CLIS-TR-00-01 May 2000

Using Implicit Feedback for User Modeling in Internet and Intranet Searching[j]

Jinmook Kim,[*] Douglas W. Oard,[*] and Kathleen Romanik[†]

[*]College of Library and Information Services
University of Maryland, College Park, MD 20742-4345
{jinmook, oard}@glue.umd.edu

[†]powerize.com
901 Elkridge Landing Road, Suite 350
Linthicum, MD 21090
kromanik@powerize.com

Abstract

Powerize Server 1.0, developed by Powerize.com, is a content-based information filtering and retrieval system that presently uses a manually constructed user model known as a search profile. User modeling captures a user's information needs. A user model can be constructed explicitly by the user or implicitly by exploiting feedback from the user about which documents are relevant. Implicit feedback can be inferred from user behavior without any additional work on the part of the user. The study reported in this paper investigates a way of implementing the implicit feedback technique of user modeling for the Powerize Server 1.0. Previous studies on Internet discussion groups (USENET news) have shown reading time to be a useful source of implicit feedback for predicting a user's preferences. In this study, we examined: 1) whether reading time is useful for predicting a user's preferences for academic or professional journal articles, and 2) whether printing behavior adds anything to what we already know from reading time. Two experiments were conducted with undergraduate students using professional articles from the telecommunications and pharmaceutical industries. The results of the experiments showed that reading time could be used to predict the relevancy of documents, although the threshold on reading time required to detect relevant documents would be higher than for USENET news articles. The experiments also showed that printing behavior adds to what can be inferred from reading time. All the documents that were printed in the experiments were relevant, but the reading time for many of these documents was below the mean reading time for all documents read. This result implies that the use of printing behavior with reading time could increase the precision and recall ratios for detecting relevant documents. Suggestions for incorporating the results of the study into the Powerize Server were made in conclusion. This paper also reports detailed technical descriptions of the experiment design, including research problem, experimental system, and data collection.

[φ] The research reported herein was supported in part by the Maryland Industrial Partnerships program and powerize.com

1. Introduction

Millions of people around the world, playing their roles as both the providers and users of information, are connected to the Internet. As the information on the Internet is increasing and changing, people are now faced with the problem of finding useful information within the panoply of sources available to them. It is the classic needle in the haystack problem, and there are now even too many haystacks. Information filtering is a process of finding the needle in the changing haystacks.

Information filtering systems, like retrieval systems, are designed to help users find the information they need and present it to the users in a timely manner. Although the distinction between information retrieval and filtering is often not clear, they can be differentiated using the concepts of "push" and "pull." Information retrieval is a "pull" service that users search for information they need from the system, whereas information filtering is a "push" service that the system finds new information and presents it to the user. Existing information filtering systems can be classified into two forms: content-based and social (which is also called collaborative). Content-based filtering systems select documents based on the characteristics of the document, whereas social filtering systems choose documents based on ratings and annotations from other users (Sheth, 1994). In this report we focus on content-based filtering systems.

Individual users seeking information may have different needs and preferences. User modeling, which captures the different needs of individual users, is a central component that a filtering system must have to perform the task of providing a personalized information service for its user. Current filtering systems have adopted one of two approaches for user modeling: explicit user modeling and implicit user modeling. Explicit user models are relatively simple to implement because they are constructed explicitly by the user. Implicit user models, by contrast, exploit feedback about desirable and undesirable documents from the user to develop or improve the user model. In some application it may be impractical for users to give explicit feedback, since this would take time away from their tasks. Implicit feedback, inferred on the basis of user behavior, offers the potential to reduce this cognitive load. It is thus a natural source to consider when constructing an implicit user model for text filtering systems (Stevens, 1993; Morita & Shinoda, 1994; Konstan et al., 1997; Nichols, 1997; Oard & Kim, 1998).

Powerize Server™, developed by powerize.com, is a content-based text retrieval and filtering system that searches multiple internal and external information sources simultaneously and presents the retrieved documents to the user in a customized publication that can be viewed with a Web browser. Powerize Server™ presently uses an explicit user model. Once a user sets up a search profile, she can choose to save the profile and have it re-executed on a regular schedule. In this report, we explore the value of alternate sources of implicit feedback that could be used to improve this initial user model over time. The behaviors that are measured should, of course, be both easily observed and useful as sources of insight into a user's preferences. Previous studies on Internet discussion groups (USENET news) have found that predictions based on reading time can be about as accurate as those based on explicit ratings (Morita & Shinoda, 1994; Konstan et al, 1997). In this report we describe the results of experiments that examined: i) whether reading time is also useful for predicting explicit ratings for academic or professional journal articles, and ii) whether retention behavior adds anything to what we already know from reading time.

Once a user model has been created by any means, it can be used to predict the value to the user of future documents found by the Powerize Server. This knowledge, then, can be used in several components of the system:

- To identify specific information sources that should be searched for potentially useful documents,
- To decide whether or not to select a document for inclusion in the publication that is presented to the user,

- To rank the documents, which will determine where they are placed in the publication, and
- To decide whether or not to produce a summary of a document.

With the ability to refine a user model using implicit feedback, Powerize Server™ could provide users with a more personalized information system. We expect that powerize.com may be able to use the results of our experiments to improve the effectiveness of the Powerize Server™.

2. Background

2.1 Content-Based Filtering

Content-based filtering systems represent and detect documents based on information that is derived from document contents. Many techniques from information retrieval, therefore, can be applied to designing and implementing content-based filtering systems. A content-based filtering system consists of four processes: profile processing, document processing, detection processing, and evaluation processing. Figure 1 shows what tasks each process performs and how they interact with each other.

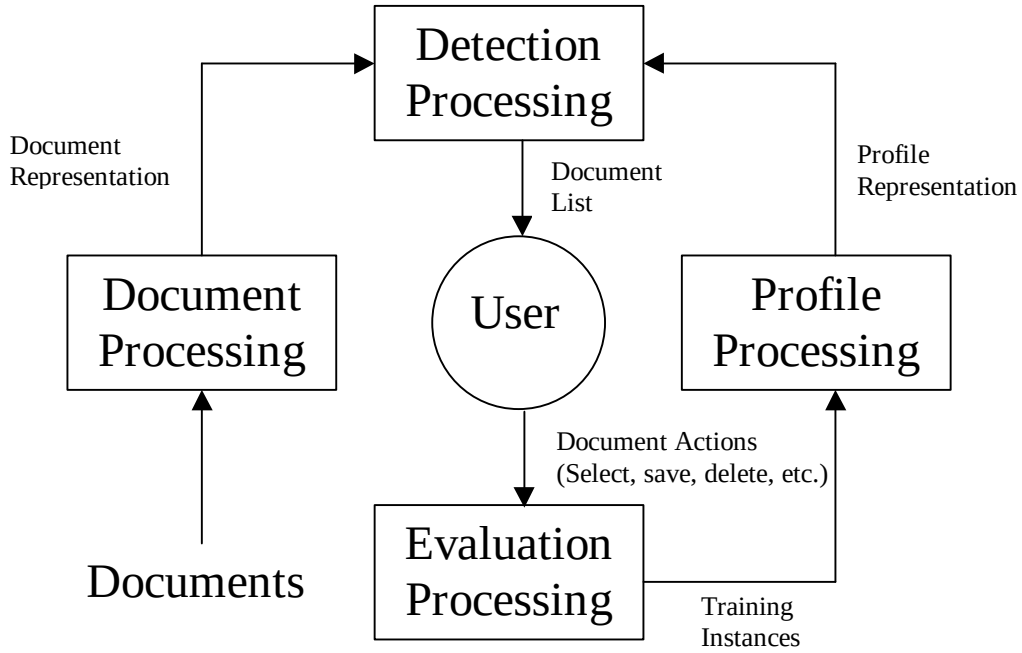


Figure 1. Content-based filtering system model

Profile processing refers to defining the information needs and modifying profile representations for each user. An information seeking process begins with a user who has an information need. In an automated system, the information need must be transformed into a query that consists of search terms. Once the query is formed, a representation of the query is required so that the system can find relevant documents. An explicit profile describing the user interests is typically initially acquired from the user. The profile can then be automatically modified using information obtained from evaluation processing.

Document processing refers to representing each document in a collection. A system searches for information that satisfies the user's needs by comparing their profile with a representation of each document in the collection. In content-based information filtering systems,

documents are typically represented based on the occurrences of individual words (and perhaps phrases). Weights reflecting the importance of each term can be based on the number of times that the term occurs in the document, the length of the document, and the number of documents in which the term appears.

The set of documents best matching the user's needs is found through detection processing. Once both user needs and documents have been represented, the system finds relevant documents by comparing the user profile with the document representations. There are three main techniques that can be used in the detection processing: Boolean matching, the vector space method, and probabilistic matching. In Boolean systems, which are based on searching for an exact match between the profile and the document, a given document either satisfies a Boolean expression or it does not. Boolean systems have the advantage that users familiar with Boolean logic can generally understand the relationship between the profile and the documents that are detected. Boolean systems, however, provide little basis for ranking the retrieved documents, since they operate on the basis of the presence or absence of terms. Two common approaches to ranked output generation are the vector space method and probabilistic matching. In the vector space method, both the document and query are represented as vectors in a high-dimensional vector space formed by computing a weight for each term, and then using those weights as the coordinates of the document in the vector space. Similarity measurements between the document and query are then based on either the Euclidean distance or the angle between the vectors. This design reflects the intuition that the documents with vectors that are nearest the profile vector are most likely to address topics that are similar to those that are desired (Korfhage, 1997). The probabilistic method, by contrast, seeks to estimate the probability that a document satisfies the information need represented by the profile. It is thus in some sense a generalization of the exact match idea, in which the system seeks to rank order documents by the probability that they satisfy the information need rather than by making a sharp decision (Turtle and Croft, 1990).

Evaluation processing seeks to gather evidence about the user's satisfaction with the documents that are provided by the system. Users will typically examine the set of documents that result from detection processing and select documents that are interesting to them. Evaluation processing begins with this selection process, which itself is a source of implicit evidence about the desirability of the selected document. Evaluation processing can include the observation of examination, retention, and reference behavior, inference of implicit ratings based on those observations, and collection of explicit ratings from the user. The process is iterative, feeding back to profile processing.

2.2 Sources for User Preference: Explicit and Implicit Feedback

Explicit feedback takes place when ratings are collected directly from the user in an information filtering system. SIFT, Tapestry, and GroupLens are some examples of information filtering systems that use explicit feedback (Yan & Garcia-Molina, 1995; Goldberg et al, 1992; Konstan et al., 1997). Although explicit feedback is easily implemented, the increased cognitive load associated with explicitly assessing the usefulness of individual documents could serve as a disincentive in some applications. This, in turn, can limit the opportunities for profile learning, and thus the usefulness of the entire filtering system.

Implicit feedback may, of course, bear only an indirect relationship to the user's assessment of the usefulness of any individual document. But because it is easily collected, it could ultimately have even more potential to support profile processing than explicit feedback. InfoScope, which was a system for filtering Internet discussion groups (USENET News), utilized both implicit and explicit feedback for modeling users (Stevens, 1993). InfoScope used three sources of implicit evidence about the user's interest in each message: whether the message was read or ignored, whether it was saved or deleted, and whether or not a follow up message was

posted. In his study, Stevens observed that implicit feedback was effective in tracking long-term interests because it operates constantly without being intrusive.

Morita and Shinoda proposed a profile processing technique to accumulate a user's preference for information based on behavior monitoring (Morita and Shinoda, 1994). An experiment over a six-week period with eight users was conducted to determine whether a user's preference for Internet discussion group (USENET news) articles was reflected in the time spent to read those articles. The result of the experiment showed a strong positive correlation between reading time and explicit feedback provided by those users on a four-level scale: " A; very interesting," "B; interesting," "C; neither interesting nor not-interesting," and "F; not interesting." They also discovered that interpreting as 'interesting' articles that the reader spent more than 20 seconds reading actually produced better recall and precision in a text filtering simulation than using documents explicitly rated by the user as interesting.

GroupLens also addressed the potential benefit of implicit feedback (Konstan et al., 1997), although in this case it was used with social filtering. An experiment was done in 1996, using a limited number of Internet discussion groups (USENET news), to apply a news reader software for a user to enter explicit ratings and receive predictions. Over a seven-week trial, 250 registered users submitted a total of 47,569 ratings and received over 600,000 predictions for 22,862 different articles. Specially modified news browsers were provided that accepted explicit ratings and displayed predictions on a 1-5 scale where 1 was described as "this item is really bad" and 5 as "this article is great, I would like to see more like it." Their study showed that predictions based on time spent reading are nearly as accurate as predictions based on explicit numerical rating. They also suggested further actions, such as printing, saving, forwarding, replying to, and posting a follow up message to an article, as sources for implicit ratings.

Nichols presented a list of potential types of user behaviors that could be used as sources for implicit feedback, such as purchase, assess, repeated use, print/save, delete, refer, reply, mark, examine/read, glimpse, associate, and query (Nichols, 1997). Among the actions he presented, the 'refer' behavior contains all those instances where one information item links to another item, including traditional academic citations as well as hyperlinks on Web pages or the threaded links between USENET news articles. Citation indexing has been well studied in the field of information retrieval, and this appears to be a promising source for implicit feedback in some applications.

Category	Observable Behavior
Examination	Selection Duration Edit wear Repetition Purchase (object or subscription)
Retention	Save a reference or save an object - with or without annotation - with or without organization Print Delete
Reference	Object->Object (forward, reply, post follow up) Portion->Object (hypertext link, citation) Object->Portion (cut & paste, quotation**)**

Table 1. Observable behaviors for implicit feedback

Recently, Oard and Kim surveyed the state of the art in implicit feedback techniques with an eye toward their potential use for information filtering (Oard & Kim, 1998). Based on the sources of implicit feedback presented by Nichols, they identified three broad categories of potentially useful observations: examination, retention, and reference. Table 1 shows the identified observable behaviors under each category, and each is discussed in detail in the following section.

2.3 Observable Behaviors for Implicit Feedback

The category of "examination" in Table 1 consists of such user behaviors as selection, duration, edit wear, repetition, and purchase. Information systems often provide brief summaries of several promising documents using some sort of interface, and “selection” of individual objects for further examination can thus provide the first cue about a user’s interests. "Duration" is a generalized term for reading time, which can accommodate other modalities such as audio and video. Hill et al. (1992) defined “edit wear” as an analogue to the useful effects of uneven wear that physical materials accumulate over time that provide other users with cues that help discover useful materials and useful items within those materials. In text browsing, for example, edit wear might be measured by using dwell times at specific locations in the text to characterize scrolling behavior. Examination may extend beyond more than a single interaction between user and system, which is described as “repetition.” Finally, when information access is priced on a per-item basis, purchase decisions offer extremely strong evidence of the value ascribed to an object. Similar information would be available at a somewhat coarser scale when users purchase subscription access to certain types of content (e.g., subscription to a separately priced cable television channel).

The category of “retention” is intended to group those behaviors that suggest some degree of intention to make future use of an object. Bookmarking a web page is a simple example of such a behavior, and “save a reference” is a generalized term that can accommodate a wider range of actions such as construction of symbolic links within a file system. Rucker & Polanco (1997), for example, constructed a recommender system using bookmark lists. Saving the object itself is the obvious alternative, something Stevens (1993) used as a source of implicit feedback. In either case, the object may be saved with or without some form of annotation. For example, web browsers typically default to using the page title in the bookmark list, but users may optionally provide a more meaningful entry if they desire. Although numerous confounding factors would likely be present, it may be possible to infer something about the value a user places on an individual page by whether or not they go to the trouble of constructing an informative bookmark entry. Similarly, users may choose to save a reference or an object in an explicitly organized fashion or in the default manner. “Print” has been grouped with retention because of the permanence of the printed page, but users may also print document or images to facilitate examination because paper still has some decided advantages over electronic displays in many applications. Printing overlaps with the next category (reference) as well, since users may print a document or image with the intention of forwarding them to another individual or including portions in another document. Nevertheless, printing is often associated with a desire for retention, so we find this grouping useful. As with examination, it may be possible to infer something about the portions of a document that the user finds most valuable from the portions which he or she chooses to print. Finally, the retention category is distinguished by the possibility of directly observing evidence of negative evaluations as well. When retention is a default condition, as in some electronic mail systems, a decision by the user to delete an object might support an inference that the deleted object is less valued than other objects that are retained.

Each activity in the “refer to” category has the effect of establishing some form of link between two objects. Forwarding a message, for example, establishes a link between the new message and the original. Similarly, replying individually or posting a follow up message to some

form of group venue such as a mailing list establishes the same sort of link. Goldberg et al. (1987) described a simple example of this in which users could construct an electronic mail filter to display messages that their colleagues had taken the time to reply to. Hypertext links from one web page to another and bibliographic citations in academic papers create links from a portion of an object (characterized, perhaps, by some neighborhood around the link itself) to another object, although the refinement to a portion of a document has not been exploited often. Brin & Page (1998) provide an example of how hypertext links might be used, although their focus is on population statistics rather than individual preferences. Garfield (1979) describes the design of retrieval systems that are based on bibliographic citations. Alternatively, selective inclusion of another document, using either cut-and-paste or a quotation, creates a link from an information object to a portion of another.

3. Experiment Design

Although some preliminary studies on the use of implicit feedback have been done, we know little about the utility of observable behaviors other than reading time and citations for building user models. We thus chose to focus on retention behavior, asking in particular whether retention behavior added additional information that could not be inferred from examination behavior.

3.1 Overview

Figures 2a and 2b show alternative strategies for using observations to predict which future document a user will wish to see. Figure 2a depicts a modular strategy in which the inference stage seeks to produce ratings similar to those that a user would have explicitly assigned, and then the prediction stage uses those estimated ratings to predict future ratings. Konstan et al. (1997) adopted this perspective when evaluating how well observed reading time predicted explicit ratings for individual articles. Figure 2b shows an alternative strategy in which past observations are used to predict user behavior in response to new information, and then the inference stage seeks to estimate the value of that new information based on the predicted behavior. Stevens (1993) implemented a simple version of the strategy. He predicted the examination duration for a new USENET news article based on the examination durations for similar articles in the past and then constructed content-based queries that would select articles with long predicted examination durations. This essentially amounts to a degenerate inference stage in which desirability is assumed to increase monotonically with examination duration.

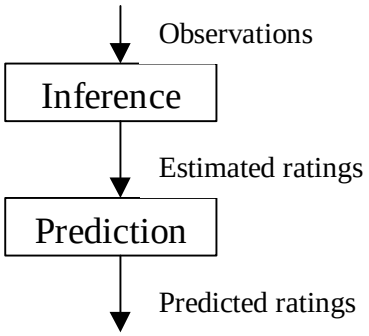


Figure 2a. Rating estimation strategy

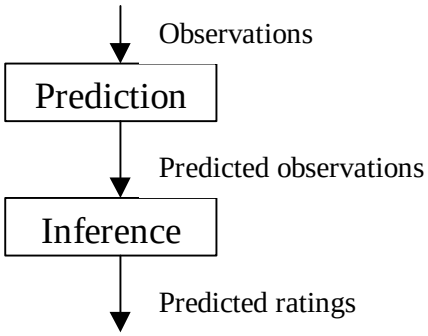


Figure 2b. Predicted observations strategy

We seek to predict ratings for new documents following the strategy shown in Figure 2a. Of the observable behaviors identified in Table 1 we have chosen to focus on reading time and printing behavior in this study. Table 2 shows the applicability of each observable behavior to the

Powerize Server application and provides an indication of the ease with which that behavior can be observed. Selection, reading time, repetition, saving and printing are appropriate to this application and measurable without modifying a Web browser, so that is the list from which we selected observable behaviors to explore. Since this was our first experimental study of the utility of implicit feedback, we tried to keep the experiment design relatively simple by choosing one examination behavior and one retention behavior. Reading time was the obvious choice for the examination behavior, both because it has been studied in other applications and because it can easily be measured with reasonable accuracy by instrumenting the web server. This avoided the need to obtain explicit ratings as ground truth for articles that users did not select, which would have been needed if selection behavior were to have been used. It also allowed us to develop a protocol in which each user participated in only a single session, avoiding the multiple session that would have been needed to study repetition behavior. We chose printing over saving behavior for similar reasons, since users would have no motivation to save articles unless multiple sessions were scheduled. Printing behavior was somewhat more difficult to measure than reading time because the server is not normally aware of printing behavior in Web-based applications. A modification on the Powerize Server was required to accomplish this.

Category	Observable Behavior	Applicability to Powerize Server 1.0	Ease of Measurement**
Examination	Selection	Yes	1
	Reading time	Yes	1
	Scrolling behavior	Sometimes*	3
	Repetition	Yes	1
Retention	Save	Yes	2
	Print	Yes	2
	Delete		N/A
	Purchase		N/A
Reference	Forward	Yes	3
	Reply		N/A
	Post follow up		N/A
	Hypertext link		N/A
	Citation		N/A
	Cut & Paste	Yes	3
	Quotation		N/A

* Scrolling behavior is not applicable when the length of articles is not long enough to do scrolling.
** Ease of Measurement: "1" indicates that it is measurable without modifying a Web browser/Powerize Server,™
"2" indicates that it is measurable by modifying either a Web browser or Powerize Server,™
"3" indicates that it is not measurable without modifying a Web browser.
"N/A" indicates that it is not applicable to Powerize Server™, thus not measurable.

Table 2. Observable behaviors using Powerize Server™

3.2 Hypotheses

The main goals for the current study were to:

- Determine whether reading time and printing behavior are good sources for implicit feedback that could substitute for explicit ratings in the context of filtering academic and professional journal articles, and
- Discover the relationship(s) that may exist between reading time, printing behavior, and explicit ratings.

Research hypotheses include the following:

a. Users will spend more time on reading relevant documents than on non-relevant ones.
b. A combination of reading time and printing behavior will be more useful for predicting explicit ratings than using reading time alone.

3.3 Experimental System

Powerize Server™ is a Windows NT Web server-based text retrieval and filtering system that enables users to search distributed heterogeneous information sources. Profiles are used to periodically monitor specific sources for new information. Our experiment was done using the Powerize Server 1.0. Users interact with Powerize Server 1.0 through two principal interfaces: Publications and Studio. The Studio interface allows users to select and manage profiles, and the Publications interface is used to browse documents retrieved by the system.

The Studio interface includes five collections of profiles known as "wizard packs:" General, Pharmaceutical, Aerospace, Telecommunications, and Energy. Each wizard pack is designed to serve the needs of a group of users. For example, the Pharmaceutical wizard pack is intended for users in the pharmaceutical industry. The Pharmaceutical and Telecommunications wizard packs were used in our experiment. Each wizard pack consists of several "wizards," and each wizard is designed to help the user complete a particular task. For example, there is a competitive intelligence wizard to help users find information about a competitor. Each wizard is further divided into "topics," which are collections of profile templates designed to retrieve information about a particular subject. For example the competitive intelligence wizard contains topics such as " Mergers and Acquisitions" and "Financial Information." Each profile template encodes the structure of a query for a set of information sources. Users create actual profiles by selecting templates and providing search terms such as a drug or company name. By using templates, users can create powerful queries without being familiar with the individual information sources or their query interfaces.

A custom version of Powerize Server 1.0 was created for these experiments by powerize.com. It was instrumented to measure reading time and printing behavior and to record user-entered ratings for individual documents. Figure 3 illustrates the procedures for using the modified system, showing how reading time, printing behavior, and explicit ratings are recorded.

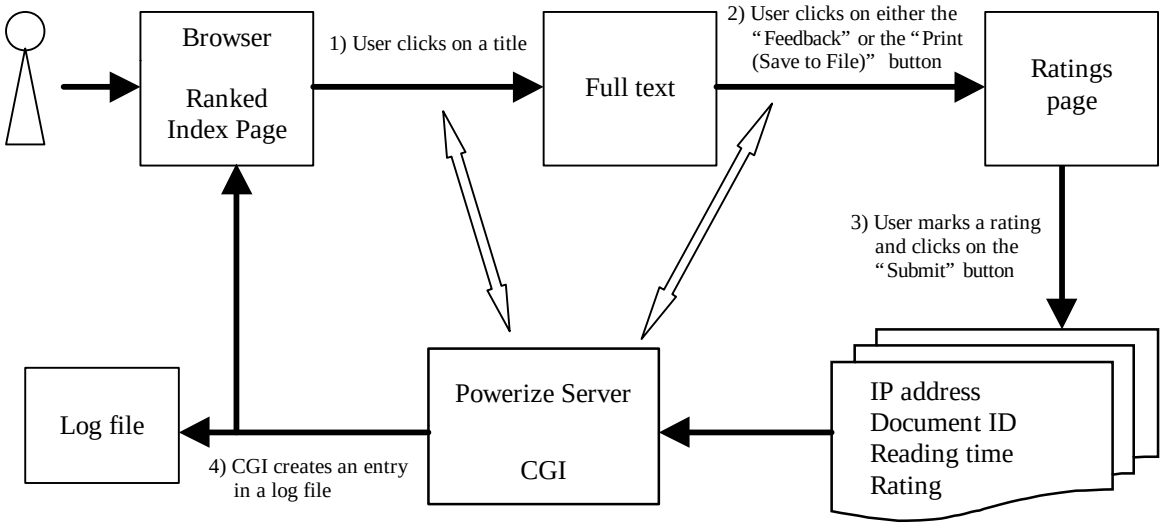


Figure 3. Procedures for using the modified Powerize Server 1.0

Powerize Server 1.0 retrieves a maximum of 20 articles for a topic, and users view titles and summaries of individual articles in a ranked list known as an "index page" in the Publications interface. In our experiment, a user examined the index page to determine which articles they wished to read. When they clicked on the title of an article for further examination, the system provided the full text of the article, records the time at which reading began, and provided "Feedback" and "Print (Save To File)" buttons at the top of the page. After reviewing the full text of the article, the user had to select either "Feedback" or "Print (Save To File)," either of which would record the time at which reading was completed and then displayed a "Ratings" page. If the user selected "Print (Save To File)," a copy of the file would be retained in a file on the server. The server was generally not located in the same room as the experimental subjects. This design allowed the desired files to be printed by an assistant during the experiment using any available printer at the experiment site. Explicit ratings were collected on the following scale: "no comment," "no interest," "low interest," "modest interest," and "high interest." When the user clicked on the "Submit" button after assigning a rating, the system made an entry in a log file that contained the IP address of the user's machine, the reading time, a unique document identifier, whether or not the user chose to print the document, and the rating assigned by the user. Clicking the "Submit" button also took the user back to the index page.

3.4 Pilot Study

A pilot study was conducted to validate the experimental procedures in November, 1998. Special consideration was given to data collection procedures in order to determine whether the system could collect and process the required information. The correctness of reading time measurements was also examined. We were able to collect all the required data, but had a problem with measuring both the reading time and user actions when a subject accidentally clicked on the "Back," "Forward," and "Print" buttons on the browser. Hiding the standard tool bar on the browser solved this problem in subsequent experiments.

The pilot study was done using only "Pharmaceutical Wizards," with 4 students who were taking a microbiology course, (MICB 443, Drug Action and Design) at the University of Maryland. A total of 21 instances of reading time and rating were gathered from the pilot study. They showed the expected pattern of increasing reading time with increasing rating. The data collected from the pilot study also suggested that printing behavior might prove useful. Every one of the 9 cases in which printing was requested was rated as relevant, and any obvious way of using reading time alone to make predictions would have missed some of those cases.

3.5 Subjects and Topic Creation

Two experiments were conducted. Eight students, junior undergraduate students taking a Gemstone honors seminar (GEMS 396, Team Project Seminar II) at the University of Maryland, participated in the first experiment using the telecommunications wizards. The students were engaged in research for a group project that required examining new products, services, and technologies for wireless Personal Communications Systems (PCS). Conversations were held with the students and their instructor several weeks before the experiment to learn about their information needs. Search topics were then created using the telecommunications wizards on Powerize Server 1.0 before the experiment. A total of 97 articles were retrieved using 5 topics: digital PCS, Iridium, Teledesic, Nextel i1000, and Ricochet. All of the selected information sources were from Dialog.

The second experiment, using pharmaceutical wizards, was done with students taking a zoology course (ZOOL 422, Mammalian Physiology) and the associated lab (ZOOL 423, Mammalian Physiology Laboratory) at the University of Maryland. The experiment was conducted during one of the regular ZOOL 423 lab sessions. There were 87 participants in the

experiment, and all were either seniors or advanced juniors at the University. An interview with the instructor was conducted several weeks before the experiment and the instructor selected search terms that were designed to provide the students with information that would be related to what they were learning in the two courses. Search topics were then created using the pharmaceutical wizards on Powerize Server 1.0 prior to the experiment. A total of 96 articles were returned using 5 topics: beta blockers, antihypertensives, ACE inhibitors, positive intropic agents, and cardiac sympathomimetics. Again, all of the selected information sources were from Dialog.

3.6 Experimental Procedures

The experiment with the Telecommunications user group took place in a single session at a computer lab on March 2, 1999. Microsoft Internet Explorer 4.0 was used. The tool bar with standard buttons on the browser was hidden by the investigator to prevent subjects from inadvertently clicking on the browser's back, forward, and print buttons, since the pilot study had revealed that clicking on those buttons caused problems with measuring both the reading time and user actions. Subjects were also asked not to make the tool bar visible and not to use those browser functions during the experiment. The investigator then provided subjects with a brief description of the study at the beginning of the experiment. A demonstration was given by the investigator to show subjects how to browse articles using Powerize Server 1.0 using their Web browser. Subjects were then asked to do a trial using a different set of articles that was retrieved only for demonstration purposes before they did the actual session using the telecommunications wizard. The experiment was completed in one hour: 15 minutes of introduction, including the demo and the trial, and 45 minutes for the actual experiment.

The experiment with the group using Pharmaceutical Wizards was done in seven sessions between March 29 and April 2, 1999 at a single computer lab on campus. Sessions 1 and 2 were administered following the same procedure that the Telecommunications user group used, except that they were done in 45 minutes instead of one hour so only 30 minutes was available for the actual experiment. There were 19 subjects in each session, and it turned out that the speed of the system was unacceptably slow, resulting in unreliable measurements of reading time. This problem had not been foreseen in the pilot study or in the first experiment because no more than 8 subjects had previously participated at one time. To minimize the impact of this problem, students were paired in groups of two for sessions 3 through 7. One student in each group was assigned to do the browsing, while the other observed the session. In this way, all of the students in each lab period were able to participate in some way, but our measurements would (hopefully) still reflect the reactions of a single student. To minimize the potential effect on reading time caused by having two subjects on a machine, students were asked not to talk to each other during the experiment.

4. Data Collection

Data were collected from the two experiments as mentioned in Section 3.6. The system gathered the following information: reading time, user actions, and explicit ratings for each article that users examined. Reading time in this study was computed based on the following formula:

Reading Time = Clock time when user clicked on a title – Clock time when user clicked on either the "Feedback" or the "Print (Save To File)" button.

Table 3 shows a sample of the data that were collected in a log file for the experiment. The table shows the reading time for the article with document ID CNT282 to be 45 seconds (10:39:34 – 10:38:49). The "F" in the User Action column indicates that the user on the machine

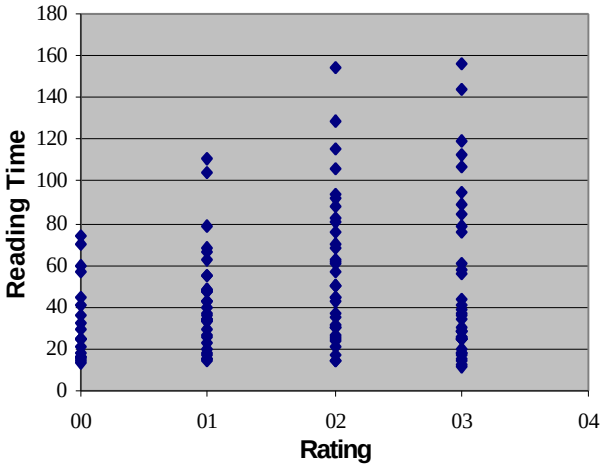
with IP address 100.2.200.101 clicked on the "Feedback" button after viewing the article identified by CNT282. A "P" in the User Action column indicates that a user clicked on the "Print (Save To File)" button, as shown for CNT284. Finally, explicit ratings were collected for every article that users selected. Ratings were recorded on a 5 point scale: "NA" for no comments, "00" for no interest, "01" for low interest, "02" for moderate interest, and "03" for high interest. In Table 3, the user with IP address 100.2.200.102 provided a rating of "03" for the article identified by CNT284.

IP Address	Time	Reading Time	User Action	Doc. ID	Rating
100.2.200.101	10:38:49			CNT282	
100.2.200.101	10:39:34	45	F	CNT282	
100.2.200.101	10:39:40			CNT282	NA
100.2.200.102	10:40:30			CNT284	
100.2.200.102	10:41:36	66	P	CNT284	
100.2.200.102	10:42:14			CNT284	03

Table 3. Examples of data collected

5. Data Analysis

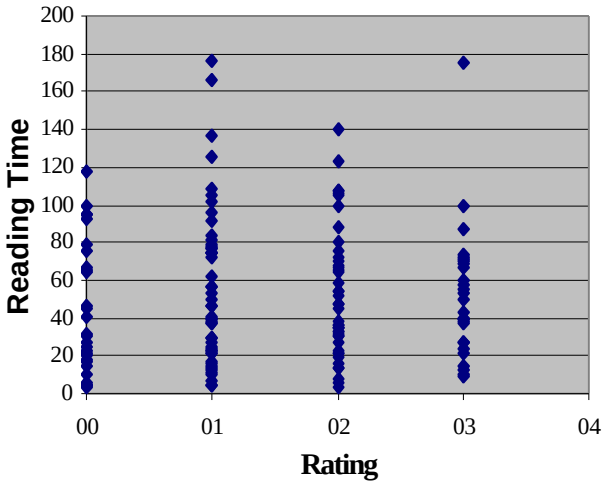
A total of 122 cases out of 130 ratings collected from the eight subjects in the first experiment (with the Telecommunications wizard) were considered valid for purposes of data analysis. All five cases collected from one subject were excluded from the data analysis because they missed the first half of the experiment. Two other cases were excluded because they were detected as outliers based on the standardized residual scores for reading time. One case was excluded because it had a rating of "no comments." Figure 4 shows the descriptive data analysis for the Telecommunications user group. An increase in reading time, in general, can be observed as the value of the rating gets higher on the scatterplot. The rating of "00," indicating "no interest," had the lowest mean reading time, and "02," representing "moderate interests," had the highest mean reading time. It seemed that subjects were able to identify highly relevant articles more quickly than those that they rated moderately relevant.



Rating	Telecommunications Wizards	
	# of Case	Mean Reading Time
00	20	32.85
01	31	42.84
02	34	57.74
03	37	50.24
Total	122	47.60

Figure 4. Descriptive data analysis for the Telecommunications user group

In the second experiment (the Pharmaceutical user group) there were 7 sessions. In sessions 1 and 2, 36 subjects provided 166 ratings, but data from those two sessions were not used in this study because of the slow system response time described in Section 3.6. A total of 532 ratings were gathered from 49 subjects that participated in sessions 3, 4, 5, 6, and 7. A total of 153 cases out of the 532 ratings gathered were considered as valid for data analysis in this study, in part because it was discovered after the experiments that only 25 of the 96 articles presented to each subject had abstracts (none had full text). The 363 ratings that were given for the 71 articles that lacked abstracts were excluded from the data analysis because we did not feel that the bibliographic citations alone could provide an adequate basis for assessment by the users. Three cases that were detected as outliers and 13 cases with "no comments" were also excluded from the data analysis. The scatterplot in Figure 5 presents the distribution of 153 valid cases, and the associated table shows both the number of cases and the mean reading time for each rating.



Rating	Pharmaceutical Wizards	
	# of Case	Mean Reading Time
00	29	42.97
01	51	55.14
02	45	52.00
03	28	51.50
Total	153	51.25

Figure 5. Descriptive data analysis for the Pharmaceutical user group

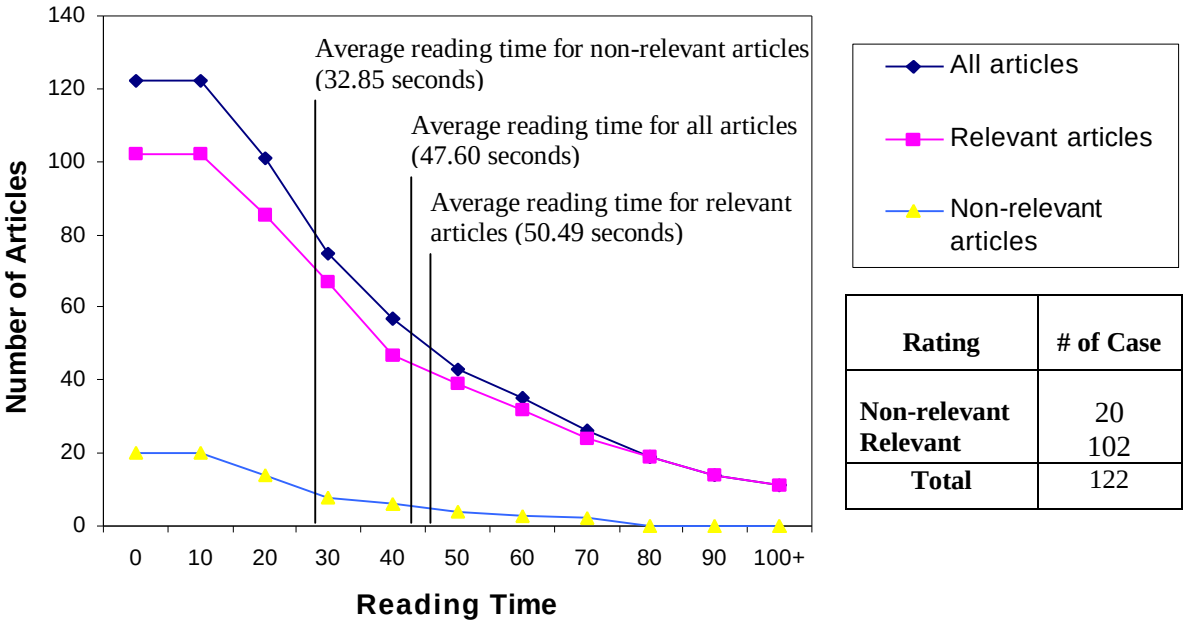
5.1 Reading Time as a Source for Implicit Feedback

In both experiments, we noted a decline in mean reading time between articles rated as moderate interest and those rated as high interest. In fact, a consistent decline in reading time in the second experiment was evident as interest increased. This suggests that we will likely not be able to reliably distinguish between degrees of interest using reading time, so we converted the ratings to a binary scale: "00" to "non-relevant" and "01, 02 and 03" to "relevant" for our subsequent analysis in both experiments.

Figure 6 presents the descriptive data analysis on reading time with this binary rating scale for data collected from the experiment with the Telecommunications user group. An increase in mean reading time was observed from non-relevant to relevant documents on the graph. Ratings made on non-relevant documents and on relevant documents were normally distributed below and above the mean reading times of 32.85 and 50.49 seconds, respectively.

An Independent-Samples t-test, comparing the mean reading time on relevant documents with non-relevant ones, was done to test our first hypothesis. A statistically significant difference between the two mean reading times was found at $\alpha = .05$. We therefore conclude that users tend to spend a longer time reading relevant articles than non-relevant articles, which is a consistent result with the two previous studies by Morita and Shinoda (1994) and by Konstan et al. (1997). Morita and Shinoda, in their study in 1994, concluded that preference of a user for an article was the dominating factor that affected time spent reading it, and they suggested using a threshold on reading time to detect relevant articles. Their results showed that 30 % of interesting articles

could be retrieved with precision of 70 % by using a threshold of 20 seconds. A much higher threshold would be required in our first experiment to reach a similar recall level. This comports with our intuition, since Morita and Shinoda used Usenet news articles, while our study was conducted with academic and professional journal articles. Several factors, such as the length of the article, levels of difficulty for understanding the contents, and differences in language skills, could affect the reading time. Subjects in our study, for example, might require longer reading time to understand the content of an article because none of them were experts in the field. Figure 7 shows the recall and precision for different ranges of reading time. For example, the recall and precision that would result from treating articles with reading time of at least 40 seconds as relevant were 0.418 and 0.894, respectively.



Rating	# of Case
Non-relevant	20
Relevant	102
Total	122

Figure 6. Number of articles on different reading time (Telecommunications user group)

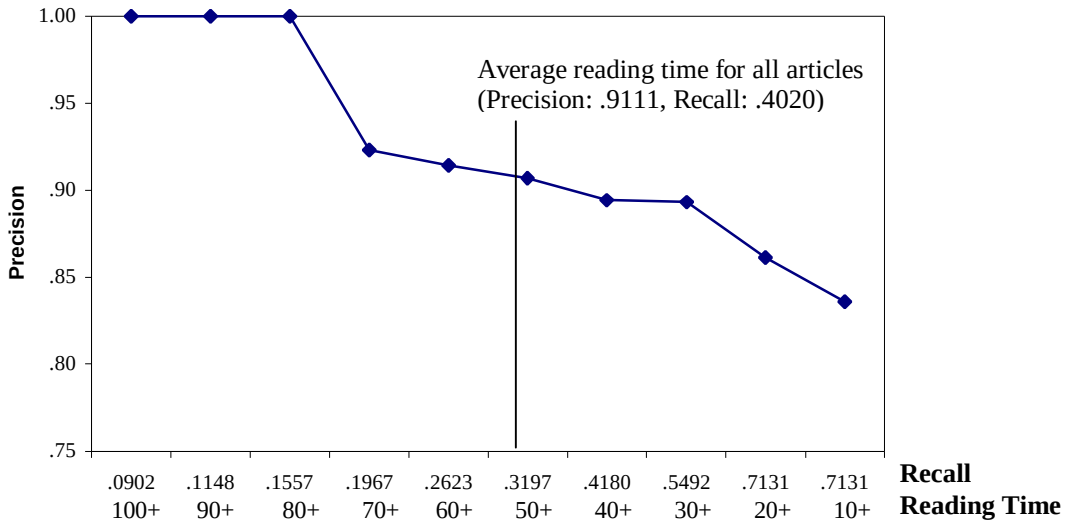
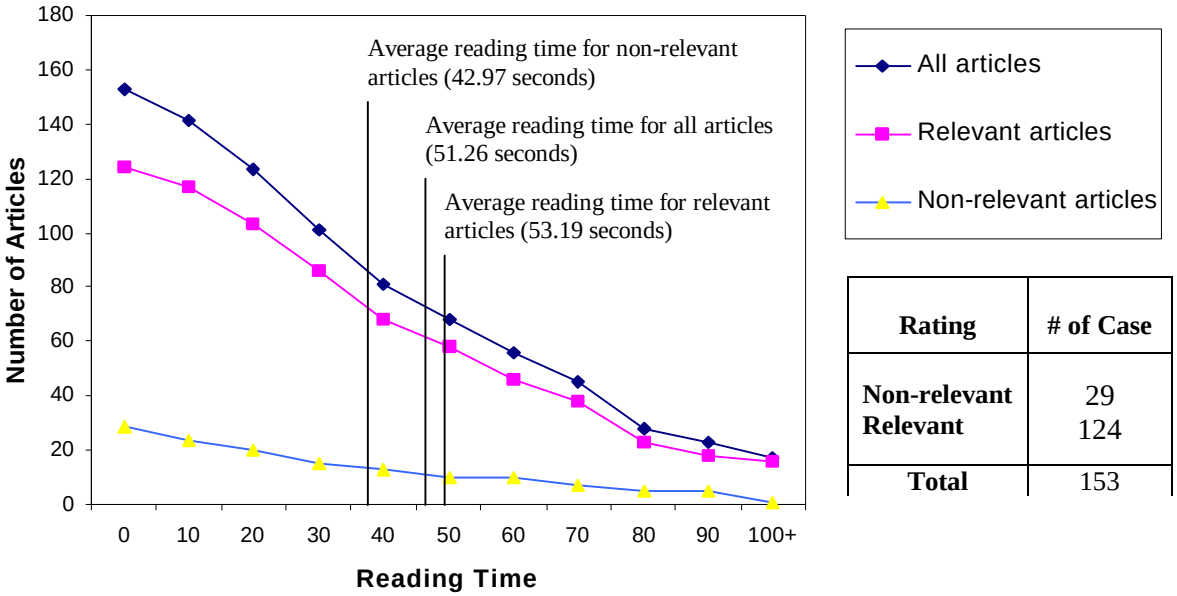


Figure 7. Recall and precision on different reading times (Telecommunications user group)

Figure 8 shows the descriptive data analysis for our experiment with the Pharmaceutical user group. There was a 10.22 second difference between the mean reading times on relevant and non-relevant documents, but no statistical significance was found at $\alpha = .05$, based on the Independent-Samples t-test. The mean reading time on relevant documents was 53.19 seconds, which was close to the one (50.49 seconds) for the Telecommunications user group in our first experiment. The mean reading time on non-relevant documents, however, was 42.97 seconds, which was 10.12 seconds more than was observed with the Telecommunications user group. We suspect that this unexpected outcome resulted at least in part from the different setting in which we paired two students together. As we mentioned in Section 3.6, one student in each group was observing the session, while the other was browsing retrieved articles. In this case, the student doing the browsing might have sometimes chosen to wait until the other student had also examined the article before clicking on the feedback button.



Rating	# of Case
Non-relevant	29
Relevant	124
Total	153

Figure 8. Number of articles on different reading time (Pharmaceutical user group)

Although the difference between the mean reading times was not found to be statistically significant, the overall pattern is similar to that which we observed with the Telecommunications user group. The increase in mean reading time between non-relevant and relevant articles is shown on the graph in Figure 8. A total of 56 of the 124 articles reported as relevant had a reading time of more than 51.26 seconds, which was the mean reading time for all documents, while 10 out of 29 non-relevant articles were found in that range. Figure 9 presents the observed recall and precision for different ranges of reading time. For example, the recall and precision that would be achieved if documents with the reading time of at least 40 seconds were considered relevant was 0.444 and 0.83, respectively.

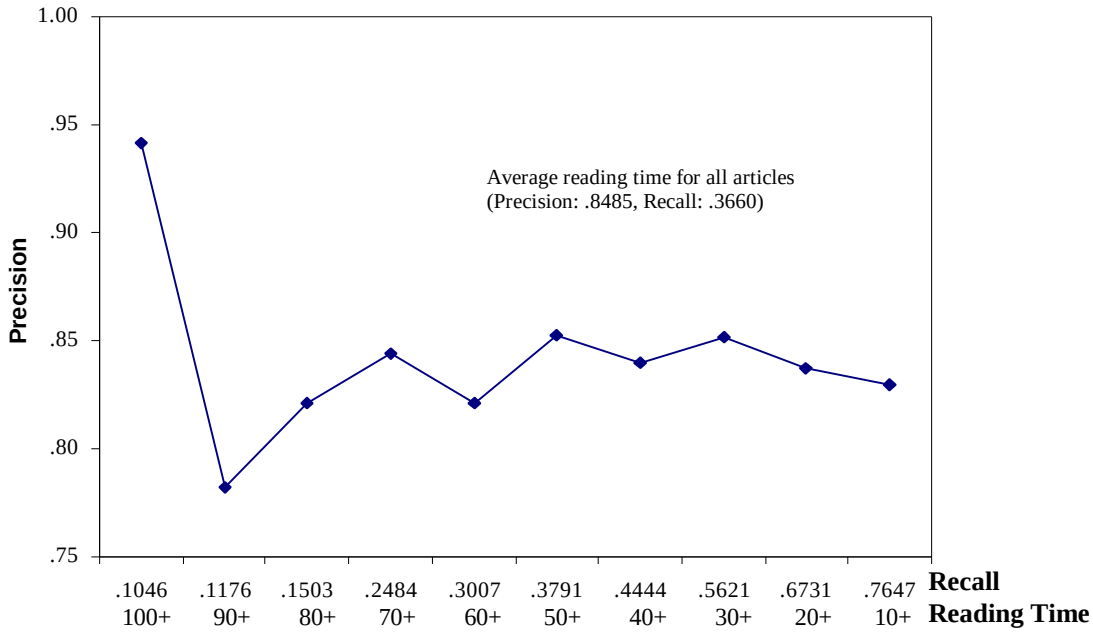


Figure 9. Recall and precision on different reading time (Pharmaceutical user group)

5.2 Printing Behavior as Evidence of Interest

Printing behavior was examined in this study with the hope that it may provide us with clues that can predict explicit ratings beyond those clues given by reading time. There were a number of relevant documents that could not be discriminated from non-relevant ones using only reading time in Figures 6 and 8. For example, using 47.60 and 51.26 seconds as thresholds for cutting off non-relevant documents in Figures 6 and 8 will also throw 61 out of 102 (59.80 %) and 68 out of 124 (54.84 %) relevant documents away, respectively. Can printing behavior provide a clue for detecting those relevant documents that would have been thrown away using reading time alone?

Unfortunately, only two cases of printing behavior were available from the data collected from the experiment with the Telecommunications user group, as shown in Table 4. No meaningful interpretation on the data collected could be made with only two cases. We believe that the low frequency of the printing behavior was because subjects in the experiment did not have a practical information need. All subjects in the experiment with the Telecommunications user group were doing a group project that required writing a term paper for their class. They, however, performed their own information searches by the time they attended our experiment, which may have reduced their desire to have information on the topic.

There were 16 cases of printing behavior for the experiment with the Pharmaceutical user group. Although no statistical significance was found between the mean reading times for relevant and non-relevant documents with this user group, an increase in reading time from non-relevant to relevant documents was observed that could be used as a source for predicting explicit ratings. Using the reading time alone as the source for implicit feedback, however, could not detect those relevant documents that fell under the threshold reading time. Our second goal was to examine how many more relevant documents could be detected by using the printing behavior than using reading time alone.

In Table 4, the mean reading time for 16 cases with printing behavior was 45.25 seconds, which was 2.28 seconds more than the mean reading time for non-relevant documents (42.97 sec.), but 6.01 seconds less than the one for all articles (51.26 sec.). In many cases, articles that

were printed were highly relevant, and users seemed to discriminate them quickly from non-relevant ones, which resulted in reducing the reading time. Printing behavior thus provides a useful clue for predicting explicit ratings over reading time, in that it can detect relevant documents below an established threshold of reading time. As in the pilot study, every printed document was judged to be relevant, and 10 out of 16 printed documents had a reading time of less than the mean reading time for all documents (51.26 seconds). Using printing behavior could identify those 8 relevant documents with short reading times.

Case	Telecommunications Wizards		Pharmaceutical Wizards	
	Reading Time	**Rating**	**Reading Time**	**Rating**
1	156	03	100	03
2	81	02	58	03
3			53	03
4			43	03
5			38	03
6			12	03
7			100	02
8			67	02
9			66	02
10			48	02
11			36	02
12			35	02
13			32	02
14			8	02
15			17	01
16			11	01
Mean	118.50		45.25	

Table 4. Case summaries for Print/Save behavior

6. Implications for the Powerize Server

The Powerize Server could exploit user modeling to perform two functions: source selection and document selection. In this section we describe how the results of these experiments could be used as a basis for document selection. The key idea is to use observable behavior (selection, examination and retention, in this case) in conjunction with computationally tractable representations for the associated documents as a basis for machine learning. Source characterization is presently an open research question, so improving the performance of the source selection model would present challenges that we are not presently prepared to address.

In supervised machine learning, the algorithm is trained by presenting a sequence of training instances that represent items of interest, and each training instance is associated with an example of the appropriate response to that item. When working with text, the most common representation for the training instances is a vector of weights, with one weight for each important term in the collection. With only positive training instances, the most important terms are generally taken to be those that are relatively rare, and hence highly selective. When negative training instances are also available, the important terms are generally taken to be those that best discriminate between the positive and negative training examples. Generally, a weight is

assigned to each term, with terms that don't appear in a particular document being assigned a weight of zero. The desired response is generally a number (typically either binary-valued or real-valued and normalized to be between zero and one) that indicates the desirability of the associated document. Once training is complete, the machine learning algorithm is supplied with instances for which the appropriate response is not known and predicted responses are generated. Real-valued estimates of desirability can be used to rank a set of documents, while binary-valued estimates are well suited for use in systems that group documents in other ways. Either approach is easily supported, although some machine learning techniques are better suited to one and some are better suited to the other. Straightforward variants of this general paradigm can accommodate initial information (such as the profiles generated by the Powerize Server's present wizard packs) and can interleave training instances with instances for which predictions are required.

Given the nature of our results, assembling suitable training instances is fairly straightforward. For each of our experiments, there was a reading time beyond which the document was assured to be of interest (77 seconds and 120 seconds respectively), and printing also provided reliable evidence of interest. Combining the two sources of evidence would produce 100% precision with about 15% recall in each case. We defer for the moment the question of how an appropriate threshold on reading time might be discovered. Examination of Figures 4 and 5 makes it clear that no similar strategy could reliably detect undesirable documents from among those that the user has selected for examination. If experience shows that the density of desirable articles among those that were presented to the user but not selected for examination is relatively low, we could choose a random sample of the highly ranked but unexamined documents as undesirable documents. Otherwise, it would be safer to choose documents below the lowest-ranked examined document as being representative of the set of undesirable documents. Such a set of desirable and undesirable documents would provide a particularly useful basis for training because the documents are exactly those that the system is otherwise unable to distinguish. By the time the user has examined 100 documents (a few hours work), we can expect to have a set of (for example) 15 positive training examples and 15 negative training examples.

Oard (1997) identified six machine learning techniques that have been used for document selection in information filtering applications: rule induction, instance based learning, statistical classification, regression, neural networks and genetic algorithms. Stevens (1993) observed that rule induction is an attractive choice for interactive applications because the compact set of rules that results could be presented to the user. In general, other machine learning techniques produce fairly opaque representations that fail to leverage the user's potential participation in the process. Rule induction produces only binary-valued results, however, so integration of this technique into a ranked retrieval system does pose some challenges. A simple approach to rule induction would be to search the space of disjunctive normal forms (disjunctions of conjunctions) over terms for an expression that balances predictive accuracy on the training set with a preference for short Boolean formulae (to avoid overfitting to the training set). Several complete systems implementing more sophisticated and efficient approaches to performing rule induction on document vectors have been developed. One of the most widely used is the RIPPER system from AT&T research.[1]

Although rule induction can be performed without the user's direct involvement, allowing the user to accept, modify or reject the rules could provide more rapid convergence on a good rule set. Furthermore, the utility of the training instances can be indirectly inferred from the user's acceptance or rejection of proposed rules. For example, rule rejection would provide evidence that the reading time threshold should be increased, while acceptance of a proposed rule would provide evidence that the threshold is at a safe value and that it might be possible to lower

[1] Information about RIPPER is available at http://www.research.att.com/~wcohen/ripperd.html.

it if more positive training instances are desired. Rule induction is generally robust in the face of a few inappropriately labeled training instances, so in practice the optimal reading time threshold may actually be below the perfect-classification thresholds described above.

Implementation of rule induction based on implicit feedback within the Powerize Server should be relatively straightforward. Powerize Server already caches the full text of the documents, so vector representations are easily constructed. Selection behavior and reading time are easily observed when users interact with documents stored on the Powerize Server, although the inability to observe scrolling behavior (which would require browser modifications) may add some noise to the observations because a relatively brief shift in attention to another task might be confused with a long (but reasonable) reading time. Retention behavior can be observed by implementing server-side personal document storage and printing functions. Positive examples could be obtained by collecting a reading time distribution over the course of a few days, rejecting outliers (which would likely result from distraction of situated users by other tasks), and then adopting the 90^{th} percentile of the remaining reading times as a fairly conservative threshold. Negative training examples could be assembled from highly scored articles that the user failed to examine. The most important terms could then be calculated using a chi-squared measure to determine which terms distinguish best between the two sets. Feature vectors constructed using these terms could then be presented to a program such as RIPPER, with the resulting rule set used to identify the most promising documents from among those found by the Powerize Server. The results could be displayed in a number of ways, but one straightforward technique would be to list the results of such "Personal Powerization" in a separate window, using the original rank ordering as a basis for sorting those documents.

Although this broad implementation strategy is clear, some interesting design issues remain to be resolved. Stevens (1993) implemented one approach for proposing new rules to users, but the user interface design space for this task is rather large and generally unexplored. Similarly, it might be necessary to try a few threshold adaptation strategies in order to strike a suitable balance between responsive learning and overcorrecting. Finally, an unobtrusive explicit feedback function might also prove useful as a way of allowing users to help customize their system more quickly than implicit feedback alone would permit, and inclusion of such a capability would further leverage the investment in the incorporation of machine learning technology. The design outlined above can easily incorporate explicit feedback as an additional source of evidence, but we are not aware of any prior work on the unification of implicit and explicit feedback, so there may be some interesting lessons to be learned along the way. In summary, incorporation of implicit feedback into the Powerize Server would entail a modest development effort, and some field testing of alternative approaches would likely be necessary during the development process.

7. Conclusion

We have shown that documents selected by the user from the Powerize Server have a good chance (typically 85% or better) of being of interest, that reading time provides additional evidence about the user's interest, and that retention behavior (printing, in our experiments) provides still further evidence of interest. These results have important practical implications for the development of personalized filtering systems, and we have illustrated how this information can be used to adapt a filter to an individual user's preferences. Because implicit feedback can be collected ubiquitously, vastly more evidence about user interests can be collected in this way than would likely be obtained through reliance on explicit feedback alone. Using this evidence, it thus becomes possible to realize the vision of building truly personalized information systems that seamlessly adapt as their users' interests change.

Acknowledgements

The authors wish to thank Nick Carmello of powerize.com for modifying Powerize Server 1.0, Professors William Higgins and Carol Pontzer at the University of Maryland for working closely with us to find subjects for our experiments and to craft meaningful tasks for them to perform, and our volunteer participants, without whom our research would not have been possible. This work has been supported in part by the Maryland Industrial Partnerships program and powerize.com.

References

Brin, S. and Page, L. (1998) The anatomy of a large-scale hypertextual Web search engine. Dept. of Computer Science, Stanford Univ.

Garfield, E. (1979*) Citation indexing: Its theory and application in science, technology, and humanities*. New York: Wiley-Interscience.

Goldberg, D., Nichols, D., Oki, B. M, and Terry, D. (1992) Using collaborative filtering to weave an information Tapestry. *Communication of the ACM*, December, 35(12): 61-70.

Hill, W.C., Hollan, J. D., Wrobelwski, D. and McCandless, T. (1992) Read wear and edit wear. In *Proceedings of ACM Conference on Human Factors in Computing Systems*, CHI '92: 3-9.

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Riedl, J. (1997) GroupLens: Applying collaborative filtering to Usenet News. *Communication of the ACM*, March, 40(3), 77-87.

Korfhage, R. R. (1997) *Information Storage and Retrieval*. John Wiley & Sons, Inc., New York; NY.

Morita, M and Shinoda, Y. (1994) Information filtering based on user behavior analysis and best match text retrieval. *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pp. 272-281.

Nichols, D. M. (1997) Implicit ratings and filtering. In *Proceedings of the 5th DELOS Workshop on Filtering and Collaborative Filtering*, Budapest, Hungary 10-12, ERCIM. http://www.ercim.org/publication/ws-proceedings/DELOS5/index.html

Oard, D. W. (1997) The state of the art in text filtering. *User Modeling and User-Adapted Interaction*, 7(3), 141-178..

Oard, D.W., and Kim, J. (1998) Implicit Feedback for Recommender System. In *AAAI Workshop on Recommender Systems*, Madison, WI: 81-83. http://www.glue.umd.edu/~oard/research.html

Rucker, J. and Polanco, M. J. (1997) Personalized Navigation for the Web. *Communications of the ACM*, March, 40(3): 73-89.

Sheth, B. D. (1994) "A learning approach to personalized information filtering." Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science. http://lcs.www.media.mit.edu/groups/agents/publications/

Stevens, C. (1993) *Knowledge-based assistance for accessing large, poorly structured information spaces*. Ph.D. thesis, University of Colorado, Department of Computer Science, Boulder. http://www.holodeck.com/curt/mypapers.html

Turtle, H. And Croft, W.B. (1990) Inference networks for document retrieval. In J.-L. Vidick (ed.) : *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, pp.1-24.

Yan, T.W. and Garcia-Molina, H. (1995) SIFT – A too for wide-area information dissemination. In *Proceedings of the 1995 USENIX Technical Conference*, pp.177-186. ftp://db.stanford.edu/pub/yan/1994/sift.ps