WPI-CS-TR-03-26 July 2003

Search Smarter, Not Harder – Using Personalization to Improve Web Search Results

by

Mark Claypool
Eugene Cushman
Daniel Murphy
and George Stuart

Computer Science
Technical Report
Series

WORCESTER POLYTECHNIC INSTITUTE

Computer Science Department
100 Institute Road, Worcester, Massachusetts 01609-2280

Search Smarter, Not Harder – Using Personalization to Improve Web Search Results

Mark Claypool, Eugene Cushman, Daniel Murphy, and George Stuart

Most search engines, indispensable tools for finding information on the Web, do not take advantage of a user's personal preferences in creating result sets from search queries. In particular, collaborative filtering, an effective personalization technique that uses peer opinions to recommend items of interest, has not been widely used in Web search engines nor have the benefits of collaborative filtering to search engine technology been thoroughly evaluated. We have designed and implemented a search engine called Foible that personalizes Web searches based on user preferences and uses collaborative filtering to enhance the result sets returned from user queries. Through a carefully designed user study, we evaluate the effectiveness of Web search with personalization and collaborative filtering compared with a traditional Web search engine. We find Web search results based on personalization and collaborative filtering provides result sets more closely related to user interests than result sets returned by traditional search engines. Moreover, users overwhelmingly prefer results returned by a personalized filter with collaborative filtering to those returned by traditional search engines.

Categories and Subject Descriptors: [ ]:

General Terms:

Additional Key Words and Phrases:

1. INTRODUCTION

The search engine has become an indispensable tool in navigating the billions of Web pages residing on the more than twenty million servers [Zakon 2003] that compose the global World Wide Web. Search engines function as filtering agents, empowering users with the ability to find the needle of desired information within the overwhelming haystack of useless bits. As the Internet continues to expand at an exponential rate, search engines must continue to refine and enhance their technology in order to remain relevant.

While Web search engine technology has made advances in storage and indexing techniques, it has not benefitted from the recent advances made from personalization. A search engine using a personalized profile should effectively remember each user's likes and dislikes across multiple searches, producing a more useful set of results for some queries. Collaborative filtering, in particular, is a personalization technique of using peer opinions to predict the interest of others. Users indicate their opinions in the form of ratings on various pieces of information, and the collaborative filter correlates the ratings with those of other users to determine how to make future predictions for the rater. In addition, the collaborative filter shares the ratings with other users so they can use them in making their own predictions. A search engine using collaborative filtering could match user interests with other users, using the aggregative preferences of the group to better predict whether a particular Web document would be of interest to a member of the group, based on the opinions of others in the group.

While there have been several systems that combine collaborative filtering with Web search technology [Wasfi 1999; Balabanovic and Shoham 1997; Goecks and Shavlik 1999; Rucker and Polanco 1997; Chan 1999; Thomas and Fischer 1997], to the best of our knowledge, there has been little evaluation of how collaborative filtering can directly enhance today's search engine technologies. Thus, it is not our goal to necessarily come up with novel collaborative filtering and search engine technologies. Rather, it is our goal to evaluate how much more effective, if any, typical search engine technologies might be if they are enhanced with collaborative filtering.

With this goal in mind, we constructed a functional search engine named *Foible* that uses core technologies employed by Google[1], the most popular search engine in the United States [Sullivan 2003]. Upon processing a search request, in addition to providing a list of Google-like search results, Foible also provides a list of search results enhanced by personalization, including collaborative filtering technologies. To evaluate the effects of Foible's personalization on Web search, we populated Foible's index database by a substantial crawl through some specific test domains. We then designed and conducted a study that had users perform several search engine tasks, each with a different level of specificity, using search results returned by Foible both with and without the personalization enhancements. We analyzed the data gathered through result set analysis as well analysis of the user surveys.

We find personalized search provides, on average, result sets that are more useful to the users' queries than are result sets from non-personalized search engines. In addition, personalized search provides a more properly ordered result sets than do non-personalized searches, meaning the documents at the top of the list are more likely to be useful than documents at the bottom of the list. Perhaps most importantly, users overwhelmingly prefer a search engine with personalization to one without personalization.

The rest of this document is organized out as follows: Section 2 provides background into search engines and collaborative filtering; Section 3 describes details on the design and implementation of the Foible system; Section 4 describes the user study and performance measures we use to evaluate the benefits of a search engine with personalization; Section 5 analyzes the results from the user study; Section 6 summarizes our conclusions; and Section 7 presents possible future work.

2. BACKGROUND

This section provides background into the Google search engine and a collaborative filtering algorithm, the two fundamental technologies employed by Foible.

[1] http://www.google.com

2.1 Google

Google was first created as a research project at Stanford University [Brin and Page 1998]. Its creators, Sergey Brin and Lawrence Page, wanted to design an indexing engine that was fundamentally better than any of the search technology that existed. Additionally, they wanted the technology they were designing to be primarily academic. It was the hope of Brin and Page that this would make Google an excellent research tool for other scholars to base future work upon.

A fundamental algorithmic feature that arose in Google is the metric of *PageRank*. PageRank is a calculation, given all the citations(links) on the Internet, of the probability that a Web page will be visited by a random Web surfer [Brin and Page 1998]:

> We assume there is a "random surfer" who is given a Web page at random and keeps clicking on links, never hitting "back" but eventually gets bored and starts on another random page. The probability that the random surfer visits a page is its PageRank. And, the d damping factor is the probability at each page the "random surfer" will get bored and request another random page.

In brief, PageRank is the following:

$$PR(A) = (1-d)(\frac{PR(T_1)}{C(T_1)} + ... + \frac{PR(T_n)}{C(T_n)}) \tag{1}$$

where A is any given page having pages $T_1...T_n$ point to it, $C(A)$ is the number of links pointing *from* A to other pages, and d is damping factor referred to in the above quote. PageRank, a powerful addition to Google, was the first time that the frequency of citations had been used to generate a ranking for Web pages on the Internet.

Another fundamental feature of Google is the way in which it handles the text associated with HTML anchors. Most search engines associate the text of an anchor with the page in which it resides. Google does this as well, but Google also associates the anchor text with the page it *points to*, allowing Google to index items that ordinary indexing engines cannot (images, programs, and databases) [Brin and Page 1998].

Finally, Google has a few additional features that improve its usability. First, it considers the font and size of text to imply their importance on a Web page. Second, it maintains information on location for each page indexed thus allowing "proximity" to be used in the search calculation. Lastly, it stores the raw HTML making it available from Google as a cached reference should the page maintainer remove the it.

2.2 Collaborative Filtering using Pearson Correlation Coefficient

When making recommendations using the *Pearson correlation coefficient*, the predicted votes for an active user are calculated using partial information from the user and a set of weights from the database. This user database consists of a set of votes for the user i on the item j, with I_i being the entire set of items that the user has voted on. The equation for the average vote is:

$$\bar{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j} \tag{2}$$

The weighted sum, $p_{a,j}$, is the predicted vote of the user. The variable n is the number of users in the database with nonzero weights and k is the normalizing factor. This equation is defined as:

$$p_{a,j} = \bar{v}_a + k \sum_{i=1}^{n} w(a,i)(v_{i,j} - \bar{v}_i) \tag{3}$$

where $w(a,i)$ is the correlation between users a and i which can be expressed using the Pearson correlation coefficient:

$$w(a,i) = \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{a,j} - \bar{v}_a)^2 \sum_j (v_{i,j} - \bar{v}_i)^2}} \tag{4}$$

This equation can be refined further by assuming that if the item is liked by many people in the database, then it should be considered less valuable when determining $w(a,i)$:

$$w(a,i) = \frac{\sum_j f_j \sum_j f_j v_{a,j} v_{i,j} - (\sum_j f_j v_{a,j})(\sum_j f_j v_{i,j})}{\sqrt{UV}} \tag{5}$$
$$U = \sum_j f_j (\sum_j f_j v_{a,j}^2 - (\sum_j f_j v_{a,j})^2)$$
$$V = \sum_j f_j (\sum_j f_j v_{i,j}^2 - (\sum_j f_j v_{i,j})^2)$$

This modification is based on *inverse document frequency* which makes more commonly occurring words have less weight than less commonly occurring words. In this equation, f_j is defined as $\log \frac{n}{n_j}$ where n_j is the number of users who voted for item j and n is the number of users in the database. f_j would be zero if everyone voted for that item, so effectively, f_j is a weight [Breese et al. 1998].

3. FOIBLE

Foible consists of a working search engine, populated by data from a substantial crawl of the Internet for our test domain, along with a collaborative filtering system that enhances the results returned by the search engine. Using a relational database as a backend, Foible constructs user profiles and, using collaborative filtering, associates the ratings of the users through their profiles with the algorithms discussed in Section 2.2. Foible uses the information gained from collaborative rating of pages in the search engine ranking algorithm.

3.1 Search Engine Technology

Fundamental to our goal of practically evaluating the benefits of collaborative filtering with Internet search engine, is the design and implementation of a basic search engine that models, as closely as possible given our relatively limited resources, the functionality of typical search engines.

The majority of search engines in existence today function in much the same manner. First, a "spider" or "crawler" scours the Internet and collects as many pages as it can. Second, the search engine takes the collected data and indexes it based on some categorization algorithm. Finally, this index combined with a user defined query produces a "page rank" which attempts defines a page's relevance to a user's query.

3.1.1 *Spidering.* At the base of any search engine is a component which scours the Internet by traversing the links it finds within Web pages. The "spider" is the first stage in building a database of online data that can be indexed and queried. Typically the spider's duties are simple. It "walks" through the links that it discovers and stores whatever data it finds. This aspect of search engine technology is often called "crawling" (the fact that the component is named a "spider" is apropos). The actual act of crawling is a breadth-first tree walk of interlinked Web pages. A start node, or root, is chosen from which to begin the search. This page is parsed to discover any links to other Web pages. For the purposes of our project, we consider only those documents that link to other HTML web pages that are parseable by our own engine, and discard other types of content. Such links have the form <A HREF="http://LINK.html">Anchor text</A>.

We have created a spider that functions in the manner described above, understanding HTML links, and constructing an interlinked graph structure of Web pages. This graph is then explored, with special checks for previously seen nodes and depth limitations in place to prevent the expenditure of more resources than necessary.

Foible's spider constructs its node-network by matching the URL string to certain predefined patterns. By limiting the pattern to certain extensions (.html, .htm, .shtml, etc.) we are able to avoid crawling potentially large documents to which an HTML page may be linked. The spider is intelligent enough not to follow links in which it is difficult to analyze content, such as PDF files or multimedia content.

3.1.2 *Indexing.* After the "crawling" has completed, the search engine must categorize the data it has collected. This stage, often termed "indexing", involves finding keywords and building association tables that can be queried efficiently. Generally the index consists of the main words or phrases that appear in the pages crawled by the spider. The indexing process creates a database of information that relates these main words or phrases to the pages they can be found within. As described in 2.1, more advanced search engines, like Google, make some additional assumptions, such as PageRank and the association of anchor text with what it references. This is postulated to produce "better" query matches by introducing selected heuristics and probabilistic ranking algorithms to the indexing calculations.

During the analysis phase, Foible evaluates the retrieved documents based on a number of different factors that later become relevant during the collaborative filtering stage. Foible measures the following characteristics for later use in matching with user profiles:

—*Document Size* - The document size refers to the total number of bytes of not only the HTML but all associated materials, such as inline images. This byte estimate can be used as an indicator of the amount of time needed to download the page. This is a factor of definite interest for users with low-bandwidth connections, and which we expect to have high impact on personalized queries.

—*Number of Words*, *Flesch-Kincaid Reading Level* [Flesch 1949], *Flesch Readability Score* [Flesch 1949], and *Fog Reading Index* [Miles 1990] - The depth of detail of document can be approximated using a count of the number of words combined with an analysis of the reading level of the document. These factors, taken together, vary greatly across users in predicting interest since many of whom have differing preferences for longer or shorter documents. When personalized, we anticipate that these factors will be of great utility to younger people searching the Web. Although elementary students are an ever growing segment of Internet users, few search engines are capable of adapting themselves to meet the specific needs of this demographic.

—*Number of Images*, *Number of Links* (external and internal), *Word Frequency*, and *Markup to Content Ratio* - The visual style of a page can play a large role in influencing the user's level of interest. Although we do not provide direct means of examining layout, we attempt to classify a page as visually appealing by examining thee number of images displayed inline and the ratio of bytes of HTML tags to bytes devoted to content. When examining the number of images, it is also necessary to check the size of the images, since a page will appeal graphically heavy if dominated by large pictures, and a large file size will usually correlate with a large image size.

3.1.3 *Storage.* From the previous sections, it is apparent that a great deal of disk space is needed to store all the data collected from spidering and indexing.

In Foible, while the search engine is crawling the Internet, it indexes what it finds and stores the contents in the database. This allows it to build a comprehensive database while permitting off-line analysis of the results of a spider crawl. In addition, a pleasant side effect of this approach is that Foible can also provide users with cached copies of the pages. The price for this method is the speed of crawling in that the Foible spider crawls fewer pages than might other search engines. However, since the goal of our work is to improve the effectiveness of the search engine, the moderate slowdown in crawling speed is relatively unimportant.

3.1.4 *Querying.* Once the search engine has compiled a database of indexed data, it is able to perform queries on that data. Most basic search engines use some form of word frequency algorithm. Using the index created earlier, the Foible search phrase matches up against the indexed data in order to determine what pages are most relevant to the current query.

3.1.5 *Architecture.* To allow better visualization of the relationship between the various components of our search engine, Figure 1 depicts the interactions between the spider, the analyzer, the query engine, cache, databases, and the user.

3.2 Collaborative Improvements

Although traditional search engines are a powerful means of filtering information, a major problem with conventional search engines is their lack of state; each search

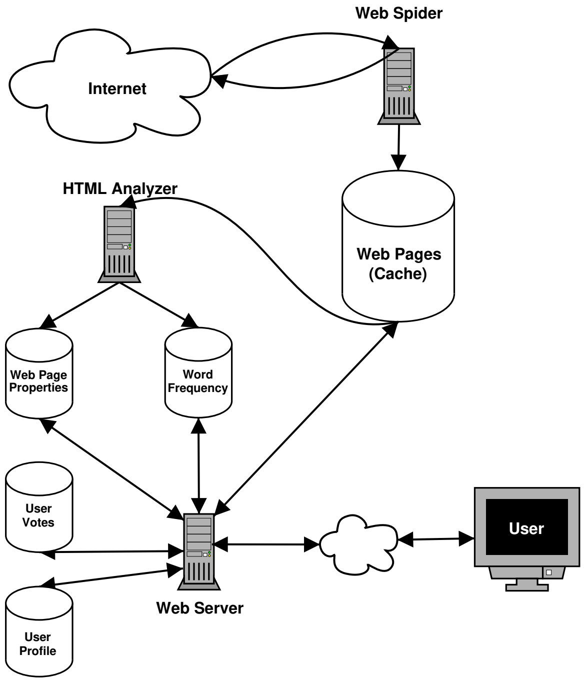

Fig. 1. Architecture Implementation Overview

is treated as an individual query, with no attempt to associate queries to users and take advantage of a user's past queries. In Foible, we have extended the functionality of the search engine to incorporate the concept of individual user profiles. By tying identity to search, we permit the collection of data that may be used to return a more accurate and personalized search. By using collaborative filtering, individuals are matched with persons with similar tastes, allowing ratings of similar users to predict whether or not they will prefer the types of certain pages more than others.

3.2.1 *Establishing Identity.* In order to harness the power of collaboration, it is first necessary to define a distinct user identity to queries performed on the system. The most common means of tracking user usage on the Web is through the use of cookies. Cookies are small bits of textual information that are transferred to the Web server by the client browser during each request. Web servers can store and retrieve these cookies to add state to the otherwise stateless act of requesting a Web document. Internet advertisers currently make heavy use of this method. The marketing world recognizes the utility of having as much demographic information associated with a user as is possible. Major banner advertisement providers will track users through a similar system, by storing a unique ID with the client browser.

To identify the user in Foible, we make use of a simple cookie consisting of a unique integer ID. All other information associated with the user is stored in the database backend, with this ID serving as a key. Each time that the front page to the search engine is requested, the server checks to see if the unique ID is passed along with the request. This indicates that the user has already visited Foible and any searches performed from this point onward is associated with the user.

If the request represents the first time that a user is visiting the search engine, or if the cookie has been removed from the user's system, a new unique ID is automatically generated and stored on the user's client. This results in the establishment of identity in a manner that is completely transparent to the user: no cumbersome logon or password tokens are needed. This approach, while the most easy to use for the user, is not without certain negative attributes. Security in this model is weak, since there is a single token that both identifies and authenticates the user. Users lose the benefits of a customized search when they change computers, and profiles can get easily confused when multiple users share a single machine. To prevent the collection of false data, the search engine provides a "clear my cookie" button that allows a user to erase their current cookie. This is useful if the cookie set on the machine with which they are browsing was used by a previous computer operator, and is no longer needed.

3.2.2 *Personalizing the Search.* Before better recommendations can be made, Foible must adapt itself to the preferences of the user through the process of personalization. The initial step in the process of personalization is for the user to conduct a search. Whenever a user gives feedback after performing a search, his or her profile is altered. Thus, there is no distinct "training phase" – the user is constantly and transparently training the system to better suite his or her needs. The initial search is carried out with the default profile of equal weights in each of the factors discussed in Section 3.1.2. The pages that result from the query are internally ranked and presented the user.

The user examines each of the links returned, and then provides explicit feedback of how useful the page was, on a scale ranging from one to five using the interface shown in Figure 2. The user's ranking is compared to the initial ranking computed by the search engine. Any differences indicate the presence of some factor influencing the user's preference. If such a presence is determined, then those pages ranked highly by the user will adjust the factor weights in the user's profile according to the factors in the document.

The algorithm Foible uses to adjust the weights is a variant of alpha-blending, in which a weighted average of the profile value and the Web page's value is computed. Specifically, our implementation associates two dynamic data structures with the profile. The first is a set of weights, values between 0 and 1, that are used to associate the user with other users. Whenever these weights are altered, they must always sum to 1, which requires balancing any additions to a single weight with an appropriate number of subtractions. During modification, each weight is adjusted by moving it toward or away from (based on if the user had a positive or negative reaction) a point on a statistical distribution curve corresponding with the percentile into which the attribute of the page in question falls in relation to all other pages. The second part of the user's profile models an "ideal page". Any time that a user indicates a preference for a Web page, his internal "ideal page" is adjusted to be more like the page he positively rates.

An example of the personalization can better illustrate the process. The new user, Alice, goes to the Foible site. The site notices that she does not have an identifying

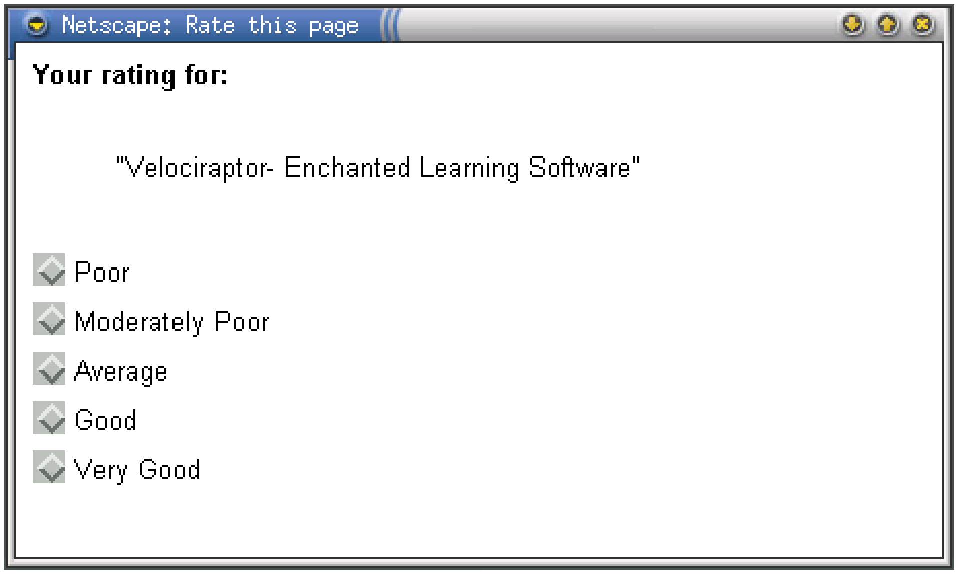

Fig. 2. User Interface for Page Ranking

cookie, and creates a new profile in the database for her. The associated unique ID is stored as a cookie with Alice's browser. Alice's initial profile is set so that her weights are equivalent (in this case equal to $0.\bar{1}1$ since there are 9 such weights that Foible tracks), and her idealized page has attributes equal to the averages of all pages in the database. Alice then performs a search, the results of which are displayed in ranked order within her browser. She then visits each one, and begins to rate the usefulness of the pages on a scale of one to five. Alice examines the presented links. The first link is a good recommendation, so she scores it a five. The second link, despite being highly recommended by the system, is scored at two by Alice. She scores the rest of the links as would be expected by the system, decreasing her scores as she moves down the list of links. By rating the first page highly, Alice has already begun the process of customization. Let us assume that the first document is a relatively simple document, with a number of images and complex layout (which we infer through our Markup-to-Content ratio). Her internal weights are reoriented so that a higher priority is placed upon images and Markup-to-Content ratio. These weight increase, while the others decrease, maintaining the requirement that they sum to 1. Alice's "ideal page" adjusts itself to be more like the page that she has just rated so highly. Since she rated it a five, her internal page attributes will move half way to these new values. Had she rated it a lower value, such as a four, her values would have moved less of the distance to those of the page (in the case of four, this would be one quarter of the distance). Since the page had many more images than the average page in the database, Alice's internal image preference is now above the average for the database. Now let us examine what happens when a user votes negatively on an item. The page which Alice dislikes is very long, verbose, and lacks the images and content that she enjoys. Alice's weights are again adjusted. This page is found to have a Flesch-Kincaid reading level in the 90th percentile and document length in the 80th percentile. Since she registered negative preference, Alice's weights will be adjusted towards the inverses of these values, 0.1 and 0.2 respectively. Her weights are adjusted, and her profile updated to represent her preferences. The system can now infer that she prefers short, easy to read documents containing many pictures. In the future, long and difficult to read documents will be ranked lower, and short documents with many pictures will be towards the top of her search results.

3.2.3 *Matching Users.* Once users have established profiles that express their individual preference, it is possible to associate them with other users to allow access to a greater pool of ratings. Even if an individual user has not viewed a particular page before, it is possible to make a prediction of whether or not this user will find the page of interest based on whether others with similar profiles to the user have found such a page of interest. For example, if Alice has never viewed the Web page "Ten-Thousand Words on Immanentizing the Eschaton", but her semi-literate friend Bob (with whom Alice's preferences correlate well) has both viewed this page *and* hated it, then we can predict that Alice too will dislike it.

3.2.3.1 *Matching Users Using Correlation Factor.* One means of associating users is to compute a correlation matrix between all users of the system. With n users, this would produce an nxn matrix, the elements of which would be a correlation factor ranging from a -1 indicating a complete inverse match, through 0 indicating no correlation, to 1 indicating a complete match. These ratings are computed using the Pearson correlation coefficient, as described previously in equation 4.

This correlation factor is computed in two parts: through vote correlation and profile correlation. The correlation between profiles is based upon a computation of the mean squared difference of the various weighting factors that compose the user profiles. The result of this computation is combined with a similar result that relates the degree of similarity between the set of pages that both users have voted on. The final result is a value between 0 and 1 that indicates the level of correlation between the users. Because of the many different factors that such a computation takes into account, we consider any users with a correlation factor greater than 0.4 to be strongly correlated. Users who are strongly correlated (we use the term "comrades" internally for such a relationship) are capable of influencing each other's search results. Pages ranked highly by one user are likely to turn up higher in the search results of users to whom the user is strongly correlated. This equation can be summarized as follows:

$$w_{profile} * \frac{\sum_j (w_{a,j} - \bar{w}_a)(w_{i,j} - \bar{w}_i)}{\sqrt{\sum_j (w_{a,j} - \bar{w}_a)^2 \sum_j (w_{i,j} - \bar{w}_i)^2}} + w_{votes} * \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{a,j} - \bar{v}_a)^2 \sum_j (v_{i,j} - \bar{v}_i)^2}} \tag{6}$$

These calculations are performed in real-time for our project since one of our design specifications was not to support more than one hundred users at a time. Unfortunately, the computational time associated with these operations does not scale linearly with the number of users. As discussed in Section 7, future work would be to explore doing these calculations nightly, during a period of low usage, and then carry out the functions of a day's worth of queries with these precomputed values.

4. EVALUATION TECHNIQUES

Our hypothesis is that the introduction of personalization techniques, especially collaborative filtering, into a traditional search engine will noticeably improve the quality of the results returned. We tested our hypothesis by conducting a blind user study. The study consisted of two disparate, yet unified searches. When the user performed a search, the results were returned in a table with two columns one column containing search results obtained using personalization, including collaborative filtering, while the other column containing search results obtained without personalization.

4.1 User Study

We focused our user study on the typical task of using a search engine to find answers to questions with various levels of specificity. We chose questions in a limited domain, that of dinosaurs, in order to allow the Foible spider to obtain a significant level of depth in the result set it could return. The users were asked to answer five questions:

(1) You are being attacked by a Velociraptor. What sort of nearby dinosaurs could you point him toward to distract him (i.e. that he would like to eat more than you)?
(2) Was the Styxosaurus an aquatic animal or land animal?
(3) What modern day class of animals did the *Archaeopteryx lithographica* evolve into?
(4) What dinosaur family did the Carnotaurus belong to?
(5) What are some common theories about why the dinosaurs became extinct?

The questions were chosen to please a diverse user group on the basis of both difficulty and the size of the data set returned. For example, the Styxosaurus question retrieved a relatively small set of data, which is apparent in the results, while the Velociraptor question had a relatively large result set. In terms of difficulty, the *Archaeopteryx lithographica* question is much more difficult to find answers to than the one about the extinction of the dinosaurs.

Using these questions, the users set out to find the answers to the questions using Foible. The results of the search were displayed in two columns in the browser window, as shown in Figure 3. Set A, the left column, contained the results made using personalization, while Set B contained the results using word frequency alone.

Once given the results, the users were instructed to visit at least three links from each result set and to rate each page visited based upon relevance to the question asked. We felt that if the users knew what the system used as parameters, it may have skewed the results, thus the specifics about how Foible created the two result sets was not announced in the instructions to the user. At the conclusion to the study, the users were asked to complete the exit survey as shown below in Figure 4. The survey was used to correlate how well our system adapted and to determine

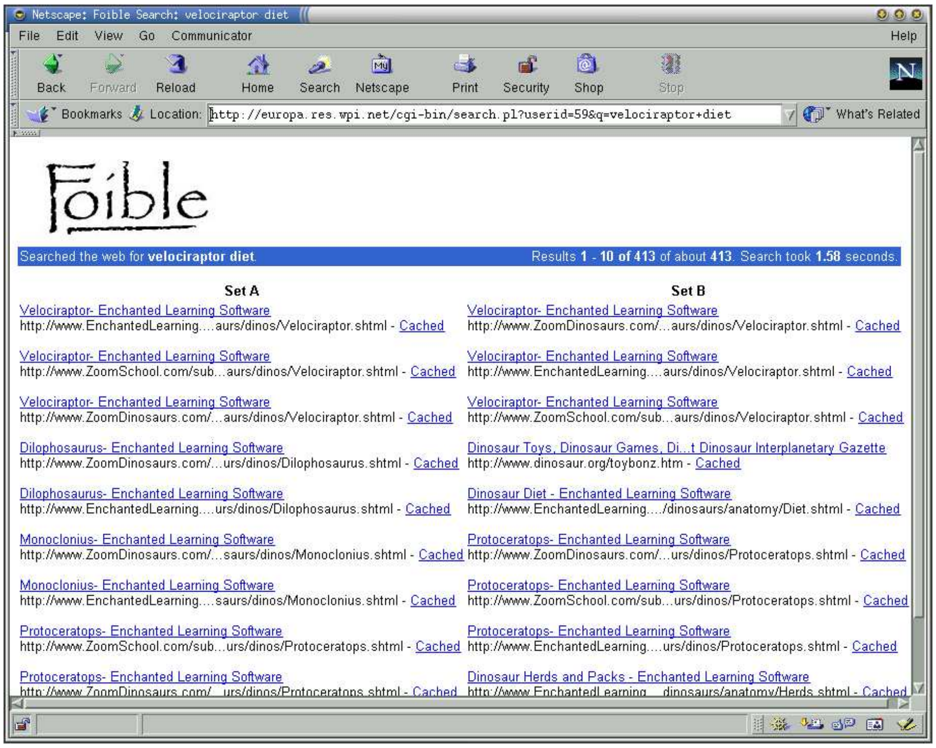

Fig. 3. Example Search Results

if the changes in the page ranking due to personalization were perceptible to the users.

A total of 55 users participated in the user study. These users consisted mainly of friends and family with computer and Internet experience ranging from beginner to expert. Each user session took approximately twenty minutes to completed, and the entire user study lasted about a week and a half.

4.2 Measures of Performance

After gathering data, we needed to accurately assess the "value" of a search result set. We define a "ranking scale" by which a search result set can be rated. For simplicity, the value of any result set is normalized to values between 0 and 1.

4.2.1 *Result Set.* We incorporate a "relative rank" into any equation used to determine a result set's value. If a user ranks only one link in a result set, there needs to be a factor which differentiates between the same rank in different positions within the result set. This differentiation is determined by weighting the user defined rank based on position. Based on pilot studies, we determined that the rank given to results toward the top of a result set should be more influential in determining the value of the set as a whole. After some pilot studies, it was decided that the weighting should be sinusoidal as opposed to linear, allowing results higher in a result set to be given proportionally more weight than would a simple linear

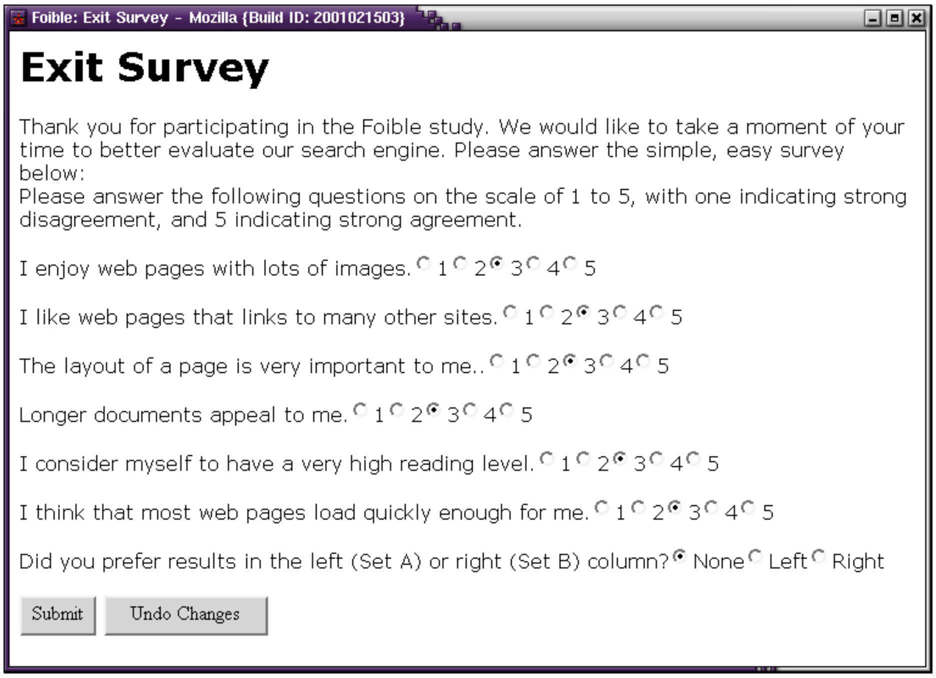

Fig. 4. The Exit Survey

approach.

We use the sum of all explicitly ranked pages in the set R that have been weighted by a weighting function. The value for the set is then normalized across the "ideal" ranked set S in which all values are assumed to be "5", which is the maximum explicitly ranked value, for all ten possible set positions.

In order to compute the rank for an entire set, the equation is:

$$\frac{\sum_{i \in R}(r_i \cdot w_i)}{\sum_{j \in S}(5 \cdot w_i)} \tag{7}$$

The weighting algorithm is a function of *cos* and is:

$$\cos\left(\left(\frac{\pi}{2}\right) \cdot \left(\frac{i}{10}\right)\right) \tag{8}$$

where the i represents the zero based index of the ranked page (thus values are between 0 and 9) and the value 10 represents the number of positions in an entire result set. Thus, the weighting appears as in Figure 5.

This equation yields a value for a set of ranked pages within a result set. The value obtained from this calculation can be used to correlate the "relevance" of result sets to users' search queries. During analysis, this correlation is then be used to determine whether it is beneficial to incorporate elements of personalization within a search engine framework.

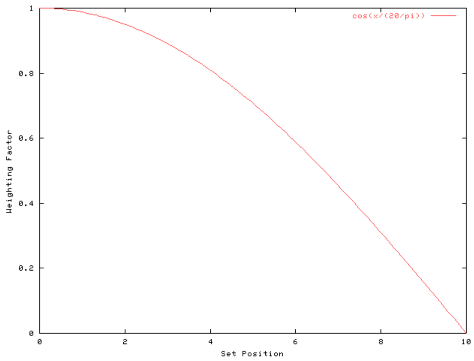

Fig. 5. Weighting Method for Result Sets

4.2.2 *Ranking.* A slightly different method of looking at the quantitative data is to chart user ratings per page ranking. All votes for the highest ranking page would be averaged together. Then all votes for the second highest ranking page would then be averaged, and so on. Eventually, all ranking pages would have a set of average user ratings for that particular rank. A well ordered result set should have a high average user rating for the highest ranking pages, and then it should slowly drop off in rating, in a smooth, linear fashion. A poorly ordered result set would have the average user ratings looking more like random noise, with no visual structure. Second degree polynomial regression lines can show a downward trend, as well as other structures of the data.

4.2.3 *User Preference.* We obtained subjective opinions and impressions on the Foible search engine through pre- and post-use surveys. The main results we report are the indicated preference for the result set that used personalization techniques or the result set based on solely word frequency.

5. ANALYSIS

We did three sets of analyses: 1) ranking analysis and 2) result set analysis attempt to determine by use of user rankings, if the results using personalization, including collaborative filtering, outperformed those that use word frequency alone; and 3) user survey analysis to measure the correlation between the user profiles and the feedback in the surveys.

5.1 Ranking Analysis

A well designed search engine should produce relevant results in the proper order. Using the ratings provided from the user study, we developed correlations between the relative rank of the pages and user ratings. For each query performed by the user for a particular objective question asked, the ratings were grouped together by page rank. All votes for a the first ranked page were averaged together, while all votes for the second highest ranking page were averaged together, and so on. If properly ranked, the average rating of a page should decrease as the page rank becomes worse, meaning the top ranked page should, on average, have a higher user rating than the second ranked page, and so on.

5.1.1 *Velociraptor.* The users were asked to determine what the Velociraptor primarily ate. Figure 6 shows the average user rank for each returned URL's position on the results page. The line with the squares shows the average rating for the result set that was produced solely from word frequency analysis. This is what a typical search engine would produce. Its polynomial regression line is the dotted line. The average rating for the result set using profiles is the line with the diamonds. Its polynomial regression line is the dashed line. Similar data and regression lines are used in subsequent graphs in this subsection.

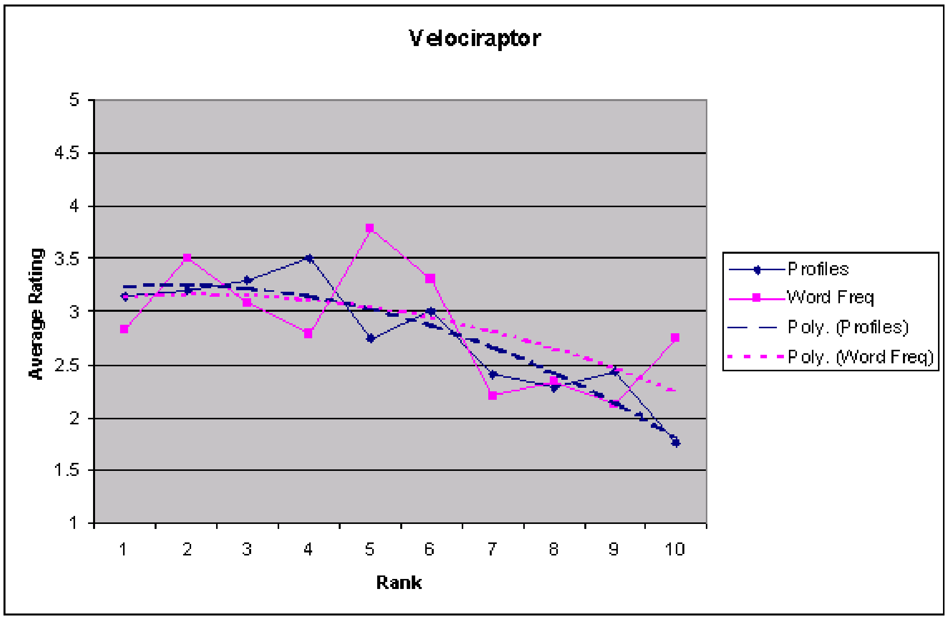

Fig. 6. Velociraptor Result Chart

Using the polynomial regression lines as a guide for analysis, the result set based on user profiles is shown to be in a more proper order. Although both data sets have some degree of downward progression, the regression line for the word frequency plot has a flatter slope than the line for the profiles. This can be attributed to a data set that is more random than linear. To affirm this postulation, looking at the actual data for the word frequency shows that it has random attributes associated with it. The average rating for the highest ranked page (the page whose rank is 1) is approximately equal to the lowest ranked page. The highest rated page, which, in an ideal situation, should be ranked first, was actually ranked fifth. While the data based on the profiles result set shows that it is more properly ordered. It has a regression line that clearly shows a downward trend which implies an ordering that is better than the data set using word frequency alone.

5.1.2 *Styxosaurus.* The next question users were asked concerned the Styxosaurus habitat. Figure 7 shows the average user rank for each returned URL's position on the results page. The data set using word frequency analysis was already fairly well ordered. However, there were a few data points, namely the pages with a rank of eight and nine, that appeared in an improper order. Personalization improvements smoothed the graph by either removing offending data points or by increasing their rank so that it resulted in a more proper total ordering for the result set.

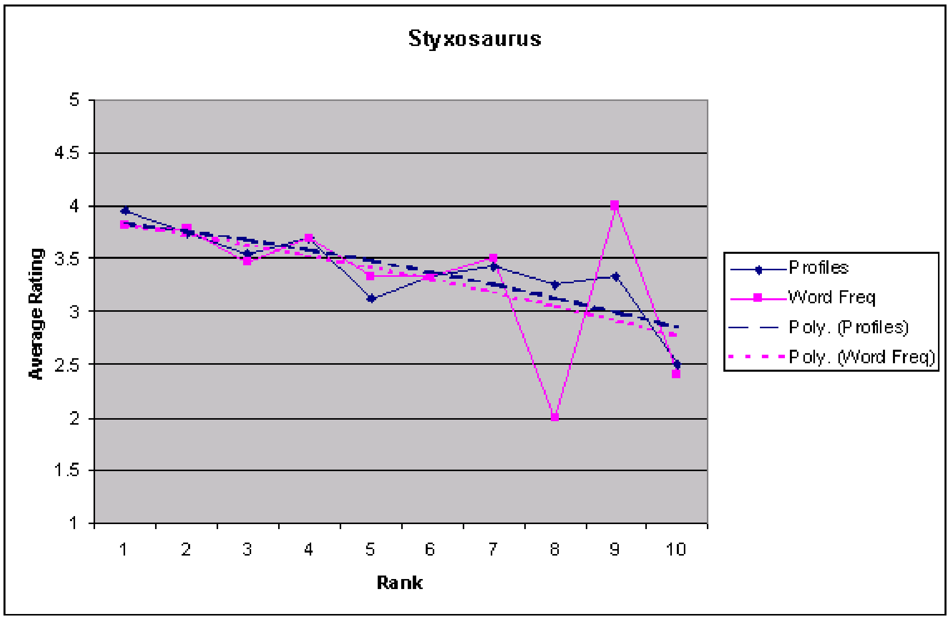

Fig. 7. Styxosaurus Result Chart

5.1.3 *Archaeopteryx lithographica.* The users were then asked to find out what modern animal is associated with *Archaeopteryx lithographica*. Figure 8 shows the average user rank for each returned URL's position on the results page. Analysis of the results from this question provides an excellent example of how personalization can correct flaws in the result sets returned by search engines that use word frequency-only. The first ranked page for the word frequency data set is actually one of the lowest rated pages. The highest rated page is near the middle of the data set. As a result, the polynomial regression line is shaped like an upside-down letter *U*. The graph of the data set using personalization shows much more linearity as it generally moves from a high rating to a lower rating without too many outliers far from this trend.

5.1.4 *Carnotaurus.* After that, the users were asked to find to what family the *Carnotaurus* belongs. Figure 9 shows the average user rank for each returned URL's position on the results page. This chart is interesting because it seems as though, initially, that the profiles-based data set is doing well, as opposed to the word frequency-based data set. However, near the end, there is an outlier with

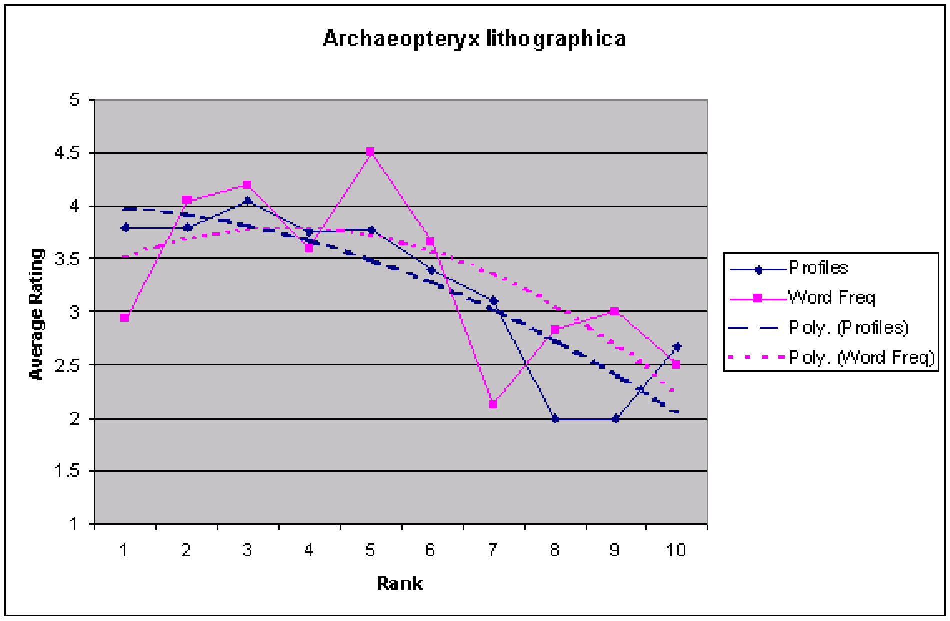

Fig. 8. Archaeopteryx Lithographica Result Chart

the profiles data set that should not be there. Instead of keeping the trend of being relatively stagnant, it suddenly moves up quite a bit considering that these are averages. Up until that point, the profiles data set seemed to have avoided the misplacement of the page that was ranked fourth by moving it to its proper location.

5.1.5 *Dinosaur Extinction.* For the final question, the users were asked to determine the current theories of the dinosaur extinction. Figure 10 shows the average user rank for each returned URL's position on the results page. Again, the profiles data set starts out very strong in comparison to the word frequency data set. However, there is an outlier at the seventh rank position that contaminates the relatively clean looking data set. The polynomial regression lines do show that overall, the profiles based data set performs better than the word frequency data set. Similarly to the results on the *Archaeopteryx lithographica* question, the word frequency polynomial regression line is shaped like an upside-down letter *U*, which does not comply well with a properly ranked set of results.

5.2 Result Set Analysis

Another quantitative means by which we tested the benefits of personalization for Web search was an evaluation of the search result sets returned for each user's queries. Using the formula described in Section 4.2.1, each result set was assigned a normalized value from 0 to 1. The data collected from both the profiled queries and the word frequency queries was then compared in several different ways to assess the effectiveness of the system. Figure 11 shows the average result set ratings for

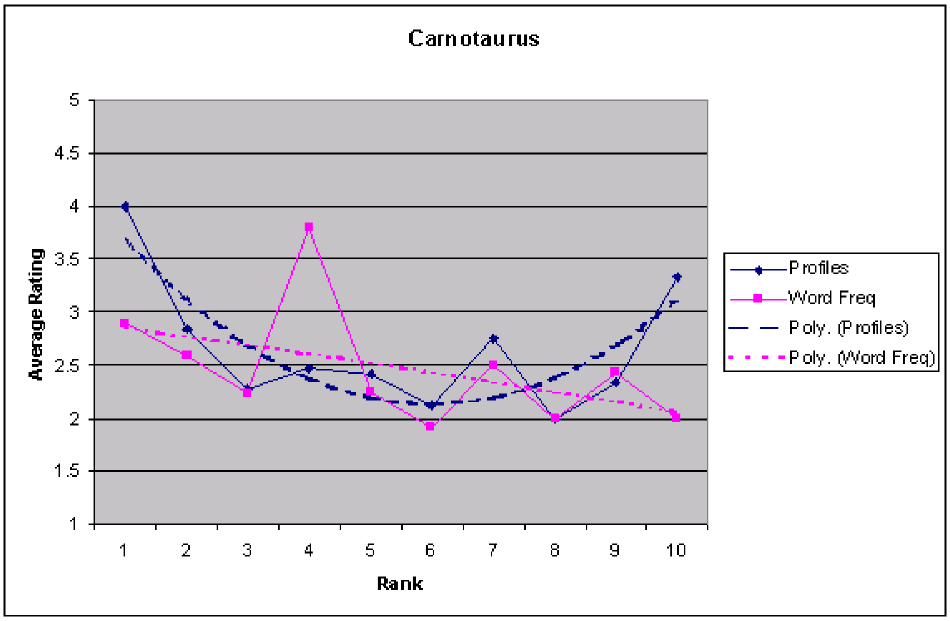

Fig. 9. Carnotaurus Result Chart

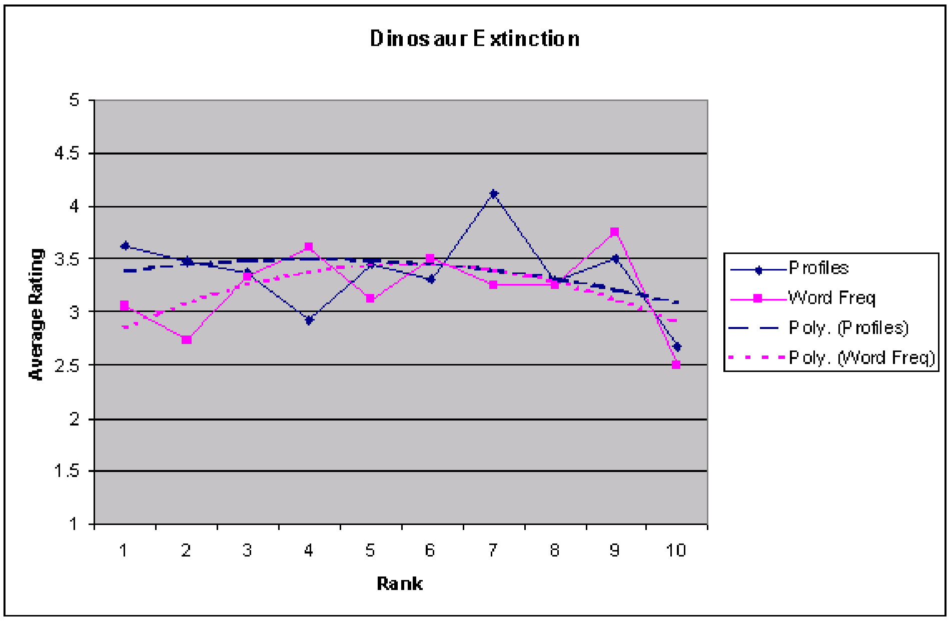

Fig. 10. Dinosaur Extinction Result Chart

the profile and word frequency result sets. It can be seen more clearly how a profiled search returned more relevant information.

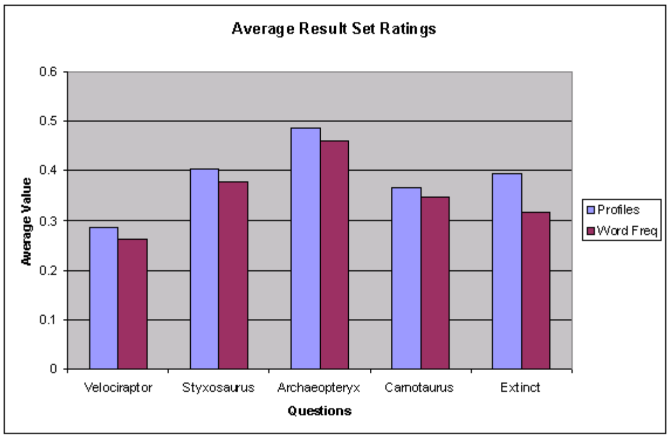

Fig. 11. Average Result Set Ratings per Search

5.3 Surveys

In addition to the above methods of analysis, users were asked to fill out a survey (shown previously in Figure 4). First, the series of questions were asked to the user to self-evaluate their browsing preferences. These questions corresponded to different attributes that make up the user profiles. If a user responded that they enjoy documents with many links and a high reading level, then the values of their profile should show a similar preference. This was tested by comparing each user's survey response with their profiles. Each attribute of the profile was analyzed to find its distribution percentile, and was compared with the user's input. User responses were mapped to desired percentiles by considering 5 to indicate a preference for values in the top percentile, and 0 to show preference for the bottom percentiles. A linear map between these two poles allowed us to convert user survey responses into percentiles, and then match these with profile data. Figure 12 shows results of this analysis.

All users showed a high level of agreement between their self-rated scores and the preferences that were determined automatically by the system during testing. This level of correlation leads us to conclude that the algorithms used to adjust the user profiles during page rating accurately reflected the real preferences of the user.

The second aspect of the survey was a question that asked the user which column of the two column search results display generated the proper results (remember, the users were "blind" as to how the result sets were generated). Figure 13 depicts these results. Users preferred the results presented by the personalized ranking algorithms 31% of the time. Not a single user preferred the traditional search engine to search engine using personalization. However, many users, up to 69%, saw no noticeable difference between the two columns, possibly due to the limited number of searches that users performed in our user study.

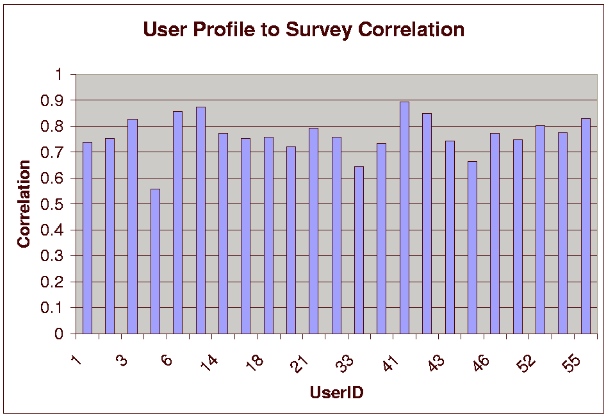

Fig. 12. User Profile to Survey Correlation

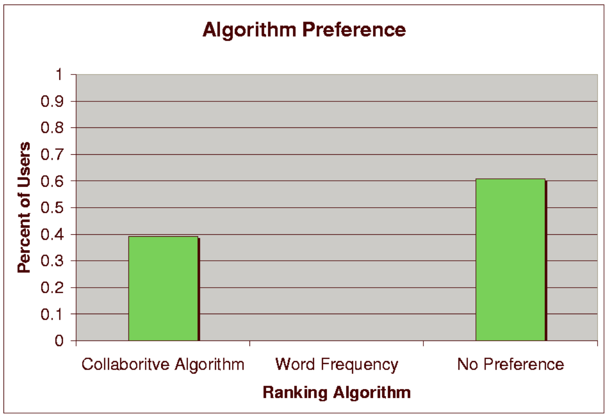

Fig. 13. User Algorithm Preference

6. CONCLUSION

Present day search engines, in general, do not have a concept of a stateful user. By introducing personalization into a typical search engine, it may be possible to produce better results by remembering user preferences and collaboratively pairing users with others with similar interests. In order to assess some of the potential that personalization may provide to traditional search engines, we have developed a custom search engine called Foible. Based on Google, Foible uses search engine technologies common in many search engines with the enhancements of personalization based on document size, visual style and document detail. In addition, Foible moves beyond basic personalization by applying collaborative filtering in using the past agreements of users to enhance search result sets.

To evaluate the benefits of personalization and collaborative filtering to web search, the Foible spider indexed over twenty thousand documents, and finished with nearly a gigabyte of data stored in our database. This data was later indexed, and processed to produce the database back-end used for searches. The ranking algorithm used combined word-frequency, matching of page properties to stored user preference, and the prediction of interest based on correlating similar users. A carefully users study had over 50 users perform specific tasks using Foible, evaluating pages returned based on personalization and pages returned without considering personalization.

Our analysis of the user study results show advantages to using personalization in Web search. For those users that ranked enough pages for the system to distinguish their profile from a default profile, the "relevance" of their personalized result set was marginally higher than then set obtain by a simple word frequency search. For those users with fewer ranked pages, the personalized result set closely mirrored the result set from a word frequency search. This showed that Foible's personalization maintained a level equal to or higher than the relevance of the simple word frequency search.

Additionally, the average data showed clearly that the personalized engine outperforms the word frequency search. By averaging the values for each search, thus reducing the effect of statistical outliers and individuals with few ranked pages, it could be seen that overall values for result sets returned by the system were higher when personalization was involved.

Polynomial regression line analysis showed our personalization techniques to perform consistently better than word frequency analysis. Particularly within the first three positions of the returned results, the personalization techniques show a clear advantage. In examining the results produced by word frequency alone, user preference were randomly distributed across the top ten entries, instead of being concentrated at the top results. Personalization techniques addressed this problem by allowing dynamic reordering of search results, based on the feedback generated by the user. The result is a more uniform distribution of the top ten elements, with the highest ranked (in terms of user votes) elements appearing in the top positions.

Surveys filled out by users provided confirmation that the adaptation algorithms used by the search engine were working properly. After using the search engine, user's profiles were automatically adjusted to reflect the content that the system believed the user was interested in viewing. When the users themselves provided this data, we found an average of 70-80% correlation between the system's predictions and preferences stated by the user.

Additionally, analysis of user surveys shows that users overwhelmingly prefer the collaborative search techniques to traditional methods of searching. When presented with two columns showing the results produced by either method, of those who were able to discern a difference between the two columns, *every* user, without exception, preferred the column representing the collaborative techniques.

In summary, Foible represents a working implementation of a search engine with personalization enhancements, including collaborative filtering. The implementation of Foible gave us an opportunity to test the hypothesis that Web search can be improved through personalization and collaborative filtering. Our user study shows that users prefer results returned by a personalized filter with collaborative filtering to those returned by traditional search engines. Our other data also supports our original assertion that collaborative filtering provides a more personalized search experience that results in better rankings.

7. FUTURE WORK

Despite the successes of our user study, we have identified several aspects of the system that could be improved. Many of these stem from the fact that our system was intended primarily as a proof-of-concept implementation. While we do believe that the underlying technology is ready for production deployment, there are several improvements that must be made before widespread adoption of these techniques occurs.

7.1 Scalability

The algorithms used within our user study were not designed to scale to hundreds of thousands of users. Unfortunately, the computation of user-to-user correlation grows exponentially with the number of profiles stored in the system. New algorithms or techniques would need to be explored for scaling into thousands of users. Possible improvements could include precomputing user correlations at intervals, rather than on the fly as our current system implements. Additionally, it might be possible to introduce group functionality that would artificially constrain the number of correlation computations that would need to be performed.

7.2 Increased Domain

Because we were working within the confines of limited resources, we were not able to crawl as large a section of the web as originally desired. The Foible spider ran for almost two weeks, and amassed 950 megabytes of data within our database. It would be interesting to architect a better back end for data storage capable of handling hundreds of thousands, and multiple tens of gigs of data storage. The actual amount of data composing the entire Web is a truly staggering quantity, and developing effective means of cataloging and storing it would certainly be rewarding Future work.

7.3 Expanded User Study

During the analysis of the data we obtained from the user study, it became clear that the system was better able to distinguish users once they passed the "sixth vote" mark. After the user has rated six votes on each set (profiled and word frequency results sets), or a total of 12 votes, the system shows a greater separation between the values of their result sets for profiled queries and word frequency queries. It would be worthwhile to expand the user study to encompass the ranking of groups with many different numbers of pages each. In this manner, the system's learning rate can be charted. It would be interesting to know exactly how fast the user was meaningfully distinguishable to the system.

7.4 More Attributes

We were only able to create a limited number of attributes that characterized Web pages. While we believe that our choices of attributes, such as readability, image content, document length, etc., provided a reasonable cross section, the accuracy of the correlations between users could be increased by introducing more attributes. Suggestions for these include color, image analysis, and better means of analyzing the text of a document. Most of the text analysis indexes used, such as Fog and Flesch-Kincaid, are designed to analyze dense blocks of well structured text. Often, navigational elements of Web pages are analyzed as broken sentences, and thus adversely influence the computation of these text-based indexes.

REFERENCES

BALABANOVIC, M. AND SHOHAM, Y. 1997. Content-based, collaborative recommendation. *Communications of the ACM 40,* 3 (Mar.).

BREESE, J. S., HECKERMAN, D., AND KADIE, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. Tech. Rep. MSR-TR-98-12, Microsoft Research. Oct.

BRIN, S. AND PAGE, L. 1998. The anatomy of a large-scale hypertextual web search engine. In *The 7th International World-Wide Web Conference.* [Online] at http://www7.scu.edu.au/-programme/fullpapers/1921/com1921.htm.

CHAN, P. 1999. A Non-Invasive Learning Approach to Building Web User Profiles. In *ACM Workshop on Web Usage Analysis and User Profiling.* Springer-Verlag, 7 – 12.

FLESCH, R. 1949. *Art of Readable Writing.* New York, Harper.

GOECKS, J. AND SHAVLIK, J. W. 1999. Automatically Labeling Web Pages Based on Normal User Actions. In *Procedings of the IJCAI Workshop on Machine Learning for Information Filtering.*

MILES, T. H. 1990. The fog index: A practical readability scale. *Critical Thinking and Writing for Science and Technology.*

RUCKER, J. AND POLANCO, M. 1997. Siteseer: Personalized Navigation for the Web. *Communications of the ACM 40,* 3 (Mar.), 73 – 76.

SULLIVAN, D. 2003. Nielsen NetRatings Search Engine Ratings. [Online] via http://searchenginewatch.com/reports/.

THOMAS, C. AND FISCHER, G. 1997. Using agents to personalize the web. *Proceedings of the 1997 International Conference on Intelligent User Interfaces.*

WASFI, A. M. A. 1999. Collecting User Access Patterns for Building User Profiles and Collaborative Filtering. In *Proceedings of the International Conference on Intelligent User Interfaces.* 57 – 64.

ZAKON, R. H. 2003. Hobbes' internet timeline v6.0. [Online] at http://www.zakon.org/robert/-internet/timeline/.