

Client Side Personalization

Lillian Cassel
Department of Computing Sciences

Villanova University
800 Lancaster Avenue, Villanova PA 19085-16099
cassel@acm.org

Ursula Wolz
Department of Computer Science

The College of New Jersey
Box 4700 Ewing, New Jersey, 08628
wolz@tcnj.edu

Abstract

We describe an approach to personalization that emphasizes the "client side." We posit the need for a highly individualistic user context that resides on the client machine. This context and the system that exploits it can then be used in conjunction with a broad range of search services, from highly specialized and structured digital libraries to the completely undisciplined World Wide Web. We present an initial version of the "Web Host Access Tool", (WHAT) that provides us with a testing suite for personalized search. A data collection system, the "WHAT Observer" is also described that allows us to automatically and efficiently test the veracity of iterations of the WHAT system.

Introduction

From a desire to improve user searching in the World Wide Web, we have developed a system to support user searches by learning about user preferences and by observing responses to prior search experiences. Our work is directly focused on how to develop an effective personalized profile of a user's search goals in order to aid future search. That profile is developed collaboratively with the user, minimizing the need for explicit input from the user, while delineating between preferences made explicit by the user, and those that have been inferred from previous queries. We posit the need for a localized, personal "context" that contains information contributed directly by the user and also inferred by the system. This information is applied in future searches to increase the probability that the new search will more nearly match the user requirements.

Personalization within the confines of a digital library represents a special case of our general strategy. Our approach assumes that queries will be comprised of keywords. We began our work focusing on the organic and dynamic nature of the World Wide Web, where word meaning is highly ambiguous, and thus keyword choice is dramatically unconstrained. Our intent is to help the user refine the keyword expression in order to obtain successful search results. Our approach is an attempt to augment explicit knowledge of keyword meaning with users' past experience. Our system builds a semantic interpretation of keywords both directly from user input and from inferences derived from users' actions, creating a balance between known semantics and inference. Thus our approach makes almost no assumptions about constrained word usage, but can adjust its inferences regardless of whether it is applied to the Web or a well-defined and highly constrained library.

Our work has two distinguishing features. First, personalization is done entirely on the user's own local system (the client side). This has implications for privacy, for ethical treatment of the user's requirements, and for system effectiveness and efficiency. In particular, it allows a single system to develop a user profile that can be applied to a broad range of digital libraries. Rather than putting the personalization burden on the library, the client side architecture provides a personal portal through which a user can apply localized knowledge to a number of search services. Second, we have developed a methodology for determining the effectiveness of personalization strategies. This method allows us to evaluate the "goodness" of a search result and should have broad applicability in testing personalization systems.

The major focus of our work is to aggregate search results from numerous sources, merge these results with what we know about our user's preferences and responses to prior searches, and present a recommendation about how to view the merged search results. We do not know the methods used by each of the search engines to address the query and must do filtering and reassessment in the absence of that information. As a result, our presentation of results to the user depends on characteristics of the user merged with multiple distinct assessments of the appropriateness of these responses to a given query. We keep our personalization procedures independent of the actual structure of the knowledge representation used by the information source. Thus, our approach is not tied to any particular digital library organization and can be considered a generalization of the problem of

personalization of search support in a digital library.

The Web Host Access Tool Project

The WHAT (Web Host Access Tool) project is a collaboration that brings together our interests in networks and user interfaces. A fundamental premise is that users want control of their search and the search results, but should not be burdened with the details of obtaining the results. We posit the need for a highly individualized *search context* [2,5] that includes knowledge of prior search experiences, general knowledge of search and search engines as well as explicit information provided by the user. The user should be sheltered from the details of formats, query construction rules, and search strategies of particular search services. In particular, if a user does repeated searches within a particular domain, the past experience of those searches should remain accessible and inform the new search. The burden for providing this history should be undertaken by the search system. This last requirement suggests that the search tool must reside on the user's system, where the user's preferences and search history can be recorded and consulted as needed.

A user of the WHAT system poses a query through keywords. A set of results is retrieved, based on those keywords, and the user is required to scan an ordered list of results and select those most relevant to the current interest. We assume that there is never a perfect match between the user's intent and the query expression because keywords and keyword phrases have inherent ambiguity. Examples include "Java", "four seasons", "horn" (e.g. musical instrument, animal body part), "nuts" (e.g. legume, fastener). Context cues can disambiguate meaning. For example, for the keyword "Java", adding "coffee" to the search eliminates items that deal with "programming." In acting upon a query, a system implicitly exploits context. For example, as the result of querying for "Java", those items related to coffee might be listed before those related to programming. From our perspective, personalization techniques allow the user to explicitly control context, or to allow a personalized context to develop automatically that can facilitate such disambiguation. For example, if a personal context includes information that the user is a caffeine addict and not a programmer, then a search based on "Java" will prioritize results involving coffee, possibly eliminating those to do with programming altogether.

A significant component of our work involves building automated observational tools that allow us to systematically test the veracity of components of our system. This support system, called the "WHAT Observer" is a research tool that can be completely de-coupled from WHAT at compile time in order to guarantee user privacy. We have created a clean separation between systems that are part of "WHAT" including client and server databases, from the "WHAT Observer" systems and their corresponding databases that facilitate research data collection.

The WHAT System

To date the WHAT system (version 5.0), implemented in Java, running in real time on a standard personal computer, consists of a user interface, a query constructor, a context manager (and database), and a response filter. Undergraduate research assistants have presented student posters on iterations of implementations since 1998[1][3][4]. Early work focused on individual components, exploring the range of functionality that would aid search. Through these iterations we explored a variety of user interfaces for both input and output. We implemented natural language input, graphic-based logic expression input, as well as icon-based methods of manipulating the search results. Experimentation with a variety of data structures and knowledge representation forms led us to conclude that local information should be stored in a relational database. The most recent iteration, WHAT 5.0 was implemented in order to let us focus our research effort on the context manager. Consequently we stripped down the complexity of the other components in order to reduce their impact on experimental outcomes.

The user interface consists of two windows, used for query construction and response analysis respectively. The current version of WHAT uses a simple keyword entry box into which a user types a string of keywords. The response display is a list of "hits", or search results that are ranked in order of applicability to the query by the context manager. The user can annotate and manipulate resulting hits.

In WHAT 5.0 annotation has been constrained to buttons marked "yes", "maybe" and "no." The user can specify applicability of hits to the search. The results can then be reordered. Exploiting such buttons provides an example of the degree of intrusiveness of an interface. Clearly, users are not going to annotate a large list of hits. In particular, once a likely hit is found the user will probably not scan the remainder of the list. The key is to motivate the user to see some payoff from some effort, especially if there are gains in system responsiveness over the long term. Furthermore, if the method of data entry, in this case, buttons, is at least as natural and

efficient as scrolling, then the user is likely to engage in the activity. The payoff does not come from a single interaction; instead it comes from the cumulative effect of providing information from which the system can draw inferences.

The query constructor interacts with the context manager and the user interface. Taking the current query and a user description of context, the query constructor attempts to form a good search string to send to a set of search services. From the context manager, the query constructor will learn of terms that have previously been included in similar searches that can assist the current search.

The response filter uses the context to see how previous searches impact the current search goals. The responses from those services may include duplicates and variations in format that the response filter condenses and organizes into a format preferred by the user. The user can view merely the search target (the URL for the World Wide Web), or add the service of origin (e.g. Yahoo, AltaVista), or annotated information provided by the search service. The current response filter delivers the annotated list to the user interface in the order determined by the context manager.

In collaboration with the user, the context manager accumulates and analyses prior search topics enhancing the search context. Both the query constructor and the response filter use this information. An initial context in WHAT 5.0 consists of weighed keywords obtained directly from the user. The context manager conjoins these to the keyword provided in a session by the query constructor. The resulting strings are sent to remote search services. The response filter orders the resulting list based on the weighting impact of the context keywords. After the user interacts with the response window, the context manager can analyze the impact of the weighting scheme and adjust it accordingly. It can also add the new keywords from the keywords to the context.

Both the current context knowledge structure and the context manager algorithm are very simple. Our goal thus far has been to get a working system up and running in order to begin systematic testing. We envision the context manager as well as the context itself being enhanced through natural language processing techniques such as thesaurus based analysis of keywords. We must stress that the context is to be reused over time. Thus the distinction between keywords that form part of a single search, and those that are part of a persistent context is significant. While the weighting scheme at present is ad hoc, we anticipate that machine learning techniques can provide a more principled basis for weighting keywords within the context.

The WHAT Observer

System assessment requires evaluating the "goodness" of a response as impacted by the various systems we have implemented and plan to enhance. As we developed specific evaluation questions we saw the need for the WHAT Observer [7]. This continues the work of Hartson, et al [6] on remote interface evaluation, but extends it to allow the evaluation of the impact of underlying analysis systems (such as the context manager). The WHAT Observer allows us to selectively observe the interaction between the user and the WHAT system. Data is automatically stored in a server-based database. User surveys elicited through web forms can augment the database. The WHAT Observer is intended to be exclusively a research tool.

Our initial "goodness" metric is determined by the order of query responses. We posit that in any search, the "best" responses should appear before "worse" responses in the presentation order. When a user initiates a query through WHAT, three ordered lists result that can be captured by the Observer: (1) The web search services return an ordering determined by their metric. (2) The WHAT context manager reorders the initial list and presents it to the user. (3) The user implicitly reorders the list when assigning "yes", "no", "maybe" tags to some items and ignoring others.

The tags create a tripartite grouping of the examined items. We presume in this analysis that examination is systematic and complete up to a point in the initial list. Items that are unexamined come after such a point and are consigned to the "no" group. Note that this differs from the interpretation that unmarked items are "unknowns" and "maybe" could indeed be hits. This latter interpretation aids the context manager, but the not the analysis of search success.

In a perfect response the contents of each group would exactly match the groups assigned by the user. We can analyze the migration of items from WHAT's response list to the user groupings. Little migration suggests a "good" ordering. Note that there is no significance within the grouping. An item's position in the group is derived entirely from its position in the original list. An exact definition of "little migration" awaits analysis of our initial test results. Furthermore, it is dependent upon whether the user thought the search was "successful."

Consequently we anticipate that the degree of migration as a metric will not be fixed for all users, but will be dependent upon the user and the scenario in which the search occurred.

The "goodness" metric can also be applied to the initial ordering produced by the commercial services (or by any digital library to which WHAT is applied). Presumably, the WHAT context manager will produce less migration than the original list. Furthermore, as we improve the context manager we can test the degree of migration between implementations. We will be able to see directly whether enhancements to the context manager improve the presentation or not.

The migration metric also provides insight into the efficiency of personalization. A system that expends significant resources in time and space may not be cost-effective if there are only small improvements in minimizing migration.

We are poised to begin systematic testing across the Internet, collecting observational data from remote sites. The initial application of the WHAT Observer to the WHAT system itself proved useful at a fundamental level. Two deeply imbedded bugs in the context manager that were not apparent from small testing suites emerged during trials with the WHAT Observer. We are about to conduct a controlled study of context impact on search query using two domains: "train engine horns" and "Java coffee". By seeding a range of contexts with thesaurus entries we can then analyze the migration rate. Our extended data analysis strategy is twofold: we will collect data on voluntary users who have their own agenda. We are poised to conduct a controlled study in which undergraduates at our institutions will be asked to search for the two topics listed above.

Ethical Attention to User Needs

Holding the user information at the client side has obvious implications for protecting user privacy. Because the user-specific information is obtained and stored entirely on the user's own system, the user retains control of the information. The user can determine when and if to share that data with others and will have better control of how it may be used. The information provider is not able to determine characteristics of the user and use those characteristics to push unwanted materials to the user. Instead, the user retains control of what information is pulled from the source. This is a far more reliable form of privacy control than depending upon signed agreements.

A second ethical implication is that of user trust. When users relinquish control of data they must trust that filtering at the service side is in their best interest. By more directly controlling response filtering at the client side, users have more direct control of how their personal information is used, and whether it was used as they intended. The profile can be examined and modified as the user wishes in order to obtain the most appropriate responses. One practical example is age appropriate responses balanced with first amendment rights. The search service or digital library can be freed of the burden of censorship, deferring filtering decisions to the personalized client side system under parental or school control.

One issue raised repeatedly raised by our undergraduate researcher assistants is the potential impact on individual users of the data collected by the WHAT Observer. Clearly we can learn from the aggregate results derived from the Observer's analysis and give those generalized inferences about specific contexts back to individual users. But is this ethical unless the user fully understands the degree to which the Observer is watching? While these issues are fascinating, we currently take the stand that we want to support user privacy. Our research question is focused upon the question of how far we can get without resorting to stereotypical information derived from analysis of such aggregate information. If we can provide the user with significant help without resorting to server-side data collection then why invade the user's privacy in the first place? However, if significant gains can be made by contributing collective data from multiple users to create meaningful stereotypes, then our work should help delineate between reasonable server-side acquisition of information and data collection that is exploitative and unnecessary for the individual user.

System Efficiency and Effectiveness

Previous sections have alluded to issues of efficiency and effectiveness. The WHAT system and the WHAT Observer are experimental systems intended to shed light on the degree to which personalization can take place without unduly intrusive behavior. We have just begun to systematically look at the degree to which personalization can take place.

Personalization, especially that which does not rely upon stereotypes comes at a price. In our case we assume

that personal contexts will be built over time. A new context is a sparse entity that may thwart rather than support the user's direct goals. An open question is how long it will take to "grow" useful contexts. How quickly does some sense of context contribute to more efficient search?

As contexts grow in sophistication users should have to do less to get the search results they desire. It should be possible to adapt a highly developed context to a new domain rather than building each context from scratch. In our vision of client-side personalization, the user maintains a collection of contexts that are extremely sophisticated tools to provide efficient search. Users might even voluntarily share contexts with one another. Our daunting research question is to develop protocols to show that such contexts can develop, can be adapted to new domains and that they will ultimately enhance user search.

Effectiveness of the client-side approach is more immediate. Placing the user preference description on the client side relieves the digital library of responsibility for establishing criteria for personalization that adequately captures the needed information about all possible users. Rather than defining which characteristics to capture and how much history of this user's prior use to retain, client side personalization frees the digital library to focus on an effective response to a query that is assumed to be well formed. The burden of anticipating the needs of an individual is off-loaded.

At another level, the use of client side personalization is critical to effective service for the user. With the personalization information stored on the client system, the user is able to apply that personalization information to searches at many different digital libraries. In the WHAT system, a query is sent to a number of search engines, relying on one set of information about the user needs. Searching digital libraries should also allow the user to make full use of the accumulated profile and search history information to make effective use of diverse resources available from many sources.

Finally, there is an issue of effectiveness in terms of how well the personalization represents the individual user. We maintain that a profile based on individual choices and the user's own search history can more easily reflect the user's true goals than can profiles that are at least partially derived from generalizations or stereotypes of other users. The WHAT system and the WHAT Observer provide us with the tools to address continue to explore these issues.

References

1. Anderson, Jonathan and Jason Dobies. "The Web Host Access Tools (W.H.A.T.) Project" SIGCSE 2001 Charlotte, NC
2. Baldonado, Michelle Q Wang and Winograd, Terry. "SenseMaker: An Information- Exploration Interface Supporting the Contextual Evolution of a User's Interests." Conference proceedings on Human Factors in Computing Systems, pages 11 - 18, 1997.
3. Behringer, Brice, Nikolsky, Mark and Sipper, Michael. "A GUI for Web Host Access Tools." The College of New Jersey, Ewing, NJ.
4. Bronevetsky, Greg. "The Brains of WHAT: A Data Structure for Internet Searches." SIGCSE 99, New Orleans, LA, page 378, 1999.
5. Buckley, Chris, Gerald Salton, and James Allan. The Effect of Adding Relevance Information in a Relevance Feedback Proceedings of the 17th ACM-SIGIR Conference on Research and development in information retrieval, 1994, Pages 292 — 300
6. Hartson, Rex, José C. Castillo, John Kelso and Wayne C. Neale Remote evaluation: the network as an extension of the usability laboratory; H. Conference proceedings on Human factors in computing systems, 1996, Pages 228 - 235
7. Klett, Jared. "Web Host Access Tools Observer." SIGCSE 99, New Orleans, LA, page 379.