# ScentTrails:
# Integrating Browsing and Searching on the Web

CHRIS OLSTON
Stanford University
and
ED H. CHI
Palo Alto Research Center

---

The two predominant paradigms for finding information on the Web are browsing and keyword searching. While they exhibit complementary advantages, neither paradigm alone is adequate for complex information goals that lend themselves partially to browsing and partially to searching. To integrate browsing and searching smoothly into a single interface, we introduce a novel approach called ScentTrails. Based on the concept of information scent developed in the context of information foraging theory, ScentTrails highlights hyperlinks to indicate paths to search results. This interface enables users to interpolate smoothly between searching and browsing to locate content matching complex information goals effectively. In a preliminary user study, ScentTrails enabled subjects to find information more quickly than by either searching or browsing alone.

**Categories and Subject Descriptors**: H5.1 **[Multimedia Information Systems]:** Hypertext Navigation**;** H.1.2 **[Information Systems]:** User/Machine Systems-*human factors***;** D.2.2 **[Software Engineering]:** Tools and Techniques-*user Interfaces***;** H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems--*Evaluation/methodology.*
**General Terms**: Human Factors, Algorithms, Experimentation, Performance
**Additional Key Words and Phrases**: ScentTrails, Information Scent, Searching, Browsing, World Wide Web

---

## 1. INTRODUCTION

Finding information on the World Wide Web (WWW) is often difficult. Currently, there are two predominant interface modes for locating information: *browsing* and *searching*. Browsing is the process of viewing pages one at a time and navigating between them sequentially using hyperlinks. Searching is the process of entering a *search query* (usually a list of keywords) into a search engine, which produces a ranked list of links to pages that match the query. (These two paradigms of locating information are more appropriately termed by Jul and Furnas [1997] as "search by navigation" and "search by query," respectively, but we will use the more common terms "browsing" and "searching.")

Authors' addresses: Chris Olston, Department of Computer Science, Stanford University, California 94305; Ed H. Chi, Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, California 94304.

Both paradigms are problematic. In general, browsing is not an efficient means of locating specific information, because users must painstakingly guide themselves using *browsing cues*: textual or graphical indications of the content reachable via a link. Searching, on the other hand, often returns inappropriate results and loses the important context present in the pages leading to the search result.

Although browsing and searching each have disadvantages, they both persist due to their complementary advantages [Jul et al. 1997, Manber et al. 1997]. Searching is popular because of its ability to identify pages containing specific information quickly—each search is tailored specifically to the user's particular information need. On the other hand, browsing is useful in cases where appropriate search keywords are either nonexistent or unavailable to the user for a variety of possible reasons.

First, during browsing the user may not be certain of what he or she is looking for until the available options are presented. For example, a user shopping for photocopiers may not be aware *a priori* of the choice between digital and analog copiers. Alternatively, the user may start with only a vague notion of what he or she wants. In the process of browsing, the exact criteria may crystallize. For instance, a customer may wish to find a "fast" copier. Upon browsing, he or she observes the range of speeds available and is able to form a quantitative criterion for "fast." Moreover, users may sometimes alter their initial criteria while browsing as new information becomes available, *e.g.*, if a slower copier is on sale.

Second, certain criteria do not lend themselves well to keyword search, so browsing is more appropriate. For instance, customers may wish to find copiers that can perform at least 20 copies per minute. It is difficult to search for such criteria using existing technology.

Third, the exact terminology used on the Web pages may not be known, so searching would not yield the correct result. For example, customers searching for devices that can copy, print, scan, and fax may not be aware that the vendor employs the term "multifunction" in the product descriptions.

Finally, browsing is appropriate in cases where a great deal of information and context is obtained along the browsing path itself, not just at the final page. For example, consider that pages about monochrome copiers are not likely to advertise the fact that they do not support color copying. However, suppose the path from the entry point of the site to the page describing the copier includes a page with links to copiers by category, some supporting color copying and others not. In this situation, it may only be possible to determine whether a certain copier supports color by inspecting the browsing cues leading to the copier description page.

So, as have others in the literature, e.g., [Manber et al. 1997], we argue that searching is more appropriate in some cases, whereas browsing is more appropriate in others. Complex information goals are made up of several criteria, often with some criteria in each category. In our example above, consider a user seeking a fast digital copier. Search is probably appropriate for finding digital copiers, whereas browsing is most suitable for identifying "fast" copiers since it is a relative criterion and is difficult to represent using search keywords.

Although desirable, it is difficult to reap the benefits of both browse and search simultaneously. On one hand, if the user starts with a search, it is difficult to recover any contextual information embedded in the path leading to the result pages. If, on the other hand, the user begins by browsing, it is difficult to perform a search that takes advantage of the insight gained during the initial browsing period.

To bring together the strengths of browsing and searching into a single interface, we propose ScentTrails. ScentTrails automatically highlights hyperlinks on Web pages to guide users toward search results while still allowing them to perform conventional browsing. In ScentTrails, the user enters keywords representing the portion of his or her information goal that is initially known and amenable to search. ScentTrails then annotates the hyperlinks of Web pages with search cues: indications that a link leads to content that matches the search query. This annotation is done by visually highlighting links to complement the browsing cues already embedded in each page. The degree to which links are highlighted is determined by an Information Scent algorithm described later in this paper. By considering both the search and browsing cues together, the user is able to make informed navigational decisions and efficiently locate content matching complex information goals that lend themselves partially to searching and partially to browsing.

In this paper we present the ScentTrails technique, specifically outlining our algorithm for determining which hyperlinks to highlight and by how much. We also present a preliminary study that points to its effectiveness. The remainder of this paper is structured as follows. First, we discuss related work. Then we describe the Information Scent concept underlying our approach before presenting the ScentTrails technique and its algorithm. Next, we present our preliminary study evaluating ScentTrails, as compared to pure browsing and searching. Finally, we discuss link highlighting methods and avenues for future work.

## 2. RELATED WORK

It is widely recognized that the traditional Web browsing interface based solely on navigational access leaves room for improvement. Consequently there is a large body of work related to improving the Web browsing interface in a variety of disciplines such as the information retrieval, hypermedia, user interfaces, artificial intelligence, and Web research communities. Here we cover some of this work.

A preliminary formal characterization of several aspects of navigational interfaces is established in [Jul et al. 1997], which also offers some perspectives on designing content for easy navigation or browsing. Our approach to assisting with navigation of Web sites is based on integrating searching with browsing, and many other approaches to integrating the two paradigms are proposed in the literature. To begin with, some approaches achieve a moderate level of search/browse integration by facilitating transition between searching and browsing and making it easier to combine the results. Greer et al. [1997] propose a "TourGuide" interface in which the user iterates between browsing and requesting page recommendations (a form of searching). Manber et al. propose a system in which each page's search function is amended based on the context of that page [Manber et al. 1997].

Other work focuses on displaying Web search results in the context of surrounding documents using overviews of hyperlink topology such as hierarchical outline displays or (custom) *Web maps* (*e.g.*, [Hersovici et al. 1998, Park et al. 2000]). For example, in *Cha-Cha* [Chen et al. 1999] and *AMIT* [Wittenburg et al. 1997], search results are organized automatically into a nested list to show their context within the hierarchical structure of one or more Web sites. Also, *Yahoo!* [2003] represents search results as hierarchical references to Yahoo's navigable catalogue of Web sites, when applicable. In *WebCutter* [Maarek et al. 1997], search results are shown as highlighted nodes in a Web map visualization, helping the user understand the context of search results within the surrounding Web topology. While these approaches are similar in spirit to ours, they require users to work with a representation of document collections (nested list or Web map), in lieu of or in addition to the standard single-page representation traditionally used for viewing documents. In contrast, our approach relies uniquely on adding guiding indicators to the regular single-document representation used for traditional browsing.

A different approach is taken by, *e.g.*, [Golovchinsky 1997, Lieberman 1995, Perkowitz et al. 1999, Yan et al. 1996], in which hyperlinks are generated dynamically based on user need and displayed in Web documents next to existing static browsing links. However, in contrast to our work, this approach does not focus on layering annotations or indicators directly on top of hyperlinks using the existing hyperlink structure and browsing interface.

Annotating hyperlinks in regular single-document views is one goal of the Web-Based Intermediaries (WBI) Project [Barrett et al. 1997, Campbell et al. 1999], and we have used their framework in our prototype implementation. Most notably, the work in [1999] proposes to annotate hyperlinks based on expected latency. Campbell and Maglio showed that users avoided slow links that the browsing cues indicated were unlikely to be relevant, and that the type of link highlighting used significantly impacts the effectiveness. By contrast, ScentTrails uses link annotation to indicate paths to documents that match search criteria.

The extensive literature on Adaptive Hypermedia [Brusilovsky 1996] includes some work on hyperlink annotation. However, work in that area has typically focused on automatically inferring user goals in specialized hypermedia applications such as on-line help systems.

The *SuperBook* [Egan et al. 1989] and *NaviQue* [Furnas et al. 1998] interfaces both annotate or highlight paths to information matching ad-hoc search queries. Rather than arbitrary hyperlinks on each page, SuperBook features hierarchical linking through a table of contents view. NaviQue was designed for a graphical zooming environment. Neither of these methods was designed for a Web context, which features graph-structured information and a single page-at-a-time browsing metaphor. Moreover, in these approaches, the value of information matching a query is not discounted according to the browsing distance (number of clicks required), which is crucial in a Web context. In contrast, we have designed our technique specifically with the Web in mind.

Furnas recognized the potential benefit of highlighting Web hyperlinks according to the result of an ad-hoc query [Furnas 1997]. This idea was also the focus of work on the WebWatcher project [Joachims et al. 1997, Mladenic 2001], which uses various learning techniques to highlight links according to user needs. However, no evaluations of the time required by users to locate information with these techniques were conducted. By contrast, we have conducted an empirical study showing that ScentTrails significantly reduces task completion time compared with both regular searching and browsing for the tasks studied. The ScentTrails technique is based on a concept called Information Scent, which is described next.

## 3. INFORMATION GOALS AND INFORMATION SCENT

The ScentTrails technique is based on the theoretical notion of *information scent* [Chi et al. 2001, Pirolli 1997] developed in the context of *information foraging theory* [Pirolli et al. 1999]. In this context, Information Foraging is related to other studies, such as Berrypicking [Bates 1989] and ASK [Belkin 1980], of how users interleave directed structured behavior (searching) with opportunistic and unstructured behavior (browsing). Users typically forage for information on the Web by navigating from page to page along hyperlinks. The content of pages associated with these links is usually presented to the user by some snippets of text or graphic called *browsing cues*. Foragers use these browsing cues to access the *distal content*: the page at the other end of the link. Information scent is the imperfect, subjective perception of the value, cost, or access path of information sources obtained from browsing cues[1].

We are interested in the frequent case where users have some *information goal* – some specific information they are seeking – when they visit a Web site via either a search or browse interface. To perform a search, the user selects keywords that correspond to their information goal. When choosing from a list of search results or from a set of outgoing links on a page being browsed, the user examines some of the links and compares the cue (*i.e.*, link anchor and/or surrounding text) with his or her information goal. The user takes the degree of similarity as an approximation to how much the content reachable via that link coincides with their information goal.

Chi et al. introduce an algorithm called WUFIS that simulates the expected surfing behavior of users following existing, imperfect browsing cues to satisfy their information needs [Chi et al. 2001]. As discussed later, our ScentTrails algorithm employs some techniques that are similar to those used in WUFIS. However, the goal of our work on ScentTrails is quite distinct. Specifically, the goal of ScentTrails is not to understand how users would browse using existing and likely imperfect cues, but rather to assist browsing by adding new supplemental cues tailored to the needs of individual users. The supplemental cues do not simply convey the likelihood with which a user is expected to follow links, but instead provide guidance based on distal content that lies one or more clicks away and is not currently visible to the user. Before describing ScentTrails in detail we first motivate the need for supplemental cues by discussing some of the limitations of traditional browsing cues.

---

[1] Furnas referred to such information cues as "residue" [10].

### 3.1 Browsing Cues Have Limited Scent

It is well known that users often find it difficult to satisfy their information goal quickly while browsing a Web site. The reason for this difficulty lies in the fact that browsing cues often lack enough information to conduct the appropriate scent to the user. This limitation can be attributed to at least three causes. First, poor link labeling can lead to inappropriate cues. Second, since each Web page tends to contain large number of potential destinations, the cues are typically short and thus cannot convey a large amount of information. Third and most importantly, browsing cues are usually not customized based on each user's information goal. Instead, they tend to conduct scent that represents the aggregate distal information, which is often more diluted than the specific portion of the distal information that coincides with a particular user's information goal.

## 4. SCENTTRAILS

We introduce a technique called ScentTrails to augment the limited capability of browsing cues with an extra layer of scent that corresponds closely with the user's information goal. A ScentTrails user can input a list of search terms (keywords) into an input box at any point while browsing. These keywords represent the user's *partial information goal*, or the portion of the user's information goal that is known and representable as keywords at any one time. ScentTrails highlights hyperlinks on the current page that lead toward pages that match this partial information goal. The user navigates by using a combination of pre-existing browsing cues and the search cues provided by the link highlighting. At any point, the user is able to modify the search terms as his or her partial information goal changes and crystallizes.

Fig. 1. A sample Web page whose link anchors have been highlighted by ScentTrails for the partial information goal "remote diagnostic technology."

Figure 1 shows a screen shot of a Web page that has been highlighted by ScentTrails using the keywords "remote diagnostic technology." In this example, link highlighting is achieved by increasing the font size of the link anchor text. However, as discussed below, other link highlighting methods may be used instead. The size of each link anchor (selected from seven possible font sizes) indicates how well the pages accessible via that link match the partial information goal "remote diagnostic technology." Suppose that the remainder of the information goal is to find copiers that can perform at least 75 copies per minute. Existing search technology would have difficulty isolating the value corresponding to copier speed, which would require sophisticated inference since “speed” is implied by the use of a metric such as “copies/min” or “pages/min.” However, it is easy for a human to identify links to copiers having this criterion using the browsing cues on this page, a summary of products provided by the Web site designer to assist browsing. In short, for this Web site, finding a copier with remote diagnostic technology that supports at least 75 pages per minute (*i.e.*, the 5388 copier) requires the simultaneous consideration of browsing cues and search cues.

When ScentTrails highlights links, it takes into account not only the relevancy with respect to the user’s partial information goal, but also the distance (measured in number of clicks) to relevant pages. Since we know that in traditional browsing methods the likelihood that a user will continue surfing typically decreases with each click [Huberman et al. 1998], ScentTrails tries to direct users to information that is near by, as well as relevant. ScentTrails achieves this goal by discounting the value of more distant pages accordingly.

### 4.1 ScentTrails Technique

We now turn to a conceptual description of the ScentTrails technique for highlighting links. (The exact algorithm is given in the next subsection.) The first step is to identify pages that match the user’s partial information goal (search keywords). These pages are called the *search result pages*, and are the pages that would normally be returned by a search. Hyperlink paths leading to the search result pages are considered scent conduits. The scent of the search result pages is “wafted” outward along the scent conduits, losing intensity as it travels. Note that scent is wafted backward along hyperlinks—the reverse direction from browsing. Scent from multiple search result pages wafting along the same path is additive. The amount of scent reaching each link determines how much to highlight the link.

Fig. 2. A diagram of a small Web site showing some sample keywords. Star icons indicate the search result pages for the partial information goal “remote diagnostics.” The widths of the link arrows indicate the Scent strengths.

Figure 2 shows a diagram of a small Web site with some sample keywords from each page. The weights of the link arrows illustrate the link highlighting strengths. Links are highlighted according to the partial information goal “remote diagnostics.” The scent flowing out of page 5 (*i.e.*, along the link from page 4 to 5) is the strongest since it is a search result page and it also offers a link to another search result page (page 7). However, starting at page 1, the path with the strongest scent leads directly to page 7, because it is significantly closer than page 5.

There are many conceivable ways to implement link highlighting, aside from increasing the font size (as illustrated in Figure 1), that involve the use of color, animation, text outlining, etc. Furthermore, in cases where the link anchor is an image or icon instead of text, we could apply other methods more appropriate for the graphical domain, such as increasing image brightness or contrast, or simpler methods like a red box outline or increasing the size of the graphic. While in this paper we focus primarily on the algorithm used to determine how much to highlight links, described next, we provide some initial discussion of options for both automatic and designer-specified link highlighting methods at the end of the paper in Section 6.

## 4.2 ScentTrails Algorithm

The goal of the ScentTrails algorithm is to compute a *scent vector* representing the amount of scent reaching each link from the search result pages. This information is used to highlight links with different gradations. The overall algorithm used to compute the scent vector is illustrated in Figure 3. The first step is to extract the content and hyperlink topology of the Web collection. The content can be conveniently represented as matrix **W** giving occurrences of words in pages. The link topology can be represented by a normalized "to x from" adjacency matrix **T**, with each row summing to 1.

Fig. 3. ScentTrails algorithm.

Once this data has been extracted, the next step is to perform *spreading activation* [Anderson 1983, Chi et al. 2001] of the identity matrix **I** through the transpose of the link topology matrix $\mathbf{T}^{\mathbf{T}}$, removing cycles in each iteration. Removing cycles of all lengths ensures that scent is never directed back to its origin, which could result in inappropriate link highlighting. The spreading activation computation simulates scent flow backward along links to determine the degree to which scent from one page will reach another page, and is defined recursively as: $\mathbf{A}(0) = \mathbf{I}$; $A(t) = \mathbf{I} + \alpha \cdot zdiag(\mathbf{T}^{\mathbf{T}} \cdot \mathbf{A}(t - 1))$. The *zdiag*() subroutine sets the diagonal entries of the matrix to zero; we discuss α shortly. Note that spreading activation is usually performed on vectors, while our algorithm performs spreading activation over the identity matrix, with the purpose of simulating multiple spreading activation runs, one for each unit component vector.

An important parameter of spreading activation controls what fraction of the scent intensity is lost during the propagation through each link, thereby discounting the value of distant information. This *decay parameter* $\alpha$ may vary according to the number of links traversed, and can be set according to the "Law of Surfing" discovered in [Huberman et al. 1998]. The result of the spreading activation computation using $I$ iterations is the *scent conduit matrix* $\mathbf{C} = \mathbf{A}(I)$, which can be thought of as indicating the aggregate width of the scent conduit between each pair of pages. The scent conduit matrix can be precomputed from the link topology alone.

Every time a query (list of keywords) is entered, a *relevance vector* $\mathbf{r}$, assigning to each page a numerical score, is computed using some technique for estimating the relevance of documents to a query. For example, a traditional information retrieval technique such as TF.IDF [Salton et al. 1973] might be used.

Once a relevance vector has been created, the final step is to metaphorically waft scent backward from search result pages by multiplying the scent conduit matrix with the relevance vector to compute the scent vector $\mathbf{s}$ (*i.e.*, $\mathbf{s} = \mathbf{Cr}$). The intensity values in the scent vector are used to determine the link highlighting weights. For every link from page A to B, the strength of highlighting is set proportional to the vector element $\mathbf{s}[B]$ from the scent vector $\mathbf{s}$. Notice that it is not usually necessary to compute the full scent vector $\mathbf{s}$. Instead, when highlighting links on a given page A having $k$ within-site outgoing links $B_1, B_2, \ldots, B_k$, we need only compute the $k$ elements of $\mathbf{s}$ corresponding to pages $B_1$ through $B_k$. For each page $B_i$ we can compute $\mathbf{s}[B_i]$ by taking the dot product of the relevance vector $\mathbf{r}$ with the row of the scent conduit matrix $\mathbf{C}$ corresponding to $B_i$.

*4.2.1 Algorithm Complexity.* The offline portion of our algorithm, *i.e.*, precomputation of the scent conduit matrix $\mathbf{C}$ from the link topology matrix $\mathbf{T}$, has a worst-case asymptotic running-time of $O(I \cdot n^3)$, where $n$ is the number of documents in the corpus and $I$ is the number of spreading activation iterations performed. Slight improvements to the cubic running time of matrix multiplications are possible using well-known techniques. In practice, however, that for most Web sites we expect the topology matrix $\mathbf{T}$ to be rather sparse, and have significantly fewer than $n^2$ entries. In light of this observation we rewrite the complexity expression as $O(I \cdot n^2 \cdot m)$, where $m << n$ represents the average number of outgoing intrasite links per page. For our test corpus, described in Section 4.2.3, the topology matrix is very sparse, with only 0.02% non-zero entries, and $m \approx 1.6$. The number of iterations $I$ of spreading activation required depends on the depth of the site hierarchy. For our test site, $I = 5$ iterations was sufficient. Since both $m$ and $I$ tend to be quite small, it is realistic to consider the complexity of the offline precomputation phase of our algorithm to be essentially $O(n^2)$, or quadratic in the number of pages in the corpus.

We now consider the complexity of the online component of our algorithm. The relevance vector **r** is computed using conventional search engine techniques such as inverted lists combined with TF.IDF weighting, which have known scalability properties, so we do not discuss the computation of **r** further. Computing link highlighting strengths for a given page A being browsed requires computing dot products between **r** and $k$ rows of the scent conduit matrix **C**, where $k$ is the number of outgoing intrasite links from page A. The worst-case complexity of this computation is $O(k \cdot p)$, where $p \leq n$ is the number of pages matching the search keywords (*i.e.*, the number of non-zero entries in **r**). The running time can be tuned as needed by treating small entries in **r** as zero, although we did not experiment with this technique. In practice we achieved interactive response times for the entire link highlighting process, which includes this computation as one step, in our test-bed implementation running on the Xerox.com Web corpus. Our implementation is described next, and performance measurements over our test corpus are presented in Section 4.2.3.

*4.2.2 Implementation.* We implemented ScentTrails as a proxy that lies between the user and the Web server. (In an alternative architecture, Web content providers can make ScentTrails available as a feature of their Web site, eliminating the need for proxies.) A proxy (or Web site) implementing ScentTrails must store the scent conduit matrix **C**, which can be precomputed from the link topology matrix **T** since it is independent of user queries. In addition, the proxy or Web site must maintain information to support content searching and scoring (*i.e.*, inverted index, etc.), much like a search engine. Whenever a new user query (keyword list) is provided, common search engine techniques can be used to compute the relevance vector **r**, which indicates the degree to which pages match the search keywords, efficiently.

Subsequently, each Web page requested by the user is intercepted by the ScentTrails proxy, which highlights the links and forwards the highlighted page to the user. In order to highlight links on a Web page, ScentTrails must compute the entry in the scent vector **s** corresponding to the page linked to by each outgoing link. This computation is performed for each link by taking the dot product of the relevance vector **r** with one row of the scent conduit matrix **C**, which can be performed quite efficiently since the two vectors tend to be sparse. Increased efficiency can be achieved by treating small relevance and conduit strength values as zero and by caching scent values for later reuse on other links to the same page. In practice, we achieved interactive response times for browsing using our prototype implementation. Our performance results are described next.

*4.2.3 Test Data and Performance.* We tested our implementation using a May 17, 1998 crawl of the www.xerox.com Web site. Although a bit dated, this data was used for the purposes of relation to previous work in the Information Scent project. The Web site corpus consists of 7549 text and html documents. The total number of distinct words in the corpus is 30,973, and the largest document contains 58,165 words (after stemming and removing stop words). The Web site has a roughly hierarchical structure with a fair amount of cross-links and cycles, and there are a total of 11,921 intrasite links among documents in the corpus.

We used five iterations of spreading activation to compute at the outset the scent conduit matrix **C**, which is sufficient for most Web sites including the www.xerox.com site we used for testing. The running time to precompute **C** from the content and topology matrices for our test site on a 733 MHz Pentium III workstation with 256 MB of RAM was just 6.89 minutes. Since **C** only needs to be computed occasionally, our technique should scale to larger Web sites. The resulting **C** matrix had 3.6 million non-zero entries, requiring only 74 MB to store, and could easily be cached in memory. (Selective caching of individual rows of scent conduit matrices of very large Web sites is a topic for future work.) At runtime, computing the dot product between one (in-memory) row of **C** and a typical relevance vector **r** took only about 200 milliseconds on our modest workstation. These performance results suggest that our algorithm is cheap and easy to implement for existing Web sites of roughly ten thousand pages. In Section 7 we discuss avenues for scaling our algorithm to handle larger sites.

*4.2.4 Discussion.* It is important to note that ScentTrails works by first performing a search, and then wafting the scent from the search results toward the user. Therefore, any advances in search engine technology apply directly to ScentTrails.

As mentioned earlier, the ScentTrails algorithm bears some resemblance to the WUFIS algorithm [Chi et al. 2001], which also uses spreading activation. However, the algorithms differ. In particular, WUFIS uses spreading activation to cause a simulated population of users to flow along links in proportion to the degree to which existing browsing cues match their information goals. In contrast, the ScentTrails algorithm uses spreading activation to cause scent placeholders to flow along links in reverse and with all links treated equally, independent of existing browsing cues. The result is an artificial scent conduit system for propagating scent from search result pages to links on pages being browsed by users. At runtime, the artificial scent conduit information is used in conjunction with search result pages identified by user queries to create supplemental cues used to assist browsing.

## 5. EVALUATION

To study the effectiveness of ScentTrails, we conducted a preliminary user evaluation using a prototype implemented as a Web proxy using the WBI framework [Barrett et al. 1997]. Our prototype supports four information-seeking interfaces. First, it supports pure browsing trivially by leaving pages unmodified. Second, pure searching is available via a standard full-text keyword search interface that returns the matching pages in the Web site in ranked order, ten at a time. Third, in the ScentTrails interface, the user is first asked for a list of keywords. The proxy then intercepts subsequent Web pages on their way to the browser and highlights the links by increasing the font size of the link anchor text, as in Figure 1.

Finally, to determine whether the mild complexity of the ScentTrails algorithm is warranted, we implemented a variation of ScentTrails called **ShortScent**. ShortScent is identical to ScentTrails except that scent from search result pages only reaches neighboring pages, and does not travel across multiple links. In other words, the search cues in ShortScent are only computed based on a one page look-ahead. From the user's perspective, there is no visible difference between the ShortScent and ScentTrails interfaces.

In interfaces where users enter keywords (*e.g.*, the ScentTrails, ShortScent, and search interfaces), the proxy further modifies all pages by emphasizing occurrences of the search keywords. This feature makes it easy to identify relevant information within a single page. The browsing interface does not support this feature, since there are no keywords associated with browsing.

### 5.1 Experiment

*5.1.1 Subjects.* Twelve Xerox PARC employees and interns participated in our study. They had no prior knowledge of the ScentTrails project, and none were well versed on the Xerox.com consumer product line or Web site layout, which we used in our experiment.

*5.1.2 Task Materials.* The Web site data for our experiment was provided by the crawl of the www.xerox.com Web site described in Section 4.2.4, which is a corporate site of moderate size. We chose eight information-seeking tasks relevant to this specific Web site:

1. Find a copier with recyclable toner.
2. You have a business at home and you want a copier with photo support.
3. Find a copier with glossy print capability.
4. Find a digital, black & white copier that supports rotation.
5. Find a copier with remote diagnostic technology that can do at least 80 copies per minute.
6. Find a 400 dpi (dots per inch) copier with a counterfeit deterrent system.
7. Find a black & white machine that supports scan, fax, print, and copy with collation.
8. Find a 5 to 20 cpm (copies per minute) copier with photo support.

We selected these tasks based on our experience with the Xerox.com site, and verified that they represented typical ones for this site by examining feedback emails indicating tasks Xerox.com customers wanted to accomplish. Notice that we elected not to sample tasks randomly from feedback emails. Instead, since the focus of our study was to understand differences among searching, browsing, and the new ScentTrails interface, we constructed our task set by hand to ensure adequate representation of tasks amenable to searching as well as ones amenable to browsing while keeping the task set small. Admittedly, this study is somewhat limited due to the fact that we used a small task set. However, our experimental task time completion results, presented later, verify that we did succeed in obtaining a mix of tasks more amenable to searching (*e.g.*, tasks 1, 3, 6) and to browsing (*e.g.*, tasks 5, 8). Interestingly, subtle differences in the nature of the task, such as permitting a range of options for a numeric specification (*e.g.*, at least 80 cpm) versus a precise target (*e.g.*, exactly 400 dpi), can sometimes affect the amenability to searching versus browsing significantly. Our results also suggest a range of difficulty levels among tasks, *i.e.*, that certain tasks seem difficult using any interface (*e.g.*, task 7), while others appear to be easier to complete quickly (*e.g.*, task 6).

To gain some understanding of the effectiveness of our ScentTrails technique, we chose to study a prototypical corporate Web site of moderate size with tasks that are as realistic as possible. However, the reality is that our experiment only studies one site and one set of tasks specific to this site. The generalizability of the experimental results presented below is therefore somewhat limited. Clearly, to gain a broader picture of how ScentTrails performs as a technique, a broader study consisting of more sites and more tasks is necessary. Ideally, that would require a broader effort involving understanding the space of all possible Web tasks and sampling that space appropriately. Recently, Morrison et al. [2001] studied Web activities using the Georgia Tech WWW Surveys and constructed a taxonomy of the Web tasks found in that survey. With broader understandings of typical Web tasks, we will be able to study task effects more generally.

*5.1.3 Experimental Procedure.* Our experiment followed a within-subject randomized block design. Each of the twelve subjects performed each of the eight tasks exactly once. There are four interface conditions, so each subject used each interface twice. Thus, the number of trials for each interface condition was 24. The interface to be used was counter-balanced across tasks. For example, the first subject was asked to use ScentTrails for task 1, ShortScent for task 2, Browse for task 3, Search for task 4, and so on. The assignment of interface to task was varied across subjects to minimize the influence of learning effects on results.

Before the experiment, each subject was asked to observe as we performed a short demonstration of each of the interfaces available: Browse, Search, ScentTrails, and ShortScent. We made sure that all subjects were familiar with browsing and keyword searching. Each subject was then permitted a five minute training period in which to familiarize himself/herself with ScentTrails and ShortScent. The purpose of the training period was to make sure the subject understood and felt comfortable using these interfaces.

When using one of the ScentTrails, ShortScent, or Search interfaces, subjects formulated their own search queries using keywords with no restrictions, and they were permitted to revise their keywords or submit new search queries as many times as they wished. In the ScentTrails and ShortScent interfaces, search result pages used to compute link highlighting strengths were selected by the algorithm from pages within the www.xerox.com Web site. Browsing of highlighted documents began with the root page, with the ability to reformulate search keywords as desired. In the Search interface, pages internal to the www.xerox.com site were selected and displayed in a ranked list of search results, and users were permitted to browse freely and without restrictions, starting from the ranked result page, after performing each search.

We measured the time it took to complete each task. Tasks that took longer than five minutes to complete were interrupted to reduce the length of each user session and avoid user fatigue. (In most cases, when the user had not been able to complete the task within five minutes we observed that they had become frustrated and had lost hope of completing the task efficiently.) After each subject had finished all the trials, we conducted a short interview. The subject was asked which interface they preferred, and what criteria they used when deciding which link to click on. We also requested feedback on the ScentTrails interface.

### 5.2 Analysis

Figure 4 shows the task completion time results from the experiment. In Figure 4a, the average task completion time is shown for each task/interface pair, along with standard error bars. Recall from Section 5.1.2 that when selecting tasks we wanted to ensure a range of difficulty levels and include some tasks that are amenable to searching as well as others that are more amenable to browsing. Indeed, from Figure 4a we see that some tasks are more suited to searching (*e.g.*, task 1, 3, 6), and some to browsing (*e.g.*, tasks 5, 8). Furthermore, some tasks appear to be easier than others (*e.g.*, tasks 3, 6 versus tasks 4, 7).

In Figure 4b, for each interface, the average task completion time combined over all tasks is shown along with the standard error. On average, searching performed better than browsing, and ScentTrails performed better than both searching and browsing. Thus, the figure clearly shows that, using the same keyword entry capability, ScentTrails performs better than a simple full-text search. Recall that each trial was stopped after five minutes even if the task had not been completed by that time. In cases where the task was incomplete after the time limit the trial time was recorded as five minutes, so the average times should be interpreted with this effect in mind. To help understand the effect of limiting the trial times, Figure 5 reports the fraction of trials that were halted after five minutes with the task left incomplete. Note that only a small fraction of the trials for ScentTrails had to be interrupted due to the time limit compared with the other three interfaces. Therefore, Figure 4 and all subsequent analyses represent a conservative evaluation of ScentTrails relative to the other interfaces.

Fig. 4a. Average task completion time (minutes) for each task/interface pair, with standard error.

Fig. 4b. Average task completion time for each of the interfaces combined across tasks, with standard error.

Fig. 5. The fraction of trials that were stopped after five minutes with the task left incomplete, for each interface.

We now describe the analysis showing statistically significant differences between the interfaces. First, we performed a repeated measures analysis of variance (ANOVA) on performance time with interface as the within-subjects factor. The

ANOVA yielded a main effect of interface, $F(3,33)=6.088$, $p=0.002$. Overall, the interface condition had a statistically significant effect on the task performance.

Next, to determine the grouping and ordering of the interface conditions, we used paired t-tests to examine differences between interface conditions. We used an alpha level of .02 for these tests to account for multiple comparisons. Because we cut off performance time at 5 minutes, an event that occurred more often in the browsing and searching conditions, we felt that a more conservative alpha was not appropriate. The results show that ScentTrails differs significantly both from searching, $t(11)=-2.828$, $p=.016$, and from browsing, $t(11)=-4.275$, $p=.001$. (ScentTrails M=1.59 minutes, ShortScent M=2.35 minutes, search M=3.04 minutes, browse M=3.53 minutes.)

We wished to further test the hypothesis that Scent-based interfaces are quantitatively different from the searching and browsing interfaces by performing a Linear Contrast between the two Scent-based interfaces (ScentTrails and ShortScent) and the two traditional interfaces (search and browse). The result indicates a linear separation between the two groups, $t(33)=-3.843$, $p<0.001$. We therefore conclude that there is significant differentiation between the Scent interfaces and both searching and browsing.

The remaining question is how different the two Scent interfaces are from each other. A Linear Contrast test was not able to show conclusively that there is a linear separation, $t(33)=-1.567$, $p=0.1266$. Perhaps this result can be attributed to the fact that the test corpus we used in our experiment was somewhat shallow, *i.e.*, the length of paths from the root document to typical documents matching user queries did not generally exceed three or four. We believe that an expanded study on a variety of Web sites including ones with more deeply nested hierarchies would be desirable to conclusively test for quantitative differences between the ShortScent and ScentTrails interfaces.

### 5.3 Subject Interviews

*5.3.1 Interface Preference.* Ten out of twelve subjects, when interviewed after the experiment, stated that they preferred the Scent interfaces to both browsing and searching. The other two subjects declined to state a preference, citing lack of sufficient experience to make a judgment.

*5.3.2 Advantages of Combining Searching with Browsing.* During the interview, several subjects mentioned, without being explicitly asked, that the Scent interface permitted them to narrow down some aspects of the task with keywords, while honing in on other aspects by browsing. For example, the Xerox.com home page features two main sections: one with links to press releases and one with links to information about products. Our tasks required subjects to locate products. Using the search interface, subjects found it difficult to differentiate between products and press releases, since this context was embedded in the path to the pages. ScentTrails proved more successful. While the search cues sometimes indicated that some press releases and some products matched the search keywords equally well, the subjects tended to follow links from the products section. This demonstrates that, for the tasks we studied, users were able to consider browsing cues and search cues simultaneously to make intelligent decisions. Moreover, our quantitative analysis found that the additional algorithm and interface complexity of the ScentTrails technique did not detract from its effectiveness.

*5.3.3 Praise and Criticisms of Highlighting Method.* Several subjects mentioned that they liked the link highlighting because it made browsing more efficient. They tended to focus on the strongly highlighted links, and to choose among those links based on the browsing cues. This saved them time because they were able to narrow their focus quickly to a handful of links on each page. However, several of the subjects thought that, although link highlighting was useful, the specific method we used to highlight links, *i.e.*, increasing the font size of textual hyperlinks, could be improved upon. We discuss options for link highlighting next.

## 6. LINK HIGHLIGHTING METHODS

As mentioned earlier, there are many conceivable ways to implement link highlighting. We have chosen not to focus on the choice of highlighting method but rather on the algorithm for determining link highlighting strengths, and our experimental results show that our algorithm works well for the content and tasks we studied. That said, the choice of highlighting method is clearly an important consideration, and we provide some initial discussion of link highlighting options here, leaving further study as future work.

Although we found font size variation (as illustrated in Figure 1) to be for the most part a reasonable link highlighting method for the Web site we used in our experiments, it has at least two drawbacks that could become more severe with other Web sites. First, it tends to distort the formatting of the page, making the page more difficult to read. Second, since users generally do not know the original font sizes, it is ambiguous in some cases to what degree the font has been modified by the ScentTrails algorithm. For these reasons, font size variation for link highlighting may not be amenable to Web sites that rely heavily on tightly formatted layouts or use many different font sizes on a single page. Furthermore, the method of varying font size is obviously not applicable to graphical link anchors, and alternative methods are required for Web sites that make heavy use of graphical icons as link anchors.

Aside from increasing the font size, other link highlighting options for text include changing the text color or background color, animating the text (*e.g.*, blinking text), changing the font style, outlining the text, or some combination of these methods.[2] If the link anchor is an image or icon instead of text, we could apply methods more appropriate for the graphical domain. Since many Web sites mix textual and graphical link anchors, it is important to use a highlighting method that applies equally well and uniformly to both types of anchors.

Figure 6 suggests two candidate methods for use in highlighting both textual and graphical link anchors automatically: outlining links with lines of varying thickness, and varying the background color or degree of shading. Of these two methods, varying the background may be problematic for icons that lack an obvious "background" region that covers a reasonable area. In contrast, outlining appears to be broadly applicable to most types of icons (by using a rare color for outlines, this technique can be made resistant to icons that already include an outline). Also, the method of link outlining has the additional advantage that it does not require modifying image files, which can be computationally expensive. On the other hand, while outlining seems promising for highlighting small to medium sized icons, the technique may not provide adequate emphasis for very large graphical links whose border is well away from the main focal area. In the future we plan to perform experimental studies to determine good overall methods for automatic link highlighting.

Fig. 6. Some example link highlighting methods.

In practice, the best strategy may be to offer a range of automatic link highlighting options, so Web site designers who deploy a ScentTrails proxy are able select the highlighting method most amenable to their content and layouts. Finally, as an alternative to relying on automatic link highlighting, Web site designers could provide customized "style sheets" specifying how link anchors should appear under different degrees of highlighting.

## 7. CONCLUSIONS AND FUTURE WORK

[2] Woodruff et al. consider some methods for visually highlighting parts of Web pages in a different context [30].

In this paper we introduced ScentTrails, a novel method for finding information on the Web. ScentTrails combines browsing and searching to achieve the best of both worlds. Based on the concept of Information Scent [Chi et al. 2001], ScentTrails highlights hyperlinks according to the relevancy of each link to keywords entered by the user. In this way, search cues are layered on top of existing browsing cues to enable the user to locate pages matching complex information goals that are suited partially to searching and partially to browsing.

We evaluated ScentTrails by performing a preliminary user study, which yielded positive results. From our statistical analyses, we concluded that ScentTrails provides a way to locate information significantly more quickly than by either searching or browsing alone for the tasks and content we studied. Since ScentTrails combines the functionality of searching and browsing into a single interface, it should be suitable for a wide variety of information seeking scenarios.

As future work we plan to test the generalizability of our results by evaluating the effectiveness of the ScentTrails approach on other Web sites and with other information seeking tasks. We also intend to study the scalability of the ScentTrails algorithm to larger Web sites in terms of the costs to compute and store the scent conduit matrix and the latency incurred while accessing it at runtime. Additionally, as mentioned above, we intend to investigate algorithms for selective in-memory caching of the scent conduit matrix when the matrix is large. Finally, we plan to test scalability to large volumes of user requests, which can potentially be achieved by mirroring the ScentTrails proxy, since users are treated independently in our approach.

Another important topic for future work will be to investigate and compare different highlighting techniques for textual and graphical link anchors, as discussed earlier. Finally, we are also interested in applying the concept of ScentTrails to the domain of Web site analysis and tuning. ScentTrails can potentially be used to identify specific links whose browsing cues can be improved.

## ACKNOWLEDGMENTS

We thank Peter Pirolli and Pam Schraedley for their feedback. We also thank the anonymous reviewers for many helpful comments. We also thank the twelve subjects who volunteered to participate in our experiment. This research was funded in part by Office of Naval Research Contract No. N00014-96-0097 to Peter Pirolli and Stuart K. Card. Olston was supported by the PARC Summer Internship Program.

## REFERENCES


ANDERSON, J.R. 1983. A spreading activation theory of memory. *Verbal Learning and Verbal Behavior 22*, 261-295.

BARRETT, R., MAGLIO, P.P., AND KELLEM, D.C. 1997. How to personalize the Web. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, Atlanta, GA, March 1997, STEVEN PEMBERTON, Ed. ACM/Addison-Wesley, 75-82.

BATES, M.J. 1989. The design of browsing and berrypicking techniques for the on-line search interface. *Online Review 13(5)*, 407-431.

BELKIN, N. J. 1980. Anomalous states of knowledge as the basis for information retrieval. *Canadian Journal of Information Science 5*, 133-143.

BRUSILOVSKY, P. 1996. Methods and techniques of adaptive hypermedia. *User Modeling and User Adapted Interaction 6(2-3)*, 87-129.

CAMPBELL, C.S., AND MAGLIO, P.P. 1999. Facilitating navigation in information spaces: Road signs on the World Wide Web. *International Journal of Human-Computer Studies 50*, 309-327.

CHEN, M., HEARST, M., HONG, J., AND LIN, J. 1999. Cha-Cha: A system for organizing intranet search results. In *Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems*, Boulder, CO, October 1999.

CHI, E.H., PIROLLI, P., CHEN, K., AND PITKOW, J. 2001. Using information scent to model user information needs and actions on the Web. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, Seattle, WA, March-April 2001, 490-497.

DEVORE, J.L. 1987. Probability and statistics for engineering and the sciences, 2nd Edition, Brooks/Cole, Belmont, CA, 413-421.

EGAN, D.E., REMDE, J.R., GOMEZ, L.M., LANDAUER, T.K., EBERHARDT, J., AND LOCHBAUM, C. 1989. Formative design-evaluation of SuperBook. *ACM Transactions on Information Systems 7(1)*, 30-57.

FURNAS, G.W. 1997. Effective view navigation. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, Atlanta, GA, March 1997, STEVEN PEMBERTON, Ed. ACM/Addison-Wesley, 367-374.

FURNAS, G.W. AND RAUCH, S.J. 1998. Considerations for information environments and the NaviQue workspace. In *Proceedings of the Third ACM Conference on Digital Libraries*, Pittsburgh, PA, ACM Press, 79-88.

GREER, J.E. and PHILIP, T. 1997. Guided navigation through hyperspace. In *Proceedings of the Eighth World Conference on Artificial Intelligence in Education,* Kobe, Japan, August 1997.

GOLOVCHINSKY, G. 1997. Queries? Links? Is there a difference? In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, Atlanta, GA, March 1997, STEVEN PEMBERTON, Ed. ACM/Addison-Wesley, 407-414.

HERSOVICI, M., JACOVI, M., MAAREK, Y.S., PELLEG, D., SHTALHAIM, M., AND UR, S. 1998. The shark-search algorithm – An application: Tailored Web site mapping. In *Proceedings of the Seventh International World Wide Web Conference*, Brisbane, Australia. *Computer Networks 30(1-7)*, 317-326.

HUBERMAN, B.A., PIROLLI, P., PITKOW, J.E., AND LUKOSE, R.M. 1998. Strong regularities in World Wide Web surfing. *Science 280(5360)*, 95-97.

JOACHIMS, T., FREITAG, D., AND MITCHELL, T. 1997. WebWatcher: A tour guide for the World Wide Web. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, Nagoya, Japan, Morgan Kaufmann, August 1997, 770-777.

JUL, S. AND FURNAS, G. W. 1997. Navigation in electronic worlds: A CHI 97 workshop. *SIGCHI Bulletin 29(4)*.

LIEBERMAN, H. 1995. Letizia: An agent that assists Web browsing. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, Montreal, Canada, August 1995, Morgan Kaufmann, 924-929.

MAAREK, Y. S. AND SHAUL, I. Z. B. 1997. WebCutter: A system for dynamic and tailorable site mapping. In *Proceedings of the Sixth International World Wide Web Conference,* Santa Clara, California. *Computer Networks 29(8-13)*, 1269-1279.

MANBER, U., SMITH, M., AND GOPAL, B. 1997. Webglimpse: Combining browsing and searching. In *Proceedings of the Usenix Technical Conference,* Los Angeles, CA, January 1997.

MLADENIC, D. 2001. Using text learning to help Web browsing. In *Proceedings of the Ninth International Conference on Human-Computer Interaction*, New Orleans, LA, August 2001.

MORRISON, J. B., PIROLLI, P. AND CARD, S. K. 2001. A Taxonomic Analysis of What World Wide Web Activities Significantly Impact People's Decisions and Actions. In *Proceedings of the ACM Conference on Human Factors in Computing Systems,* Seattle, WA, 163-164.

PARK, J., KIM, J. 2000. Effects of contextual navigation Aids on browsing diverse Web systems. In *Proceedings of the ACM Conference on Human Factors in Computing Systems,* The Hague, The Netherlands, 257-264.

PERKOWITZ, M. AND ETZIONI, O. 1999. Towards adaptive Web sites: Conceptual framework and case study. In *Proceedings of the Eighth International World Wide Web Conference,* Toronto, Canada. *Computer Networks 31(11-16)*, 1245-1258.

PIROLLI, P. 1997. Computational models of information scent-following in a very large browsable text collection. In *Proceedings of the ACM Conference on Human Factors in Computing Systems,* Atlanta, GA, 3-10.

PIROLLI, P. AND CARD, S.K. 1999. Information Foraging. *Psychological Review, 106(4),* 643-675.

SALTON, G. AND YANG, C.S. 1973. On the specification of term values in automatic indexing. *Documentation 29,* 351-372.

WITTENBURG, K. AND SIGMAN, E. 1997. Integration of browsing, searching, and filtering in an applet for Web information access. In *Proceedings of the ACM Conference on Human Factors in Computing Systems,* Atlanta, GA.

WOODRUFF, A., FAULRING, A., ROSENHOLTZ, R., MORRISON, J., AND PIROLLI, P. 2001. Using thumbnails to search the Web. In *Proceedings of the ACM Conference on Human Factors in Computing Systems,* Seattle, WA, 198-205.

Yahoo! Inc. 2003. Yahoo! http://www.yahoo.com.

YAN, T.W., JACOBSEN, M., GARCIA-MOLINA, H., AND DAYAL, U. 1996. From user access patterns to dynamic hypertext linking. In *Proceedings of the Fifth International World Wide Web Conference,* Paris, France. *Computer Networks 28(7-11)*, 1007-1014.

Received May 2002; revised December 2002; accepted March 2003.

**Departmental and Production Copiers**
(60 & up Copies per Minute; Volume above 75,000 Copies per Month)

**5665 Copier:** 60 copies/min. Space efficient design, highlight color, versatile and feature rich with extensive sorter finishing options.

**5065 Copier:** 62 copies/min. Zoom R/E, up to 171"x22" originals & 11"x17" copies, feeder, duplex, other high end features.

**5365 Copier:** 62 copies/min. 100 sheet feeder, zoom R/E, up to 171"x22" originals & 11"x17" copies, duplex, other high end features.

**Document Centre 265 Digital Copier:** 65 copies/min. Scans your originals only once, and then prints as many copies as you need. Duplex, zoom reduce/enlarge.

**5385 Copier:** 80 copies/min. Up to 171"x22" originals & copies, 100 sheet feeder, highlight color, image editing, many features & options.

**5680 Copier:** 80 copies/min. Space efficient design, 100 sheet feeder, auto insertion of covers & transparency slipsheets, collating, stapling.

**5388 Copier:** 92 pages/min. Updated and enhanced design of the popular 1090 copier. Wide range of capabilities and capacities.

**5892 Copier:** 92 pages/min. Compact size, photo mode, background suppression, and 100-sheet universal document feeder. Easy-to-use control panel with message display and color graphics.

Figure 1

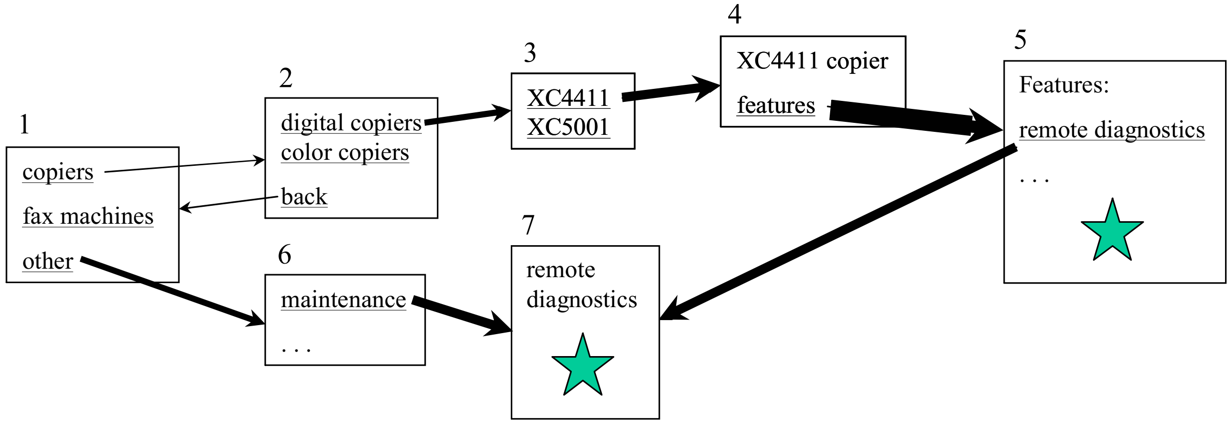
1
copiers
fax machines
other
2
digital copiers
color copiers
back
3
XC4411
XC5001
4
XC4411 copier
features
5
Features:
remote diagnostics
. . .
6
maintenance
. . .
7
remote
diagnostics

Figure 2

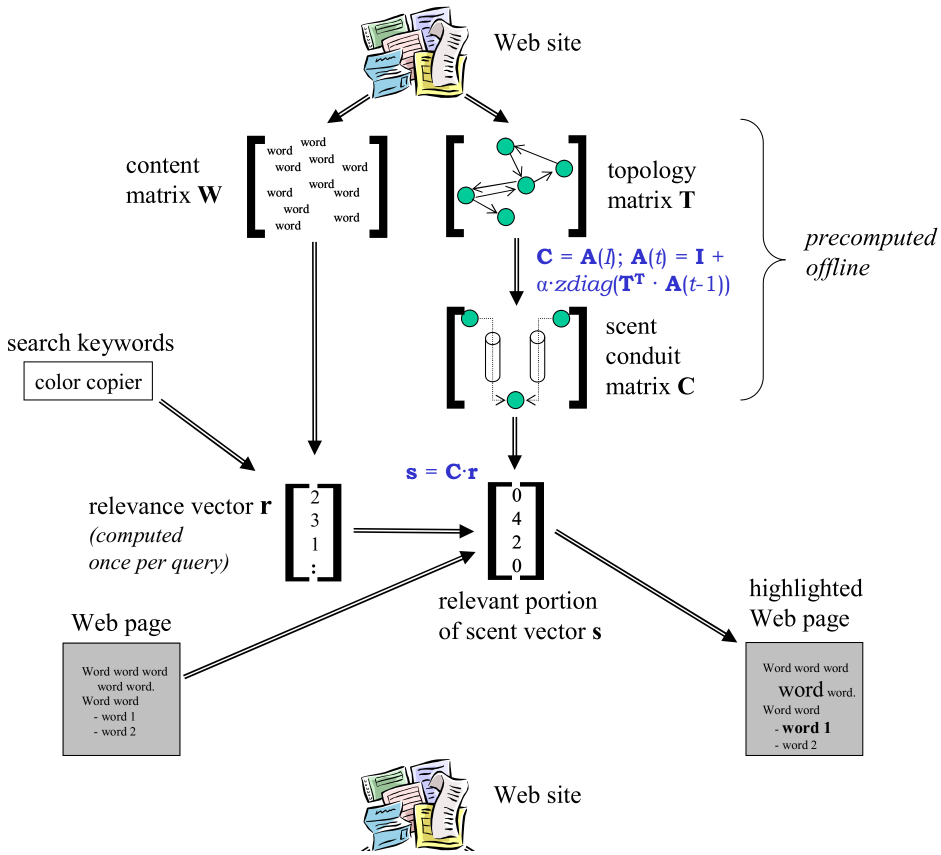
Web site
content
matrix W
word
word
word
word
word
word
word
word
word
word
word
topology
matrix T
C = A(l); A(t) = I + α·zdiag(T^T · A(t-1))
precomputed
offline
search keywords
color copier
scent
conduit
matrix C
s = C·r
relevance vector r
(computed
once per query)
2
3
1
:
0
4
2
0
relevant portion
of scent vector s
highlighted
Web page
Web page
Word word word
word word.
Word word
- word 1
- word 2
Word word word
word word.
Word word
- word 1
- word 2
Web site

Figure 3

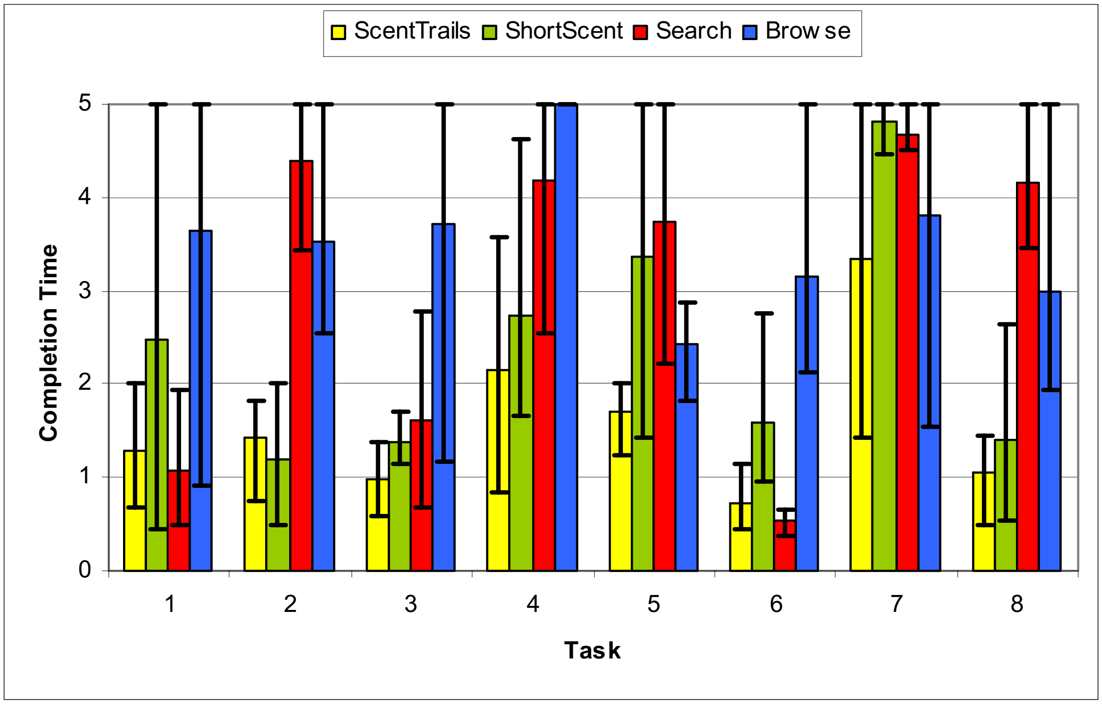


Figure 4a

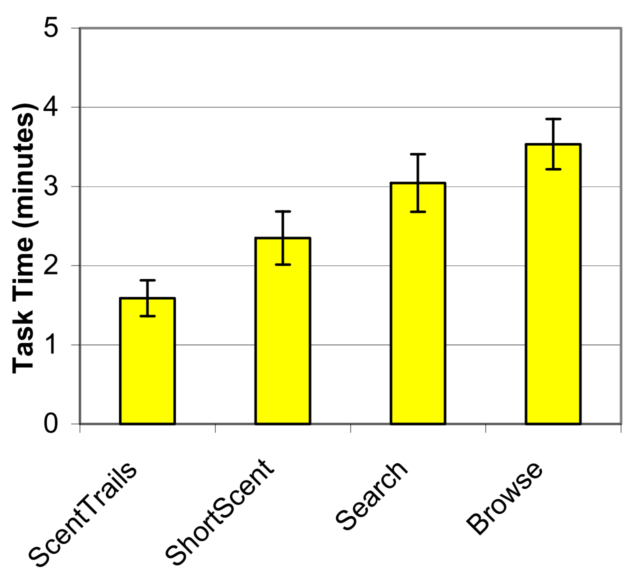


Figure 4b

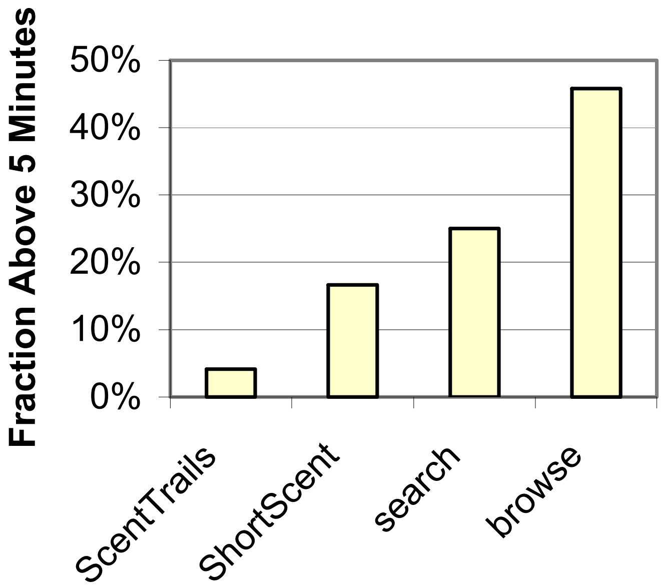
Fraction Above 5 Minutes
50%
40%
30%
20%
10%
0%
ScentTrails
ShortScent
search
browse

Figure 5

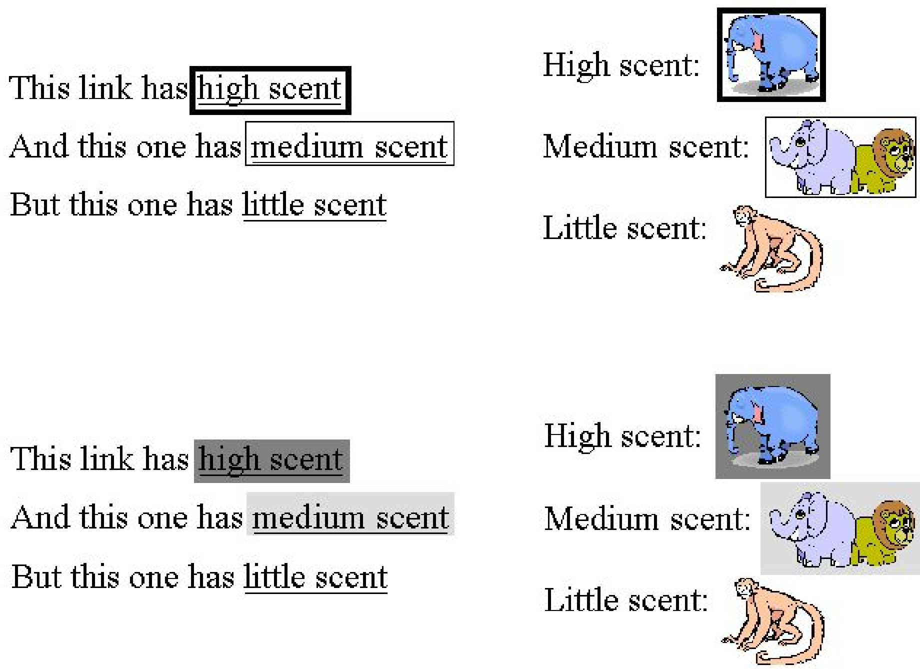
This link has high scent
And this one has medium scent
But this one has little scent
High scent:
Medium scent:
Little scent:
This link has high scent
And this one has medium scent
But this one has little scent
High scent:
Medium scent:
Little scent:

Figure 6