the same time our new responsibilities demand that we stay on top of it -- to keep abreast of the status of our family, friends, colleagues, field, economic conditions, financial market, and so on. These heightened demands on our ability to process, find, and filter information prescribe the need for better personal information tools that expand our ability to pay attention to, and act upon, the vast quantity of information arriving for us and that we have collected in our personal information repositories.

Our goal, in our research, has been to apply personal information to the management of personal information itself; specifically, to design personal information management tools that when supplied with information pertaining to its user's ongoing activities, tasks, situations and preferences, can proactively take appropriate action on the user's behalf.

In the remaining sections of this paper, we describe a framework for longitudinal activity monitoring using the web, and a simple prototype personal information management tool that uses a model derived from activity logs to enable context and activity-sensitive reminding.

SIGIR'09, July 19-23, 2009, Boston, USA.

2. User activity monitoring using the Web

This excitement surrounding social sharing on “Web 2.0” has stimulated the growth of an immense number and variety of "life-tracking" web sites that are making the chronicling everyday life activities into a popular pastime. Several of these sites have created applications to enable the automatic capture and publishing of activity data sensed via the user's own personal devices, such as their laptop, desktop, or mobile phone. Examples include Google Latitude[1], which senses the user's location using Wi-Fi, GPS and cell phone towers, Rescue Time[2], Slife[3] and Wakoopa[4], which track a users' application usage, and the audioscrobbler[5] from Last.fm, which tracks a user's music listening activity. Other sites such as fitbit[6] and Nike, sell hardware devices that capture and publish user activity to their respective sites, letting users visualize and track various metrics. The result of the introduction of these sites and their accompanying data capture tools is that hundreds of thousands of individuals have started broadcasting minute-by-minute updates of their daily life activities to the web. While the primary intended use of these data is for letting people compare their lives with others, most of these services offer the data back to users via Web APIs and syndication feeds (RSS/ATOM), turning these services into potential sources of data for adaptive and context aware-enabled applications.

Compared to directly sensing user activity, there are a number of drawbacks to using third party life-tracking sites. First, the fidelity and accuracy of user activity data acquired from the web is often lower, and is made available with substantially higher latency than if directly captured. In fact, we have witnessed a number of the sources seemingly deliberately degrading the quality of the data returned by their APIs such as by omitting certain properties or throttling query/update rates. Last.fm, for example, omits the "end time" of a played song, thus making it impossible to know the duration that the individual listened to a particular track. Furthermore, the very fact that such volumes of high-fidelity

[1] See http://latitude.google.com
[2] See http://www.rescuetime.com
[3] See http://www.slifelabs.com
[4] See http://www.wakoopa.com
[5] See http://www.audioscrobbler.net, associated with http://last.fm
[6] See http://www.fitbit.com

personal activity information are being automatically transmitted to random web services (where they are aggregated and kept indefinitely) should signal potential privacy concerns.

However, despite these disadvantages, we believe that the Web is a convenient source of a tremendous quantity of rich data about users that would have otherwise been able to obtain. For example, data aggregated from mobile phones, such as the user's call history, and a user's text messages sent and received, can easily be obtained via SkyDeck.com. Similarly, an individual's spending history, broken down by time of day and merchant, is available via Mint.com. Soon, each individual's health and medical history will be readily available via services such as Google Health. Furthermore, as these services were designed to facilitate sharing of this information with others, incorporating and obtaining information about friends' activities becomes straightforward. As the number and variety of applications that use data provided by these sites increases, we believe that these sites will be pressured to improve the quality of the data they make available via their APIs.

2.1 Modeling from heterogeneous data

While the web makes accessing the data itself convenient, building personalized applications using this data, particularly from multiple sites or sources requires addressing several challenges. First, despite standardized serialization formats (such as RSS/ATOM feeds, REST/JSON APIs), web sites typically publish data using schemas of dissimilar structure. For example, audioscrobbler RSS feeds have song and artist fields merged into a single field called "Title", while most other music-related APIs separate these out. Thus, in order for data from heterogeneous sources on the web to be effectively compared and combined, these differences and inconsistencies need to be dealt with.

Since it is undesirable to have to deal with the complexities of individual sources at the application level, we built a lightweight integration framework (called PRUNE[7]) to specifically handle this integration process. Based on data retrieved from external sources, PRUNE derives a simple world model that applications can query and explore directly. Having an intermediate model collapses the problem of schema alignment from an $O(n^2)$ pairwise alignment problem to an $O(n)$ alignment -- between external schemas and PRUNE's world model.

PRUNE's world model consists of two databases containing entities and events, respectively. Entities represent people, places, documents, events, and other "things" represented by various web data sources. Information about person entities are currently obtained from open social networking sites or web-based PIM tools such as Gmail contacts. Similarly, information about events can be acquired directly from a localizer, a gazetteer service, or event descriptions (which contain location descriptors). Relations between entities are represented by named properties on these entities. Events, on the other hand consist of time-based observations of the dynamic states or activities of those entities. Events are 5-tuples (start and time, event type, entity and state/value) representing the duration that the particular entity engaged in or assumed the particular value. Events are kept in an ordered chronology, which allows applications to easily examine sequences of events for building temporal models and analyzing correlations between states and activities.

New data sources can be added to enhance PRUNE's model. If the new data source uses the same schema as another site or source PRUNE uses already, it will be able to use data from the site directly. If not, the user may have to build an import filter, a short piece of Javascript that maps incoming fields to create/update operations on entities or events in the model. A tutorial on building such import filters makes it easy for novice programmers to construct such filters, and filters can be easily published and made available for use by other users.

With respect to predictive modeling, PRUNE's current modeling mechanisms are rudimentary, consisting of learning probabilities over event type states, and entity identity resolution. For the former, PRUNE supports online or batch learning of either full discrete probability distributions of events, or simple pair-wise co-occurrences (which can be used for Naive-Bayes style inference). These probabilities are either learned from event counts or event time durations corresponding to how long a user or entity assumes a given state. With respect to entity reference resolution, PRUNE assumes that every entity (such as a person, place or resource) at least one inverse functional property (which can be used as a unique key for merging data about entities from heterogeneous sources), and at least one familiar name. Familiar names may not necessarily be unique, and thus can only be used to retrieve entities, not modify them. This facility is used to identify mentions of people, places and things in interactions with users.

3. Notes that Float: Anticipating information needs using heterogeneous activity models

While the recent rise in popularity of personal, lightweight note-taking and scrap-booking tools have improved many individual's note capture frequency and volume, the abundance of the resulting notes can make effectively using and accessing particular notes difficult: in order for a particular note to be useful, the user must remember they took it (to make the effort of looking for it), or s/he must serendipitously rediscover it in her collection. As one's note collection grows, the likelihood of forgetting increases, while the likelihood of serendipitous discovery diminishes due to decreased visibility.

To address this problem, we have designed a system called "Notes that Float" (NTF) that proactively anticipates when a note might be needed based on its contents and previous access patterns. When NTF detects that a note might be useful in a particular new situation, it actively raises its visual salience by popping the note to the top of the user's list of notes. NTF was built on top of List-it [8], our simple personal note-taking tool for Firefox, and relies on PRUNE for observations of user activity.

3.1 Note content features (Dates and times)

Although we are currently expanding NTF to analyze other content features (particularly entity references and note types), we started with extracting date and time expressions for two reasons. First, they appeared prominently in a significant number of notes of our pre-study [1]. Second, these times often indicated when a particular event occurred or task to be done was due, and thus served as a useful indicator of times of future relevance. We designed NTF's date-time extractor to a wide variety of ways of referring to time, including vague and relative descriptions, and constructed NTF's expression relevance function to represent how likely it was that a particular expression referred to a particular

[7] PRUNE: PLUM Runtime Usually Not Exponential (PLUM = Personal Lifetime User Modeling, a previous, RDF-based life-tracking project, please see http://plum.csail.mit.edu)

moment in time. For example, the expression "tomorrow" yields a high likelihood of relevance for any calendar times that falls within the next (wall clock) day after the expression was written. Although this function was hand-constructed, we are working to replace it with one derived from a corpus such as TimeEx [7].

3.2 Correlating note use with activity/location

Our pre-study suggested that individual notes tended to be edited at particular times of day, and days of week, and while the user was looking at the same web pages as when the note was edited previously. NTF is designed to identify and leverage correlations (when present) between note-edits and any user activity or state, including time of day, physical location, weather, web page views, music listening activity or ongoing calendar activities, and use this towards ranking notes by relevance.

NTF's algorithm is simple: it listens to new events representing observations of changes to entities and their activities. These events might consist of observations of a change in what web page or document the user is viewing, the room in which they are sitting, or music to which they are listening. Then, whenever a note is accessed, NTF tallies a count, for each activity and situation dimension, of the particular activities, documents, locations, or other entities were being performed, viewed, or experienced at that particular time. These counts are then used directly in the ranking process, described next.

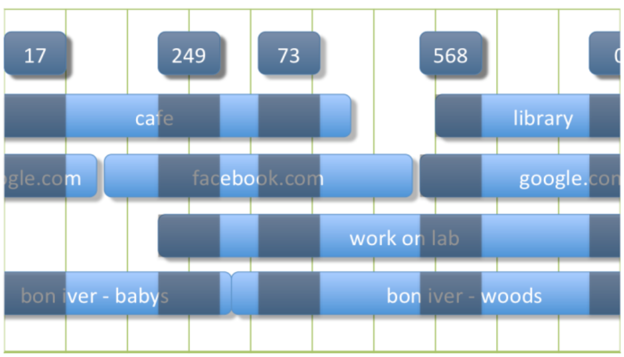

Figure 1. Learning note-activity relevance – To compute the relevance of a particular note (top row) to a particular activity or context (location, web site, scheduled calendar event or music listening activity), overlap counts are computed between the event and the other ongoing events at the same time (here, time is illustrated as flowing along the x-axis). Extremely brief overlaps are discounted.

3.3 Ranking notes

The learned associations allow NTF to simply rank notes by the posterior likelihood of the note given the user's active context and included date/time expressions. Specifically, the posterior relevance of each note is first calculated as follows:

$$\begin{aligned} &Relevance(note_i \mid note\ contents, user\ context) \\ &\quad = P\left(note_i \mid Tr(note_i, now), C_1, C_2, \ldots C_{|c|}\right) \\ &\quad \propto Tr(note_i, now) P(note_i) P\left(C_1, C_2, \ldots C_{|c|} \mid note_i\right) \\ &\quad \propto Tr(note_i, now) P(note_i) \Pi_c P(C_c \mid note_i) \end{aligned}$$

Where $P(\text{note}_i)$ is used as shorthand to represent the prior probability that Note i is accessed, $Tr(\text{note}_i, \text{now})$ is the maximum time relevance (computed by NTF's time expression evaluation function) of all time expressions extracted from the note, and each $P(C_c|\text{note}_i)$ term in the final expression represents the probability that context dimension/activity type (e.g., "web page viewed") assumed a particular value (e.g., "http://mit.edu") while a note was being accessed. This value is directly computed from the pair-wise counts previously described by taking the ratio of counts for the particular value (e.g. viewing of "http://mit.edu" while accessing a note) and the sum of the counts of all values for that activity type (e.g., viewing *any* web page while accessing the particular note). In the third line above, we made a conditional independence assumption of each context type given a particular note. While this is an obvious simplification of actual fact, this is done to let the system use pair-wise affinities instead of full conditional probability tables (CPTs) which are space-inefficient and expensive to marginalize, and forces NTF to fit a simpler model corresponding to a Naive-Bayes independence assumption.

As described in the next section, the NTF UI allows users to select which event/activity types (C_c's) are included in the calculation above, as well as whether $Tr(\text{note}_i, \text{now})$ is included. This lets the user have more control over the ranking process.

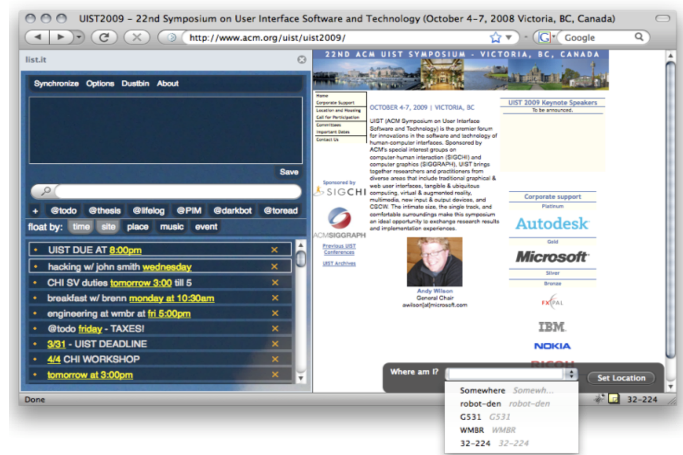

Figure 2. List-it interface – Sidebar on the left, with re-search bar, float by: bar, and notes with time information highlighted. On the bottom right shows the user's computed location.

3.4 User interface

Figure 2 shows the List-it note-taking tool embedded in the Firefox sidebar with the NTF extension installed. NTF introduces the small "float by" bar beneath the search tabs on the main UI, which is used to select floatation modes. Multiple modes may be enabled simultaneously, resulting in these terms being included as "givens" to the ranking algorithm previously described. When any of these buttons are enabled, NTF re-ranks all notes in List-it every 30 seconds (adjustable), bringing notes that exceed a relevance threshold to the top of the list. To make these notes salient and to differentiate them from the user's other notes, it "glows" floated notes with a white perimeter. When time-expression ranking is enabled, detected date/time expressions are also made to glow in yellow when the user mouses over them. The intention is to give the user feedback about the clues the system has used to rank the particular note in question. An additional configuration page (not shown) allows the user to configure PRUNE's data sources, including specifying their site-specific account usernames and passwords. Some data sources, such as our OIL localizer, require the user's system to have a WiFi card installed and, to "instruct" the system for training. Users can teach OIL about places (such as the rooms in their house) by clicking on a small widget in their status bar, and either typing a new place or selecting one they previously selected. This creates a new location state and assigns the current Wi-Fi signature to it, so that it may be recognized on subsequent visits.

3.5 INITIAL EVALUATION

Ten existing List-it users volunteered to test an early alpha release of the NTF-enabled version of List-it for 5 days, in which only 3 floating modes were available: By Time, By Place (physical location) and By Site (website). Nine users successfully installed the system (one user could not due to an unforeseen compatibility issue with 64-bit Windows). Participants used By Time mode the most (26% of the time), followed by no ranking (24%), by Site alone (14%), and by Place alone (12%). Combined modes were less popular. During the study duration, NTF re-ranked notes a total of 73 times (across all users), recommending up to 10 notes per rank. We are planning a formal study and larger deployment after implementing a few features to enhance the usability and predictability of the system, as described next.

3.6 Ongoing NTF Work

The NTF work just described demonstrates our first steps at applying PRUNE to facilitate implicit contextual retrieval for personal note collections. Implicit contextual retrieval, we believe is important in the future for helping individuals manage large quantities of personal information, some of which they may have entirely forgotten about. Our initial trial, while small, ended with encouraging results; one participant said: "[Having] tried it I decided that I liked it .. This could be the answer to an older man's increasing info and fading memory problems."

With respect to next steps, we are working to improve the NTF ranking algorithm and UI in several ways. The NTF ranking algorithm was our naive initial first shot at devising a method that was simultaneously principled and could take into account heterogeneous activity, situational and content features of notes. One initial improvement will be to automate the selection of context types/dimensions used in the ranking process; this might simultaneously improve ranking performance and permit the simplification of the UI to a single button ("ranking on/off"). To do this, NTF could learn (e.g., using feature selection approaches) the dimensions of context that are most strongly correlated with use of particular notes. A note containing the username/password for a web site, for example, is likely to be correlated only with web site viewing activity but not others. Second, to measure the effectiveness of the ranking, we plan to add facilities that let users easily give feedback about floated notes in various ways. This feedback will allow users to express nuances of "I don't want to see this now" – differentiating, whether the recommendation was a bad one (so that this feedback may be used to adjust the particular notes associations), or whether the user wants to dismiss the reminder until later for other reasons – such as in the case of deliberately putting off a to-do item. Finally, we also want to allow for greater transparency of learned associations, so that users will be able to understand why particular notes were chosen and promoted by the algorithm.

4. Conclusion

In this paper we have described our initial work towards using "Web 2.0" user activity information sources to observe user activity and information access over time, and to apply this to the construction of an implicit information reminding service. Although in its early stages of development, our simple application, NTF, supports a level of flexible, implicit context and activity-sensitive predictive reminding not available in PIM applications today. Achieving this context-adaptivity would have been substantially more difficult to implement and maintain if we had written the low-level sensing and instrumentation ourselves. In the face of the obvious complexities of dealing with heterogeneous Web APIs, feeds in different formats and the like, we have found that distilling a simple, relational world model greatly facilitates model construction and provides a useful abstraction to simplify application logic. Based on our initial experiences, we believe that this approach to using diverse information sources on the Web to characterize the user's situation and activity will foster the creation of new, more personal applications and interfaces that can effectively adapt to individuals and their dynamically changing needs[8].

Acknowledgements

This work is funded in part by the National Science Foundation, Nokia Research, WSRI, and a Royal Academy of Engineering Senior Research Fellowship. We would like to thank our PLUM/PRUNE and List-it collaborators and student researchers, including Michael Bernstein, Jamey Hicks, Greg Vargas, Katrina Panovich, Paul André, and Brennan Moore.

5. REFERENCES

[1] Bernstein, M., Van Kleek, M., Karger, D., schraefel, mc, "Information Scraps: How and Why Information Eludes are Personal Information Tools" ACM Trans. Info. Systems, 26,4 (Sept 2008), 1-46.

[2] Budzik, J., and Hammond, K. Watson: Anticipating and Contextualizing Information Needs. In *Proc. American Society for Information Science and Technology 1999.*

[3] Dumais, S., Cutrell, E., Cadiz, J., Jancke, G., Sarin, R., and Robbins, D. Stuff I've Seen: A System for Personal Information Retrieval and Re-Use. In *Proc. SIGIR 2003*, ACM Press (2003).

[4] Jones, E., Bruce, H., Klasnja, P, Jones, W. I Give Up! Five Factors that Contribute to the Abandonment of Information Management Strategies. In *Proc. American Society for Information Science and Technology 2008*

[5] "Plazer" software from Plazes http://www.plazes.com.

[6] Rhodes, B.J. Margin Notes: Building a Contextually Aware Associative Memory. In *Proc. IUI 2000*.

[7] Time Expression Recognition and Normalization. http://timex2.mitre.org

[8] Van Kleek, M., Bernstein, M., Panovich, K., Vargas, G., Karger, D., schraefel, mc. Examining Personal Information Keeping in a Lightweight Note-Taking Tool. In *Proc. CHI 2009*, ACM Press (2009).

[9] Van Kleek, M., André, P., Karger, D., schraefel, mc. Mixing the reactive with the personal: Opportunities for end-user programming in Personal information Management. *To appear in EUP-WWW, End User Programming for the Web*, ACM Press, 2009.

[8] PRUNE and NTF are released under the MIT License and available for download at http://plum.csail.mit.edu.