

Information Foraging Models of Browsers for Very Large Document Spaces

Peter Pirolli and Stuart K. Card

Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304, USA
pirolli@parc.xerox.com
card@parc.xerox.com

ABSTRACT

Information Foraging (IF) Theory addresses user strategies and technology for seeking, gathering, and using on-line information. We present IF-based models and evaluations of two interfaces: the Scatter/Gather browser for large document collections, and the Butterfly interface for surfing the citation link structure of scientific literatures. A computational cognitive model, ACT-IF, models observed users by assuming that they have heuristics that optimize their information foraging behavior in accordance with IF theory.

Keywords: Information foraging theory, cognitive models, information retrieval.

INTRODUCTION

We have been developing Information Foraging (IF) Theory [15, 16] as an approach to understanding user strategies and computer technologies for information seeking, gathering, and use. The basic idea behind IF theory is that we can analyze how users and their technology are adapted to the flux of tasks and information their environment. To do this we have merged optimal foraging theory in biology [18] with theories of human cognition [1, 2]. Our heuristic assumption is that human-computer interaction is adaptive to the extent that it maximizes the valuable knowledge gained from using a system in relation to the cost of interaction with that system. This means we need to understand things like (a) the value of information, (b) the costs of interaction, and (c) the ways in which interaction can be optimized. Here we present IF analyses of two different browser systems to illustrate models of potential relevance to research on advanced visual interfaces that connect users to very large collections of text documents.

The first set of analyses concerns the Butterfly browser [12] which provides users with ways of navigating through research literatures along citation and reference links. Because the Butterfly is a fast graphical interface connected through a slow back-end (the Internet) to this

hyperlinked scientific literature, it uses multi-threaded processes to speed up human-computer interaction. That is, rather than performing one interface action after another (i.e., processing actions in single-threaded serial manner), the user may process several actions in parallel. An IF analysis is used to understand quantitatively the effects of switching from a single-threaded to a multi-threaded interface. A second set of analyses is presented for the Scatter/Gather browser [9, 17] for very large full-text document collections. Users of Scatter/Gather face many of the same problems encountered in browsing any large collection of documents, such as the World Wide Web (WWW). Our analyses concentrate on understanding how well an interface may summarize the semantic content of documents using small amounts of display space (in this case, by using small text summaries) and how users navigate in ways that optimize their rate of gaining relevant information.

Across the two example systems (Butterfly and Scatter/Gather), we develop four quantitative models of the rate, R , of gaining valuable information. Adaptive strategies are discussed in the context of these models. The models include: an *Information Diet* model that relates R to the choice of information sources to pursue and use, and three variations of an *Information Patch* model that relates R to the time spent getting to relevant sources, the density of relevant information within sources, and time spent foraging through the sources.

These four *adaptation-level* models are engineering-style models. The models assume that people will behave in bounded rational ways to optimize their interactions with their technology in order to achieve their goals and interests. Such models need to be grounded in explanations of users' cognition. As an illustration of such grounding, we present a cognitive computational model, called ACT-IF, that explains how cognitive mechanisms work to optimize user behavior. ACT-IF includes mechanisms for judging the relevance of text information displayed on a screen. These cognitive mechanisms assess relevance by computing the

likelihood that words on the screen are relevant to an information need. ACT-IF also includes mechanisms that maximize the rate at which relevant information will be encountered and used. We show how ACT-IF is consistent with activity logs of users at a very fine grain of analysis. These results provide strong evidence in support of ACT-IF and, more generally, for Information Foraging theory. We also show how ACT-IF can be used to assess the design of a browser for very large collections of documents.

SINGLE-THREADED AND MULTI-THREADED PROCESSING IN THE BUTTERFLY CITATION BROWSER

The Butterfly Citation Browser is a program for visualizing part of citation databases, specifically the Science Citation Index, the Social Science Citation Index, and the IEEE INSPEC. Figure 1 (see Color Plate 1) shows a view of the browser. The large butterfly-like structure in the foreground represents an article. The left side of the object is a list of earlier articles that the article cites. The right side of the article is a list of later articles which cite it. Clicking on any of the cited or citing articles causes the object to move off to the left and another object to appear in its place, representing a new article to which the user has moved. In this way, users can move among the citation links in the citation network. The actual database is on Dialog. The system does searches and browses links by issuing requests to Dialog over the Internet.

The citation indexes can also be searched by keyword. The numbered boxes near the top of the screen represent articles obtained as the results of such a search. Clicking on one of these brings it to the foreground position, making it look like in Figure 1 (Color Plate 1). From this point, it is possible to browse citation links forward and backward as described above.

This is a fast interface connected to a slow back-end (the Internet). If a user's requests and actions are processed in a single-threaded serial manner, then the user must wait during delays that occur because of slow Internet processes. The multi-threaded processing of the Butterfly interface allows the user to work with articles that have already arrived while it performs pre-fetching in the background. This is like many systems that allow people to perform information handling tasks while items are being retrieved and placed on some sort of queue for processing at their discretion. Information filtering systems [6], typically adopt such an approach.

Adaptation-level Analysis: Information Patch Models

We present a simple IF analysis of the Butterfly interface that illustrates the quantitative benefit of moving from single-threaded processing to multi-

threaded processing. This is also intended to introduce a class of models called Information Patch models. These were adopted from optimal foraging theory [7, 18] where they are called Patch Models and are used to analyze the food-foraging strategies of animals.

In optimal foraging theory, the models concern situations in which the environment of some particular animal has a "patchy" structure. For instance, imagine a bird that forages for berries found in patches on berry bushes. The forager must expend some amount of between-patch search time getting to the next food patch. Once in a patch, the forager faces the decision of continuing to forage in the patch or leaving to seek a new one. Frequently, as the animal forages within a patch, the amount of food diminishes or depletes. For instance, our imaginary bird would deplete the berries on a bush as it ate them. In such cases there will be a point at which the expected future gains from foraging within a current patch of food diminish to the point that they are less than the expected gains that could be made by leaving the patch and searching for a new one. Quantitative Patch models in optimal foraging theory determine the optimal policies for allocating time to foraging within a food patch vs searching for new patches.

By analogy, the task environment of a user often has a "patchy" structure. Information relevant to a user's information needs may reside in piles of documents, file drawers, office book shelves, libraries, or in various on-line collections. Often the user has to navigate from one information patch to another—perhaps from one pile to another, or from one on-line collection to another. Often the user is faced with a decision much like our imaginary bird: is it better to continue foraging through the current patch or is it better to seek out another patch? Information Patch models provide an analysis of this problem.

We first consider an Information Patch model where retrieving information and using it are mutually exclusive activities (single-threaded processing). We then consider a model when they are not mutually exclusive (multi-threaded processing). Later, in the context of the Scatter/Gather browser we will present a third variation of an Information Patch model.

Assume that a user is doing their best to maximize the rate of gain, R , of valuable information from information patches in their environment. For concreteness, assume that these information patches are either relatively static on-line collections such as WWW sites, or temporary collections constructed by a WWW search engine in response to user queries. For a particular user with an information need, imagine that there is a gain function $g(t)$ that tells us how much valuable information will be gained by foraging through a collection for an amount of time t . Imagine that moving from one collection to

* Published, 1998, in the *Proceedings of the Working Conference on Advanced Visual Interfaces, AVI '98* (pp. 83-93), L'Aquila, Italy. New York: ACM Press.

another takes s amount of time on average. For instance, this might be the time it takes to surf from one WWW collection to another, or the time it takes to generate a new query and retrieve results from a search engine. If we assume generally that the rate of getting from one patch to another is Poisson distributed, then the rate R of gaining valuable information can be characterized as:

$$R = \frac{\Lambda g(t)}{1 + \Lambda t} \quad (1)$$

where $1/\Lambda = s$ (i.e., Λ is a Poisson parameter that determines the mean rate of moving from one collection to another). Equation 1 is optimized by finding the optimal value of t , $t = t^*$, that maximizes R , and this can often be done by simple calculus [18].

The gain function $g(t)$ assumes some assessment of information value. Assessing the value of information can be a complex issue that ultimately depends on the task and its potential outcomes. For both the Butterfly Browser and the Scatter/Gather Browser, we use a very simple assessment involving judgments of the relevance of documents to particular queries.

Equation 1 assumes that moving from one patch to another is mutually exclusive of foraging within a patch. We can elaborate Equation 1 with simple results from queuing theory [8]. Assume that while a user forages in an information patch, that other information patches are retrieved and placed in a queue.¹ Assume that while the user processes one information patch, other overlapping items arrive on the queue as a Poisson process at rate $\hat{\Lambda}$. Assuming that $\hat{\Lambda} < 1$, then the rate of gain, R , is [13, 18]:

$$R = \frac{\Lambda g(t)}{1 + (\Lambda - \hat{\Lambda})t} \quad (2)$$

Note that when $\hat{\Lambda} = 0$, Equation 2 (multi-threaded with queued items) becomes Equation 1 (single-threaded with zero queued items).

Application to the Butterfly Citation Browser

How do we define an information patch for the Butterfly system? There are actually two kinds of patches that are related hierarchically to one another. For each keyword search there will be a temporary collection retrieved by a search engine, and this constitutes one level of information patch for the Butterfly Browser. This is

¹The analogous situation in optimal foraging theory is that of web-building spiders who can process (eat) one insect while others are “queued up” by the spiders’ web [13].

* Published, 1998, in the *Proceedings of the Working Conference on Advanced Visual Interfaces, AVI '98* (pp. 83-93), L’Aquila, Italy. New York: ACM Press.

much like the temporary collections created by search engines on the WWW. Then, for each individual article there are chains of citations that can be browsed. Each article constitutes a subpatch of citation links that can be followed. This is much like the hyperlink collections found at WWW sites.

Because the search engine for Butterfly returns articles in an unordered manner, the gain function $g(t)$, can be approximated as a linear function

$$g(t) = \lambda t, \quad (3)$$

where λ is the Poisson rate parameter for returning articles.

To illustrate how a multi-threaded dialogue improves the operation of the Butterfly, we try searching for the keyword term “RSVP” and select the mode of the program that causes it to keep acquiring new articles over the Internet from DIALOG while the user processes articles that have already arrived. A user examines these articles for relevance. We then run the program again, this time not allowing new articles to be received in a manner to overlap user processing. The data are:

<u>Overlapped Encounters</u>	<u>Non-Overlapped Encounters</u>
$\Lambda = 26$ articles/260 sec	$\Lambda = 22$ articles/432 sec
$= 0.096$ articles/sec	$= 0.051$ articles/sec
$\lambda = 6$ hits/270 sec	$\lambda = 4$ hits/432 sec
$= 0.022$ hits/sec	$= 0.0093$ hits/sec
$\hat{\Lambda} = 20$ articles/270 sec	$\hat{\Lambda} = 0$ articles/sec overlap
$= 0.074$ articles/sec	
overlap	

The rate of gain, R , as a function of time spent foraging in information patches t is plotted in Figure 2a. As expected, the rate of gain for the overlapped, multi-threaded dialogue is much greater. Also note that Figure 2a suggests that users should stay in information patches as long as possible (maximize t). This is because we assumed in Equation 3 that users get linear gains for foraging in a patch. Of course, this is often not true. It is reasonable to assume that we could produce a system that has some amount of ordering to the return list. For the Butterfly Browser, the search engines (e.g., DIALOG) order articles by recency of publication, and we might assume that this has a high correlation with relevance judgments.

Let us assume, therefore, that instead of a linear gain function, such as Equation 3, that the gain function is of the diminishing returns form

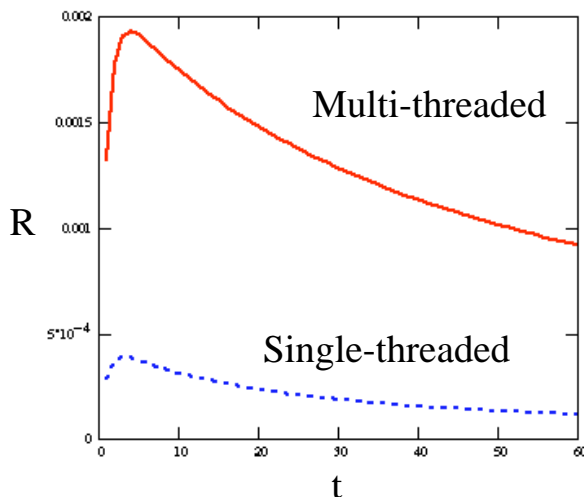
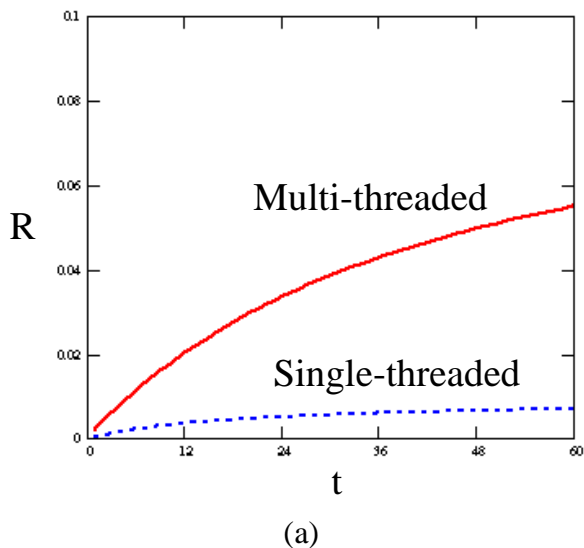
$$g(t) = \lambda(1 - e^{-kt}) \quad (4)$$

We then compute R as a function of within-patch foraging time t , using this new gain function to get the curves in Figure 2b. In contrast to Figure 2a, we now have curves that have clear optima. Note that:

- The optimum for the multi-threaded Butterfly is much larger than for the single-threaded Butterfly.
- That the optimum for the multi-threaded system occurs at a slightly higher value of t than for the single-threaded system. This means that an optimizing user will spend less time foraging within information patches using the single-threaded browser than with the multi-threaded browser.

This last point illustrates how the following general principle is implied by Equation 2 (under the assumption of a depleting returns function $g(t)$):

- The addition of queuing ($\hat{\Lambda}$) in a multi-threaded interface means that users can increase their gains by spending more time foraging within an information patch.



(b)

Figure 2. Rate of gain for the Butterfly citation browser. R is number of relevant items/sec. Time t is in secs. See text for details.

BROWSING WITH SCATTER/GATHER

Our next analysis is aimed at showing (a) that the Scatter/Gather interface does an excellent job of accurately communicating the contents of up to hundreds of thousands of documents, with only a few lines of screen text, and (b) that the browsing behavior of users, which on the surface appears chaotic, is well-modeled by assuming that they are optimizing their information foraging behavior.

The Scatter/Gather Browser

The Scatter/Gather system [9] uses the clustering of documents as the basis of a browser suitable for large numbers of documents. Figure 3 presents a typical view of the Scatter/Gather interface.² The document clusters are separate areas on the screen. The user may *gather* those clusters of interest by pointing and selecting buttons above each cluster. On command, the system will select the subset of documents in these clusters, then automatically *scatter* that subcollection into another set of clusters. With each successive iteration of scattering and gathering clusters, the clusters become smaller, eventually bottoming out at the level of individual documents. Internally, the system works by precomputing a *cluster hierarchy*, recombining precomputed components as necessary. This technique allows the interactive reclustering of large document collections in reasonable times.

The clustering in Scatter/Gather depends on a measure of inter-document similarity [19]. The method summarizes document clusters by *meta-documents* containing profiles of topical words and the most typical titles. These topical words and typical titles are also used to present users a summary of the documents in a cluster. Topical words are those that occur most frequently in a cluster, and typical titles are those with the highest similarity to a centroid of the cluster. Together, the topical words and typical titles form a *cluster digest*.

Examples of these cluster digests are presented in each subwindow in Figure 3. The idea is that the user looks at these words and assesses their relevance to their particular information needs. Below, we will use the ACT-IF model to assess how well these cluster digests

²This interface was developed by Marti Hearst at Xerox PARC.

work in communicating the relevance of clusters of documents.

In studies [16, 17], Scatter/Gather was applied to the 2.2 gigabyte TIPSTER text collection created for the TREC text retrieval conference [10]. This test corpus contained 742,833 full-text documents collected from the Wall Street Journal, the Associated Press newswire, Department of Energy technical abstracts, the Federal Register, and computer articles published by Ziff-Davis. The corpus has been extensively used by the information retrieval community. Standard information retrieval tasks (queries) have been defined on it together with lists of known relevant and non-relevant Tipster documents, as judged by experts. The test corpus provides us with a common standard against which to compare performance. The retrieval tasks involve finding as many relevant documents within a time limit as possible.

Adaptation Level Analysis

We first outline the two adaptation-level models that have guided the development of our cognitive-level ACT-IF model. The first is an Information Diet model in which one may abstractly think of the user as an “information predator” whose aim it is to select “information prey” (relevant documents and document collections) so as to maximize the rate of gain of information relevant to their task [16]. The second adaptation-level model is another variation on the Information Patch models introduced above.

Optimal Information Diet Algorithm

Users of Scatter/Gather spend much of their time evaluating clusters of documents to decide if they should pursue them (by either gathering them into a new subcollection or displaying them). The Information Diet model [15, 16] suggests that the optimal strategy would be for users to somehow rank the *profitabilities* of clusters. These profitabilities depend on the number of relevant documents that the user thinks will be found in each cluster divided by the amount of time it would take

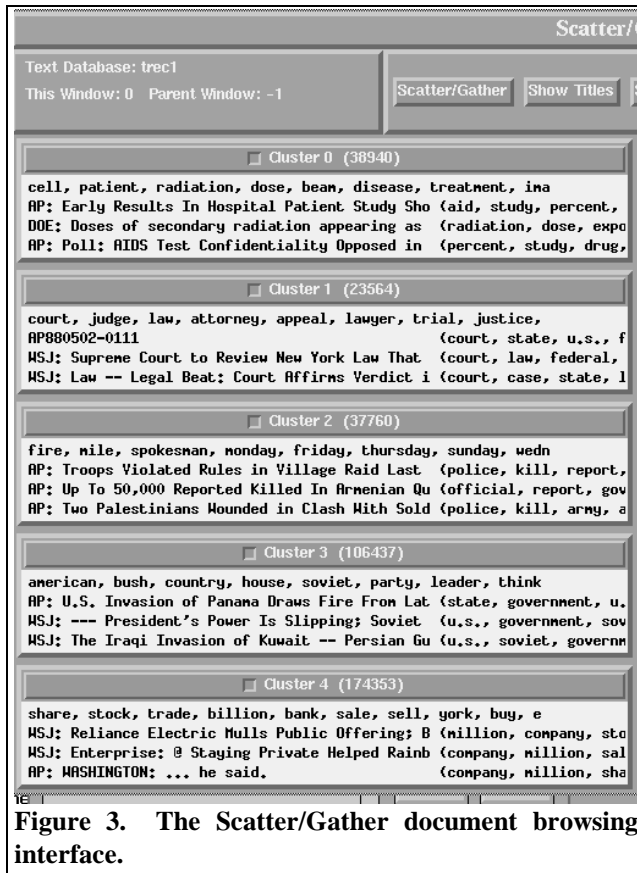


Figure 3. The Scatter/Gather document browsing interface.

to process the cluster. The Information Diet model suggests that the optimal *cluster diet* can be constructed by selecting clusters in order of descending profitability up to a threshold. That threshold is at the point where the overall average rate of finding relevant documents will be decreased by adding the next lower ranked cluster.

More generally, assume [as in 15, 16, 18] that users classify information items, i , according to their expected net value, g_i , and expected cost to handle h_i (i.e., the time it would take to make use of the information in a document), and that this classification is done rapidly. The profitability, π_i , is defined as

$$\pi_i = \frac{g_i}{h_i}.$$

Then, the following algorithm can be used to determine the rate-maximizing subset of the n information item types that should be selected:

- Rank the item types by their profitability, $\pi_i = g_i/h_i$, and let the index i be ordered such that $\pi_1 > \pi_2 > \dots > \pi_i > \dots > \pi_n$.
- Add item types to the selected set in order of increasing rank until for the final included item of rank j , the rate of gain for the j items, R_j is greater than the profitability of the $j+1$ st item,

$$R_j = \frac{\sum_{i=1}^j g_i}{\sum_{i=1}^j (s_i + h_i)} > \frac{g_{j+1}}{h_{j+1}} = \pi_{j+1} \quad (6)$$

where s_i is the time spent searching for items of type i .

We present below the heuristic evaluations that are performed in the ACT-IF model that attempt to achieve this rate-maximizing information diet.

* Published, 1998, in the *Proceedings of the Working Conference on Advanced Visual Interfaces, AVI '98* (pp. 83-93), L' Aquila, Italy. New York: ACM Press.

Information Patch Model for Scatter/Gather

Conceptually, divide the user's interaction with Scatter/Gather into two parts: (1) a browsing phase that takes t_s amount of time during which the user repeatedly gathers and scatters clusters and (2) a display phase taking time t_h during which the user will display the contents of a document cluster and forage for relevant documents. During the display phase, the user scans through a scrolling list of document citations, and must spend t_N time processing each document citation, and an additional t_R time processing relevant documents (e.g., cutting and pasting them into a bibliography, or reading them), so

$$t_h = t_N + t_R.$$

Let g be the number of relevant documents that a user can collect during the display phase. Then the rate of gain, R , in this case would be:

$$\begin{aligned} R &= \frac{g}{t_s + t_h} \\ &= \frac{g}{t_s + t_N + t_R} \end{aligned} \quad (7)$$

Generally as the user invests time t_s in gathering and scattering clusters, they reduce the eventual time t_h that they will need to spend foraging through display lists. This happens because they increase the proportion of relevant documents that will eventually get displayed, while reducing the total number of documents that will get displayed. In general, there are diminishing returns on investing time t_s , because at first the proportion of relevant documents in user-gathered clusters increases rapidly and then tapers off (eventually asymptoting near the theoretical limit of one). Note that this means that t_N and t_R both decrease as the user gathers and scatters clusters, but the ratio of t_R/t_N increases.

The optimization problem facing the user is to determine the optimal investment of time to gathering and scattering clusters, t_s , before moving on to the display and foraging phase. As we discuss below, the ACT-IF model contains heuristics that estimate the parameters in Equation 7. So long as it appears that additional investments in the browsing phase will increase R , then the model continues to browse. When it estimates that additional investments in browsing will decrease R , it moves to the display phase.

ACT-IF

The ACT-IF model [14, 15] combines the ACT-R theory of cognition [1, 2] with IF predictions. The model is instantiated as a computer simulation system

that matches its predictions against log files collected from Scatter/Gather users [using a model-tracing architecture to do the matching, 14]. Here we present simulation results matched against data from $N = 8$ Scatter/Gather users. These data were available from an earlier study done for different purposes [17].

ACT-IF is a mechanistic production system model of cognition. It consists of a *production memory* and a *declarative memory*. The declarative memory basically models the information being attended to, goal information, and information that has been recalled (*activated*) from long-term declarative memory. The production memory contains production rules which are patterns of the general form *Condition* \rightarrow *Action*. ACT-IF operates on a basic *match-execute cycle*. During the match phase, the condition part of the production rule patterns are matched against information in working memory. Those that match are then ranked by an evaluation function, the best match is selected, and its action pattern is executed during the execution phase. Actions specify updates to declarative memory, setting of goals, and actions to be performed in the world.

The raw observed data were the logged interaction protocols obtained in the Pirolli et al [17] study. To model-trace (match) a participant's log file, the ACT-IF production system is initialized with production rules and declarative information relevant to the task. The model-tracer processes the sequence of participants' actions that were observed in the log file, and it uses this to maintain a simulated model of the Scatter/Gather screen state seen by each user. Changes in screen state are "perceived" by the ACT-IF production system, which means that declarative memory elements are created when corresponding objects "appear" on the screen in the screen state model. The two main types of windows of interest are (1) the Scatter/Gather windows, which present the cluster summaries, and (2) titles display windows which present the titles of all the documents in a cluster. Each cluster summary consists of topic words and typical titles plus profile words (typics).

The model of Scatter/Gather use consists of 15 production rules. These are glossed in Figure 4. On the left of the arrow are mnemonic names for the productions and the conditions for matching declarative memory. The right side of the arrows are the actions of the production rules. Some productions are annotated with a "(2)" to indicate that there are actually two copies of the productions for the two different types of window. Our discussion will focus mainly on the productions that select and de-select clusters, and the productions that do Scatter/Gather or display titles.

NOTICE-NEW-WINDOW (2)	\rightarrow	Attend to it
New window on screen		& set goal to process it

* Published, 1998, in the *Proceedings of the Working Conference on Advanced Visual Interfaces, AVI '98* (pp. 83-93), L' Aquila, Italy. New York: ACM Press.

ATTEND-TO-WINDOW (2) Attend to window	→	Look at window
UNATTEND-TO-SCREEN (2) Goal is to process a window & different window has appeared	→	Pop the goal
SHIFT-ATTENTION Another window is present	→	Attend to that window
PROCESS-CLUSTERS Goal is to process S/G window	→	Set goal to process clusters
PROCESS-NEXT-CLUSTER Goal is to process S/G window clusters & one is unprocessed	→	Set goal to process next cluster
LOOK-AT-NEXT-CLUSTER Goal is to process next cluster	→	Look at cluster & pop the goal & set goal to process cluster elements
LOOK-AT-CLUSTER- ELEMENTS Goal is to look at cluster elements	→	Look at topics and typics & pop the goal
SELECT-RELEVANT- CLUSTER Goal is to process SG window & there is a query & there is an unselected cluster	→	Select the cluster
DESELECT-RELEVANT- CLUSTER Goal is to process SG window & there is a query & there is a selected cluster	→	Deselect the cluster
DO-SCATTER/GATHER Goal is to process SG window & some clusters have been selected	→	Scatter/Gather the window
DO-DISPLAY-TITLES Goal is to process SG window & some clusters have been selected	→	Display the titles in the window

Figure 4. Production rules used in the ACT-IF model of the Scatter/Gather protocols obtained in Pirolli et al. [17]

Relevance Computed by Spreading Activation

The ACT-IF model uses heuristic evaluation functions to select production rules. The evaluations are based on estimating the relevance of items displayed on the Scatter/Gather screen. The relevance computation in ACT-IF is based on extension of *spreading activation* mechanisms used in cognitive psychology to model human memory [1, 4]. The basic idea is that a query task activates concepts in human memory, and similarly text on the display screen activates memory concepts. Activation spreads from these concepts to related concepts in a *spreading activation network*. The amount of activation accumulating on the query and display concepts is an indicator of their mutual relevance.

It has been argued [3, 5] that human memory is adapted (optimized) to the statistical structure of events in the world. ACT-IF is consistent with this assumption. It uses a spreading activation network to compute relevance, and the structure and processing of this network reflects the statistical structure of text in on-line document collections. In essence, it can represent human word concepts for all the words in the text collections accessed by Scatter/Gather users.

When ACT-IF turns its focus of attention to the text of a cluster summary (usually about 20-30 words of text on the screen), these words become activated. Activation spreads through the spreading activation network in declarative memory. For each cluster-matching production ACT-IF computes the activation of the users query that has spread from the user's focus of attention on the screen.

The activation of a query word i is

$$A_i = B_i + \sum_j W_j S_{ji} \quad (8)$$

where B_i is the *base-level* activation of word i , S_{ji} is the *association strength* between cluster word j and query word i , and W_j is the base level activation of cluster word j . Equation 8 is a Bayesian prediction of the relevance of one word in the context of other words [3, 14]. A_i in Equation 8 is interpreted as reflecting the log posterior odds that i is relevant, B_i is the log prior odds of i being relevant, and $W_j S_{ji}$ reflects the log likelihood ratios that i is relevant given that it occurs in the context of word j .

These log likelihoods were all calculated directly from the Tipster text corpus [14]. That is, B_i is calculated from base rates of individual words, and the association strengths are calculated from the rates at which words co-occur with each other (within a span of 40 words).

ACT-IF assumes that the users of Scatter/Gather make an assessment, $g(c, s)$, of the amount of activation spread between each cluster c summary text on their screen state s and the words contained in the declarative memory representation of the query on which they are working. We modeled this activation based assessment as

$$g(c, s) = \exp\left(\frac{A(c, s)}{T}\right) \quad (9)$$

where $A(c, s)$ is the sum of the activation of all the words in the query that is received from cluster c , and T is a scaling factor. How we estimated T is discussed below.

Figure 5 shows the match of the activation-based assessment in Equation 9 to the clusters selected by

users. The log files contain points at which a Scatter/Gather display was viewed and clusters were selected by the $N = 8$ users. For each of those displays, the simulation ranked the cluster-selecting productions (Figure 4) by the activation of the clusters they matched. The rank of the cluster-selecting production that actually matched users' selections was recorded by the model-tracing simulation. Figure 5 shows that the higher activation productions were more likely to match the users selection. This suggests that the ACT-IF spreading activation network, which reflects the statistical properties of the on-line text environment, is a good predictor of human judgments of relevance.

Scatter/Gather Communicates the Relevance of Document Clusters

ACT-IF was used to assess the Scatter/Gather interface. Do the words on the interface in Figure 3 communicate the underlying distribution of relevant documents across the document clusters? We can use the activation-based assessment of ACT-IF to predict how users will assess the relevance of clusters based on what they see on the interface. We would like to match this *user's* distribution against the *algorithm's* distribution of relevant documents across clusters, as computed by the underlying clustering algorithm.

The algorithms' distribution, $d_A(c)$, of relevant documents over Scatter/Gather clusters is presented in Figure 6 [15]. If the $c = 1, 2, \dots, 10$ clusters on an average Scatter/Gather are ranked in decreasing order by the proportion of relevant documents that they contain, then $d_A(c)$ is an exponentially decreasing function of c . Next, we computed $A(c, s)$ for every cluster on every screen in all participant log files in the Scatter/Gather study (recall that $A(c, s)$ depends on each participant's particular queries). We then averaged across all observed Scatter/Gather windows to obtain the averaged activations for each rank of cluster, $\bar{A}(c)$ and the corresponding average activation-based assessment of relevant documents as $\bar{g}(c)$. We then formulated the users' distribution of relevant documents using the Boltzman equation, [11],

$$d_U(c) = \frac{\bar{g}(c)}{\sum_{i=1}^{10} \bar{g}(i)} \quad (10)$$

$$= \frac{\exp(\bar{A}(c)/T)}{\sum_{i=1}^{10} \exp(\bar{A}(i)/T)}$$

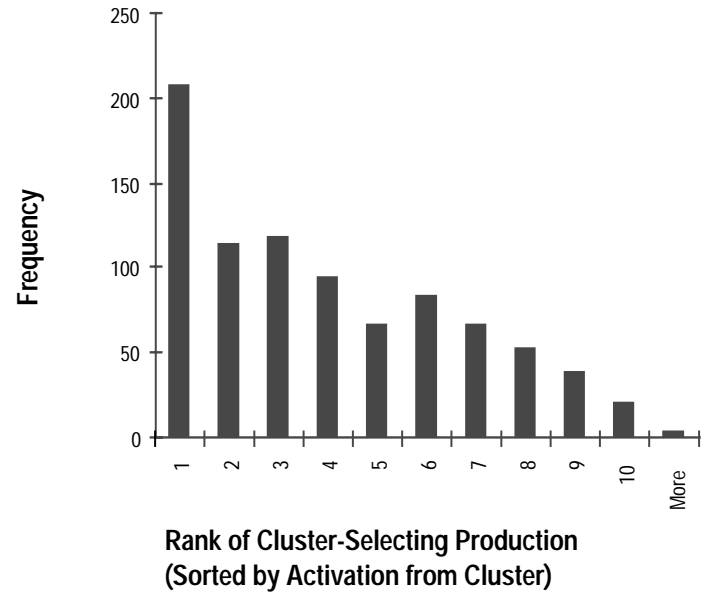


Figure 5. The activation spread to clusters from queries predicts the frequency that they were selected by users ($N = 8$).

There is one free parameter in Equation 10, which is the scaling factor T . To estimate T we fit Equation 10³ to $d_A(c)$, to obtain the curve in Figure 6. This is the only parameter estimated from the user data throughout this paper.

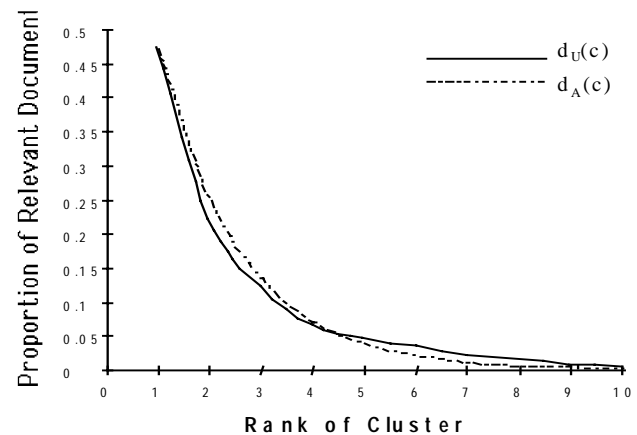


Figure 6. The underlying, algorithm's, distribution of relevant documents $d_A(c)$ and the user's distribution, $d_U(c)$ of the activation-based assessment of relevant documents across clusters c ranked in decreasing order.

The match of $d_A(c)$ to $d_U(c)$ in Figure 6 suggests that the short text summaries of clusters presented on the Scatter/Gather screen are a good reflection of the underlying distribution of relevant documents across

³Using the Levenberg-Marquardt curve fitting algorithm to minimize the mean squared error of residuals.

clusters computed by the clustering algorithm. The ACT-IF model assumes that this match is the result of using spreading activation (Equation 9) to compute the relevance of words on the display screen to a task query. The ACT-IF spreading activation network is based solely on statistical properties of on-line text, and a single scaling parameter T estimated by fitting the user data.

One may also examine what is happening to the distribution of relevant documents through time, as users iteratively gather clusters and then scatter them into a new set of clusters. Optimally, this iterative process reduces the total number of documents under consideration while increasing the proportion of relevant documents.

First, let us examine the structure of the underlying Scatter/Gather process. Let us assume that there are no backups in the process and that people iteratively gather and scatter clusters until they finally decide to display the cluster contents. Any task will involve a sequence of Scatter/Gather cluster states, $1, 2, \dots, s, \dots, S$ produced by the iterative gathering and scattering of clusters. The basic observation is that the proportion of relevant documents across all of clusters in state $s + 1$ should equal the proportion of relevant documents in the clusters that were gathered in previous state s . Letting $i = 1, 2, \dots, k$ index the k gathered clusters at any state, the following difference equation describes the distribution of relevant documents over the sequence of Scatter/Gather states:

$$\frac{\sum_{c=1}^{10} g(c, s+1)}{\sum_{c=1}^{10} N(c, s+1)} = \frac{\sum_{i=1}^k g(i, s)}{\sum_{i=1}^k N(i, s)} \quad (11)$$

That is, the total proportion of relevant documents in state $s + 1$ is equal to the proportion of relevant documents in the k clusters gathered from state s .

Next, let us examine the structure of the activation-based assessments of relevant documents. By analogy to Equation 11, the appropriate difference function should be

$$\frac{\sum_{c=1}^{10} \exp[A(c, s+1)/T]}{\sum_{c=1}^{10} N(c, s+1)} = \frac{\sum_{i=1}^k \exp[A(i, s)/T]}{\sum_{i=1}^k N(i, s)} \quad (12)$$

if there were a perfect match of the cues from the Scatter/Gather display to the underlying Scatter/Gather clustering structures (we are using the value of T estimated earlier). Figure 7 plots the data obtained from

* Published, 1998, in the *Proceedings of the Working Conference on Advanced Visual Interfaces, AVI '98* (pp. 83-93), L' Aquila, Italy. New York: ACM Press.

our ACT-IF model-tracing analysis. Each point ($N = 302$) in Figure 7 represents model values from a state s to state $s + 1$ transition. The abscissa plots the model values for the right side of Equation 12 and the ordinate plots the model values for the left side of Equation 12 (both scales are logarithmic). There is a good correlation ($R^2 = .76$) at the predicted slope of $+1$. This shows again that the text on the Scatter/Gather window accurately reflects the distribution of available relevant documents through time.

We should point out that the fit in Figure 7 is not a consequence of fitting curves in Figure 6. Figure 6 examines the average distribution of relevant documents over clusters within a Scatter/Gather display (e.g., Figure 3). Figure 7 concerns the changes in activation from one Scatter/Gather display to the next. For each Scatter/Gather display, the system computes new text summaries. There is no *a priori* reason why the activation from text on a Scatter/gather display should equal the activation from summaries of the clusters chosen on the previous display.

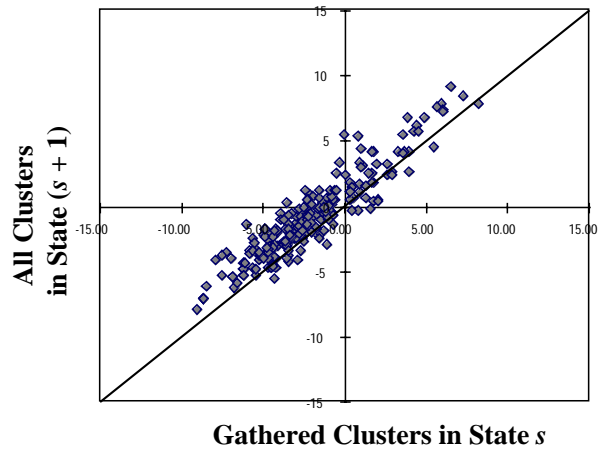


Figure 7. Correlation of log expected proportion of relevant documents in state $(s + 1)$ against the log expected proportion of relevant documents in the gathered clusters from state s as computed by ACT-IF.

Optimization of Information Foraging

We were also interested in modeling users' detailed behavior using our activation-based assessment of relevance. Here is an outline of the heuristic evaluation of the core ACT-IF productions for Scatter/Gather.

- The DO-DISPLAY-TITLES production is evaluated on the basis of the *current* overall rate of gain, $R(k, s, t)$, that will be produced for the current set of k clusters that have been gathered on the current

Scatter/Gather cluster display window (state s) at time t in the task.

- The DO-SCATTER-GATHER production is evaluated on the *projected* overall rate of gain $R(k, s, t + \Delta)$ that will be produced for the current set of k selected clusters in state s , at current time t plus some additional time, Δ , that will be invested in another round of having the system form new clusters and selecting from those clusters.
- The SELECT-RELEVANT-CLUSTER production is evaluated on the basis of an assessment of the profitability $\pi(c, s)$ of the cluster that it matches on the current Scatter/Gather cluster display window (state s).
- The DESELECT-IRRELEVANT-CLUSTER, matches an already selected cluster, but is evaluated on the basis of the maximum of the current rate of gain, $R(k, s, t)$, or the projected rate of gain, $R(k, s, t + \Delta)$.

The activation-based evaluations of DO-DISPLAY-TITLES and DO-SCATTER-GATHER serve to implement the Information Patch model discussed earlier. The evaluations of SELECT-RELEVANT-CLUSTER and DESELECT-IRRELEVANT-CLUSTER serve to implement the Information Diet model. These evaluations are computed locally, based only on declarative information matched by a production rule plus a time parameter.

The ACT-IF model for the Scatter/Gather task assumes that the profitability is evaluated by:

$$\pi(c, s) = \frac{g(c, s)}{\tau_1 g(c, s) + \tau_2 N(c, s)} \quad (13)$$

where $g(c, s)$ is an activation-based assessment of the number of relevant documents in cluster c , in Scatter/Gather state s . $N(c, s)$ is the total number of documents in a cluster (obtained from information on the display), and τ_1 and τ_2 are time-cost rate parameters which we set to be 1 second/information item.

The time cost to process all the documents in a cluster is therefore the term $\tau_1 N(c, s)$, which is an estimate of t_N in Equation 7. The additional costs of processing relevant documents is the term $\tau_2 g(c, s)$, which is an estimate of t_R in Equation 7.

The expected rate of gain, R , for a particular task and query is:

$$R(k, s, t) = \frac{\sum_{i=1}^k g(i, s)}{t_s + \tau_1 \sum_{i=1}^k g(i, s) + \tau_2 \sum_{i=1}^k N(i, s)} \quad (14)$$

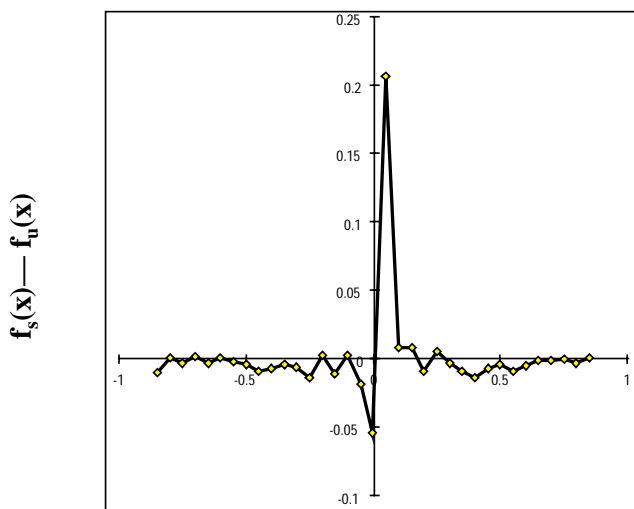
where the summations are over $i = 1, 2, \dots, k$ clusters that have been gathered at a particular state s , and t_s is the time spent so far in getting to state s . This corresponds to t_s in Equation 7. The expected rate of gain is therefore computed as the activation-based assessment of the total relevant documents in the gathered clusters (from Equation 9) divided by the sum of the time taken so far plus the expected future time it will take to process relevant and irrelevant documents in the gathered clusters.

A strong test of the Information Diet model in the example of Scatter/Gather concerns the selection of Scatter/Gather clusters. Clusters at state s should be selected so long as their profitability, $\pi(c, s)$, is greater than the overall rate of gain for the clusters gathered at state s , $R(k, s, t)$. We used the ACT-IF model-tracing simulation to collect the data relevant to this prediction.

For every Scatter/Gather display in our user logs we determined the clusters that were selected or not selected by the user. We also determined the profitability of the cluster $\pi(c, s)$ as well as the expected rate of gain $R(k, s, t)$ for that user at that point in the log. If we let $x = \pi(c, s) - R(k, s, t)$, then the prediction is that users should choose clusters when x is positive (cluster profitability > expected rate of gain) and should not choose clusters when x is negative (cluster profitability < expected rate of gain). Using these data from the logs, we computed the probability that a user selected a cluster as a function of x , $f_s(x)$, and the probability that a cluster was unselected, $f_u(x)$. In Figure 8 we plot the difference $f_s(x) - f_u(x)$ as a function of x . As predicted, it appears that there is a threshold at $x = 0$ separating the decision to select vs not select clusters. Below that threshold, when cluster profitability is less than the expected rate of gain, clusters tend to be unselected. Above that threshold, when cluster profitability is greater than the expected rate of gain, the clusters tend to be selected. This is what is predicted by the Information Diet model. Despite the fact that cluster profitabilities and expected rates of gain change dynamically from state to state and over time, it appears that users are continuously assessing and optimizing their information foraging behavior in a manner consistent with the predictions of IF theory.

GENERAL DISCUSSION

To assess new information technologies we have been developing Information Foraging theory. Models developed in the theory attempt to analyze the value of information gained by using an interface in relation to the cost of interacting with the interface. Models developed for the Butterfly interface illustrated how switching from a single-threaded interface to a multi-threaded one affected foraging. A more refined cognitive model, ACT-IF was used to show how well the simple Scatter/Gather interface communicates the location of relevant documents. Moreover, the behavior of users was well-matched by the ACT-IF model. This suggests that users may be understood by assuming that they are making bounded rational decisions that aim to optimize their browsing behavior.



$$x = \pi(c, s) - R(k, s, t)$$

Figure 8. The difference in observed probability density distributions for selecting clusters $f_s(x)$, and or leaving them unselected clusters $f_u(x)$, as a function of the distance between cluster profitability and the estimated of rate of gain: $x = \pi(c, s) - R(k, s, t)$. The tendency is to select clusters when $x < 0$ and to select when $x > 0$.

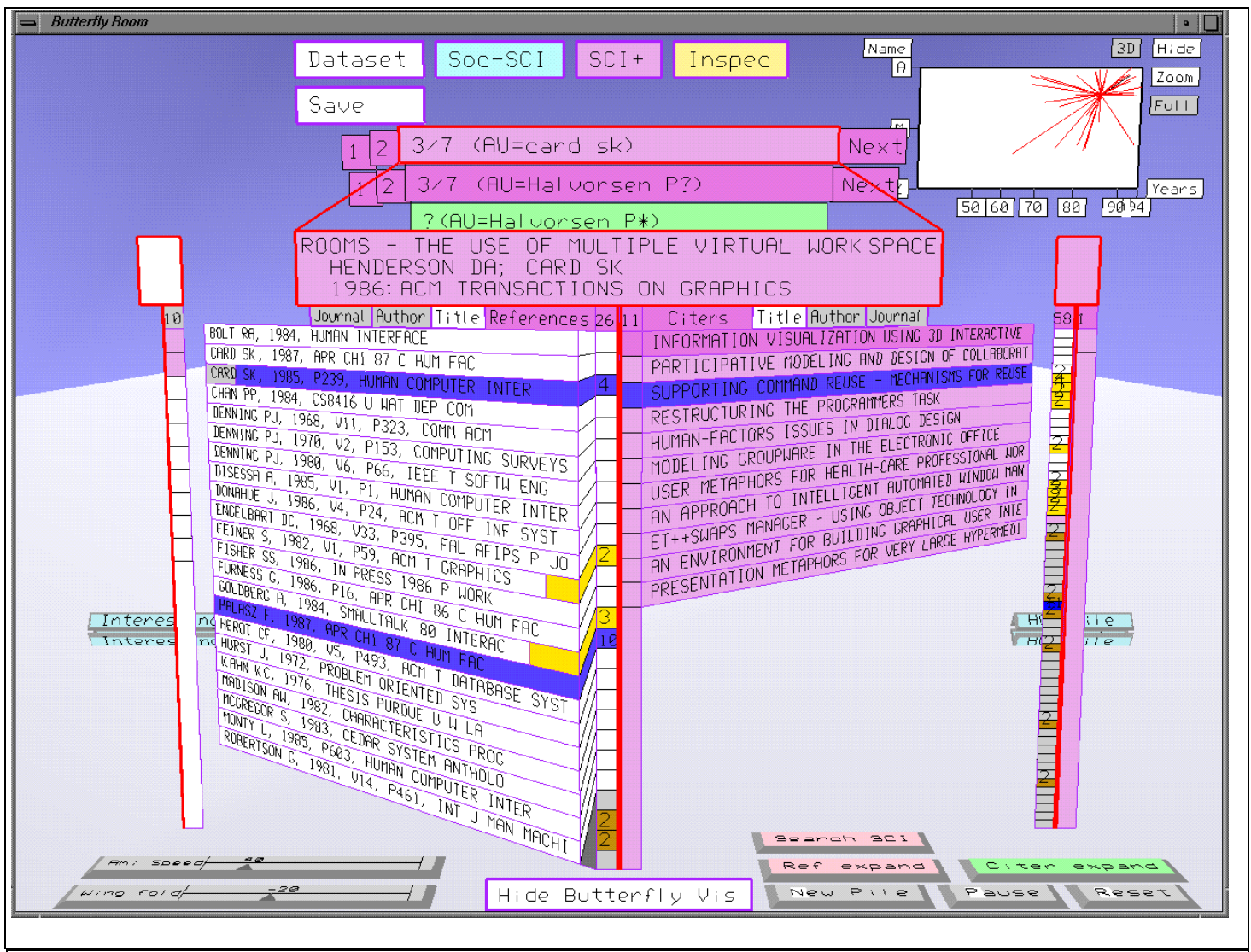
ACKNOWLEDGMENTS

This work is supported by an Office of Naval Research grant No. N00014-96-C-0097.

REFERENCES

1. Anderson, J.R. (1993). *Rules of the mind* Hillsdale, NJ: Lawrence Erlbaum Associates.

2. Anderson, J.R. and C. Lebiere. (in press). *The atomic components of thought* Mahwah, NJ: Lawrence Erlbaum Associates.
3. Anderson, J.R. and R. Milson. (1989). Human memory: An adaptive perspective. *Psychological Review*, 96, 703-719.
4. Anderson, J.R. and P.L. Pirolli. (1984). Spread of activation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 10, 791-798.
5. Anderson, J.R. and L.J. Schooler. (1991). Reflections of the environment in memory. *Psychological Science*, 2, 396-408.
6. Belkin, N.J. and W.B. Croft. (1992). Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, 35, 29-38.
7. Charnov, E.L. (1976). Optimal foraging: The marginal value theorem. *Theoretical Population Biology*, 9, 129-136.
8. Cox, D.R. and W.L. Smith. (1961). *Queues* London: Wiley.
9. Cutting, D.R., D.R. Karger, J.O. Pedersen, and J.W. Tukey. (1992). Scatter/gather: A cluster-based approach to browsing large document collections. In Proceedings of the SIGIR '92 (pp. 318-329),
10. Harman, D. (1993). Overview of the first text retrieval conference. In Proceedings of the 16th Annual International ACM/SIGIR Conference (pp. 36-38), Pittsburgh, PA: ACM.
11. Kruschke, J.K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. 99, 22-44.
12. Mackinlay, J., R. Rao, and S. Card. (1995). An organic user interface for searching citation links. In CHI '95, ACM conference on human factors in software (Denver, Colorado, May 7-11, 1995) (pp. 67-73). New York: ACM.
13. McNair, J.N. (1983). A class of patch-use strategies. *American Zoologist*, 23, 303-313.
14. Pirolli, P. (1997). Computational models of information scent-following in a very large browsable text collection. In Proceedings of the Conference on Human Factors in Computing Systems, CHI '97 (pp. 3-10), Atlanta, GA: Association for Computing Machinery.



Color Plate 1 (Figure 1). The Butterfly citation browser.

<p>15. Pirolli, P. and S. Card. (1997). <i>The evolutionary ecology of information foraging</i>. Technical Report, Palo Alto, CA: Xerox PARC.</p> <p>16. Pirolli, P. and S.K. Card. (1995). Information foraging in information access environments. In <i>Proceedings of the CHI '95, ACM Conference on Human Factors in Software</i> (pp. 51-58), New York: ACM.</p> <p>17. Pirolli, P., P. Schank, M. Hearst, and C. Diehl. (1996). Scatter/Gather browsing communicates the topic structure of a very large text collection. In <i>Proceedings of the Conference on Human Factors in Computing Systems, CHI '96</i> (pp. Vancouver, BC: Association for Computing Machinery.</p>	<p>18. Stephens, D.W. and J.R. Krebs. (1986). <i>Foraging theory</i> Princeton, NJ: Princeton University Press.</p> <p>19. vanRijsbergen, C.J. (1979). <i>Information retrieval</i> (2nd ed.). Boston, MA: Butterworth & Co.</p>
--	--

* Published, 1998, in the *Proceedings of the Working Conference on Advanced Visual Interfaces, AVI '98* (pp. 83-93), L'Aquila, Italy. New York: ACM Press.