Optimizing Search by Showing Results In Context

Susan Dumais and Edward Cutrell
Microsoft Research
One Microsoft Way
Redmond, WA 98052
[sdumais|cutrell]@microsoft.com

Hao Chen
Computer Science Division
University of California
Berkeley, CA 94720-1776
hchen@cs.berkeley.edu

ABSTRACT
We developed and evaluated seven interfaces for integrating semantic category information with Web search results. List interfaces were based on the familiar ranked-listing of search results, sometimes augmented with a category name for each result. Category interfaces also showed page titles and/or category names, but re-organized the search results so that items in the same category were grouped together visually. Our user studies show that all Category interfaces were more effective than List interfaces even when lists were augmented with category names for each result. The best category performance was obtained when both category names and individual page titles were presented. Either alone is better than a list presentation, but both together provide the most effective means for allowing users to quickly examining search results. These results provide a better understanding of the perceptual and cognitive factors underlying the advantage of category groupings and provide some practical guidance to Web search interface designers.

Keywords
User Interface, World Wide Web, Search, User Study, Usability, Text Categorization, Focus-In-Context

INTRODUCTION
Web search systems (e.g., Alta Vista, Google, MSNSearch) typically return a ranked list of pages in response to a user's search request. Such lists can be very long and daunting. A query on something seemingly specific like "CHI 2001" returned 540,000 matches in one popular search engine and 453,000 in another. More important than the absolute number of matches is the fact that pages on different topics are intermixed in the returned list, so the user has to sift through a long undifferentiated list to find pages of interest. Pages on the ACM CHI 2001 conference are intermixed with pages on the Delta Epsilon Chi 2001 meeting, Childrens Hope International (abbreviated CHI) 2001 calendar, the University of Loyola Chi(cago) 2001 basketball schedule, Tai Chi 2001 events, and so on.

SIGCHI'01, March 31-April 4, 2001, Seattle, WA, USA.

Web directories (e.g., Yahoo!, LookSmart, Snap, Open Directory) employ human editors to categorize or tag Web pages. A similar approach has been used by librarians for decades in systems like Dewey Decimal classification, Library of Congress subject headings, Medical subject headings, etc. The resulting category structures can be browsed directly, or used to help organize search results. Since this approach depends on human tagging, coverage is limited. There was, for example, no content for the query "CHI 2001" in most of the Web directories. Even when there is content in a directory or when it is automatically tagged as in our work or Northern Light, how best to integrate specific search results with the overall category structure is unclear. Several alternative approaches exist in the field, but there is little empirical evidence to guide the design of systems for showing search results in context.

The research reported in this paper develops and evaluates a variety of new interfaces for combining specific search results with information about category structure. In addition, automatic text classification techniques are used to extend the category tags provided by human editors to the broader coverage available with standard search engines.

RELATED WORK

Text Classification
By text classification we mean the ability to assign category or class labels to new documents. Statistical techniques can be used to learn a model for each category based on a labeled set of training documents with known category labels. The model can then be applied to new documents to determine their categories. A wide variety of text retrieval and machine learning techniques can be used to build category models (see Dumais et al. [5] for a review). Chakrabarti et al. [1], Dumais and Chen [3], Stata et al. [12] and others have developed automatic classifiers for web pages using content from Web directories as training data. For present purposes, the important aspect of this work is that any new web content can be tagged, thus dramatically extending the reach of directory services. How best to present the resulting category information to help users winnow through a large set of search results is the focus of experiments described in this paper.

Combining Search Results and Category Structure
A number of web directory services add some kind of category information to search results. Yahoo! [17], Snap

[16], LookSmart [14], Open Directory [13], and Stata et al. [12] all show the human-assigned category labels associated with each retrieved page. The category information is provided as part of the summary of the web page, but the main organization of the search results is a ranked list. Yahoo! does some grouping of search results, but only at the lowest category level. Even here there is little global information available about the category structure or about the distribution of search results across categories. There is, for example, no way to quickly see that the search results fall into five different categories or that the majority of the matches fall into a single top-level category. Northern Light [15] provides 'Custom Folders' to organize search results. The folders are automatically created according to several dimensions – subject, type (e.g., press releases, product reviews, resumes, recipies), source (e.g., commercial sites, personal pages, magazines, encyclopedias, databases), and language. Individual categories can be explored one at a time. But, again no global information is provided about the category structure or about the distribution of search results across categories.

Few studies have evaluated the effectiveness of different interfaces for organizing search results. Egan et al. [6] compared two interfaces for accessing chemistry journal articles. SuperBook used a hierarchy of categories from chemical abstracts as a kind of table of contents and posted the number of search hits in each category against this static structure. PixLook used a post coordination technique to rank results from a Boolean retrieval system. Browsing accuracy was higher for SuperBook than PixLook. General search accuracy and search times were about the same for the two interfaces, with SuperBook showing a small but unreliable advantage. These results are encouraging, but different text pre-processing techniques, search algorithms, and display formats were used in the two conditions so it is difficult to compare precisely. Pratt et al. [11] compared DynaCat, a tool that automatically categorized results using knowledge of query types and a model of domain terminology, with a ranked list and clustering. Participants liked DynaCat's category organization of search results and found somewhat more new answers using it, but the results were not statistically reliable, presumably because there were only 15 participants and 3 queries in the experiment.

More recently, Chen and Dumais [2] compared SWISH, a category interface, with a traditional ranked list interface for presenting web search results. In their experiment search results were automatically categorized using text classification techniques, and pages in the same category were grouped together. They found large and reliable advantages for the category organization. Participants liked the category interface much better than the list interface, and they were 50% faster in finding information that was organized into categories.

The category presentations used in SuperBook, DynaCat and SWISH all use a kind of focus-plus-context or detail-plus-overview technique (Furnas [7], Mackinlay et al. [9], Green et al. [8]). Specific search results (focus) are shown in the context of a category structure (context). Since results can fall into multiple categories, there are often multiple foci of interest.

Understanding the Advantages of Category Structure

In this paper we developed and evaluated a series of new interfaces in order to better understand the perceptual and cognitive factors underlying the large category advantage reported by Chen and Dumais and hinted at in earlier work. There were many differences between the category and list conditions they tested (e.g., the category condition grouped items perceptually, had category labels for each item, etc.) and it is not clear which of these is most important. Our approach was to provide additional semantic category contexts for the list interface, and to remove aspects of the context from the category interface to determine what interface elements were most important in searching.

INTERFACE CONDITIONS

Basic Category and List Conditions

Figure 1 shows the two presentation conditions used by Chen and Dumais. In the *Category* interface, search results were organized into hierarchical categories as shown in the left pane. Under each category, the best matching web pages in that category were listed. Additional pages in the category could be seen on demand by category expansion. To show both category context and individual results in limited screen space, only the title of each page was shown. The summary of each page was available as hover text (i.e., when the user hovered over a title, the summary was displayed). The subcategories for each category were also available as hover text.

The *List* interface, shown in the right pane, was similar to current systems for displaying web search results. For comparability to the Category condition, only titles were shown initially with summaries available as hover text. Additional matches could be seen by expanding the list.

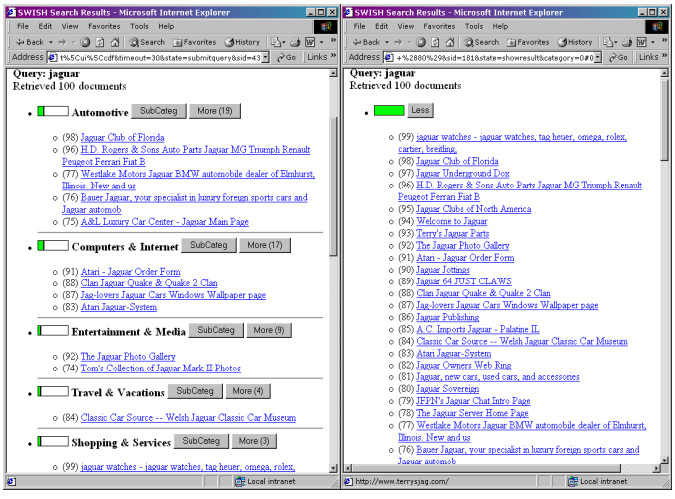


Figure 1. Interface conditions used by Chen and Dumais (2000).

Adding Context to the List Interface

In contrast to the Category interface, the List interface had very little contextual information associated with the returned page titles. We explored two methods for adding contextual information to the lists. The first approach presented summaries inline with the page titles; the second approach added the category name to each list result. Screenshots of all interfaces used are shown in Figure 5.

Summaries Inline (Figures 5b and 5e).

One reason for the advantage of the Category interface might be that the category labels provided an easy way for users to disambiguate ambiguous page titles. For example, a page entitled *Kenny Rogers home page* could refer to the singer, the baseball player, or others. The context given by the category names could provide quick disambiguation. There were some indications in the Chen and Dumais study that this was the case. In the List condition, participants looked at 54% more summaries (4.60 vs. 2.99) and looked at 15% more pages (1.41 vs. 1.23), suggesting that the titles alone were often not good indicators of the relevance of the page to the query. Adding page summaries inline could provide a kind of contextual disambiguation. The additional context comes at the cost of more scrolling to view the same number of results. This interface is shown in Figure 5b. To fully explore this interface element, we also added summaries inline in the category condition. This condition was particularly interesting because it may not provide all that much additional contextual information but requires more scrolling. This interface is shown in Figure 5e.

List Plus Category Names (Figure 5c).

Another way to add contextual information to the list interface was simply to add the category name to each item in a standard ranked list. This interface is shown in Figure 5c. This is currently done by many web directory services (e.g., Yahoo! [17], Snap [16], LookSmart [14], Open Directory [13]). The category name for each item was bold to make it stand out and facilitate quick scanning. Note that in this list augmentation we provide exactly the same information that is present in the Category interface (Figure 5e), but we present it in a very different format.

Removing Context from the Category Interface

We explored two methods for removing contextual information from the Category interface. The first technique removed the category names while the second removed the individual page titles.

Removing Category Names (Figure 5f).

This operated exactly the same as the basic Category interface except that no category names were shown above the groupings (e.g., *Automobile* or *Computers & Internet* were omitted). The search results were still grouped by category, and users could see more items by expanding the groups. However, there were no category names associated with each group. This interface is shown in Figure 5f.

Removing Page Titles, Browsing (Figure 5g).

Finally, we explored removing the page titles from the Category interface. In this presentation we displayed only the category names initially, with page titles available only after expansion of the top-level categories. This yielded a browsing interface, and allowed us to explore how much example instances (page titles) helped to disambiguate the category names. This interface is shown in Figure 5g.

USER STUDIES

Our basic experimental procedure followed that developed by Chen and Dumais [2]. Category tags were automatically assigned to each search result using text classification techniques described in Dumais and Chen [3]. In a series of four experiments, we examined five new interfaces in addition to the basic Category and List interfaces explored by Chen and Dumais. The interfaces examined in each experiment were as follows: Experiment 1 (Category Hover 5d, List Hover 5a), Experiment 2 (replication of Category Hover 5d, List Inline 5b), Experiment 3 (Category Inline 5e, Category without Category Names 5f), Experiment 4 (Category Browse 5g, List with Category Names 5c).

Methods

Participants

Participants were adult residents of the Seattle area recruited by the Microsoft usability labs. All participants had intermediate web ability and represented a range of ages, backgrounds, jobs and education levels. Between 18 and 20 people participated in each experiment. Almost all participants used the Web every week (74 of 76), and most searched for information on the Web every week (66 of 74).

Procedure

Each experiment was divided into two halves. Participants used one interface in the first half and another interface in the second. The order of presentation was counterbalanced across subjects. Users performed 15 web search tasks in each half, for a total of 30 search tasks. At the end of the experiment, participants completed an online ques-tionnaire. The total time for the experiment was about 2 hours.

During the experiment, participants worked with a three-window experimental display (Fig. 2). A small *control window* on the top showed the task, the query keywords, and a timer. The *search results* were displayed in the left window either as a list or grouped into categories, depending on condition. When participants clicked on a hyperlink, the *Web page* was shown in the right window. The relevant web page had to be visible in the right window in order for the subject to indicate they had found the answer and end a trial.

When participants found an answer they clicked “Found It!” in the control window. If they could not find the answer, they clicked “Give Up”. There was a timer that alerted users after five minutes had passed, and they could continue searching or move on to the next task. We logged a variety of user interactions such as hovering over a hyperlink to read the

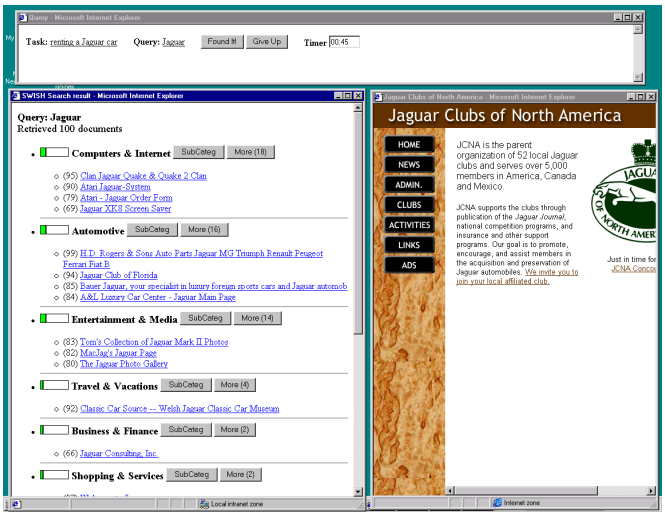


Figure 2. Screen of the User Study

summary, clicking on a hyperlink to read the page, and expanding or collapsing search results.

Search Tasks

Thirty search tasks were selected from a broad range of topics, including sports, movies, travel, news, computers, literature, automotive, and local interest. Example tasks included: *Find the homepage for the band They Might Be Giants*, and *Find profiles of the women of NASA*. Each task had an answer in the returned pages, as judged by the experimenters ahead of time and verified by participants during the experiment. The top 100 results for each query were presented to participants. The top 20 search results were available with no expansion required, although scrolling was sometimes needed. The tasks varied in difficulty – 17 had answers in the top 20 items returned, and 13 had answers between ranks 21 and 100. To ensure that results from different participants were comparable, we fixed the keywords for each query, and cached the search results before the experiments so that each participant received the same results for the same query. The actual following of links to examine the web pages was done live.

All participants performed the same 30 search tasks. They used one interface for the first 15 tasks and another for the remaining 15 tasks. The order in which participants saw the interfaces was counterbalanced across participants (except for the condition involving no category names which was always shown first). Queries were also counterbalanced.

Results

The main independent variable in all experiments was the interface used. Some interface comparisons could be made within subjects (because the same participant experienced both interfaces being compared), but most were analyzed as between subjects variables. We analyzed both subjective questionnaire measures and objective measures including search time, accuracy, interactions with the interface such as hovering, and which web pages were displayed. The focus here is mainly on overall search time, supplemented by other measures where appropriate.

Accuracy/GiveUp

We looked at both a strict scoring criterion in which participants had to agree with our assessments of the relevance of a page, and a liberal scoring criterion in which any answer participants judged as relevant was counted as such. There were few differences, so we used the liberal criterion in all the search time analyses. Participants were allowed to give up at any time during a trial if they could not find an answer. There were no significant differences in the number of queries on which participants gave up for any interface style. On average participants gave up on less than 1 of 15 queries per condition.

Search Time

Mean log search times were used in these analyses to normalize the common skewing and variability associated with response time data. Figure 3 shows the log means associated with each condition. Note that each column also shows the mean search times (in seconds) because these are easier to understand than log mean time. Relationships between the search times for each interface are similar for both formulations.

The first analysis explored the addition of inline summaries to each interface as compared to summaries presented in hover text -- see the first two columns in each part of Figure 3: Cat Hover, Cat Inline, List Hover, List Inline. In addition, because we had approximately equal numbers of male and female participants, we also looked to see if there were any gender differences. We performed a 2 (List vs. Category) x 2 (Hover vs. Inline) x 2 (Gender) ANOVA. The Category interface was significantly faster than the List interface, $F(1,86)=38.1$, $p<0.01$ (see Figure 3). In addition, there was a borderline significant effect for summary condition: Inline summaries were faster than Hover summaries, $F(1,86)=3.5$, $p<0.06$. There was no significant effect for gender or any significant interaction. This analysis revealed two important points: First, that the Category interface continued to be faster than the List interface regardless of how title summaries were presented; and second, that inline summaries improved performance on *both* the List and Category interfaces. This is particularly interesting because one might expect users to be slower due to the additional scrolling inline summaries require. We suggest that this scrolling time is offset by the cognitive effort required to decide which items to hover over for additional information.

Our second attempt to improve performance in the List interface entailed including category names with each returned item. As seen in Figure 5c, this was identical with the List Inline interface, with the addition of bolded category names. We performed two t-tests, first comparing this interface to the List Inline interface and then to the Category Inline interface. The first of these showed that the addition of category names yielded no improvement in performance over the normal List interface, $t(38)=1.04$, NS. The second test

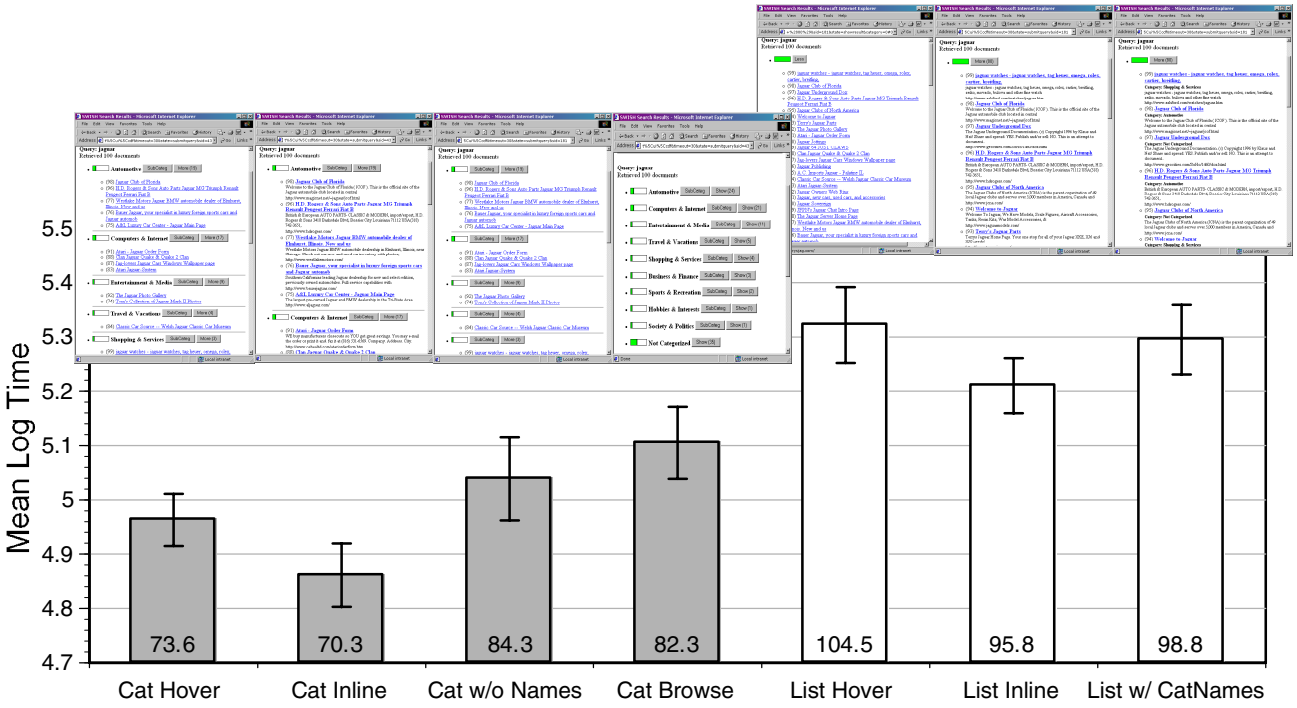


Figure 3. Mean log time to complete tasks for each condition (± standard error about the mean). Mean time in seconds appears at the bottom of each column, for easier interpretation of the magnitude of the differences.

mirrored this conclusion, showing that the Category interface continued to be faster than the List interface, even with the addition of category names to the list, $t(36)=5.0$, $p<0.01$. See Figure 3.

Since neither attempt to add context to the List interface improved performance relative to the Category interface, we wanted to explore what elements of the category interface made it superior. We did this by systematically eliminating sources of contextual information that might be helping users. The first attempt was to remove the category names from the interface. In this condition, page titles were still grouped as in the normal Category interface, but no category names were presented above the groupings (see Figure 5f). To further remove contextual information, page summaries were presented in hover text (as opposed to inline). A t-test comparing this degraded interface to the Category Hover interface (the comparable condition) showed no significant difference between the two, $t(31)=0.84$, NS. Moreover, users remained significantly faster with this degraded category interface than when using the List interface, $t(34)=2.73$, $p<0.01$.

The final attempt to remove contextual information from the Category interface removed the page titles from the results. This allows us to determine how much example instances (page titles) helped to disambiguate the category names. This interface returned only the top-level category names initially. Page titles with inline summaries were available only after expansion of this top level. We called this condition a "browsing" interface (see Figure 5g). This interface was compared with the Category Inline interface (the comparable condition). We found that the browsing interface was significantly slower, $t(36)=2.73$, $p<0.01$. The same participants who used this browsing interface also used the List with Category Names interface described above (Figure 5c). This allowed us to perform a paired-sample t-test comparing these two interfaces. These users were still faster using the browsing interface than using the List with Category Names interface, $t(19)=2.69$, $p<0.02$. Thus, while the browsing interface degraded performance relative to the optimal Category interface, it was still superior to the List interface.

Figure 3 summarizes the principal findings. Even when using degraded Category interfaces, users still completed searches faster than when they used List interfaces. This was true even when category names were included in the List interface. The addition of inline summaries improved performance in both conditions, despite the cost of additional scrolling. Surprisingly, the removal of category names from the category did not significantly hurt performance.

Individual search task results

There were large individual differences in search times, ranging from a mean of 36 to 176 seconds per query for different participants and interfaces. Similarly, there were large differences across search tasks. As noted above, 17 of the queries issued had answers in the top 20, while 13 had answers between ranks 21 and 100. We performed a 2 (List vs. Category) x 2 (Hover vs. Inline) x 2 (Top 20 vs. Lower) ANOVA to examine whether performance with different interfaces might be affected by the difficulty of the queries. Again, we found that the Category interface was significantly

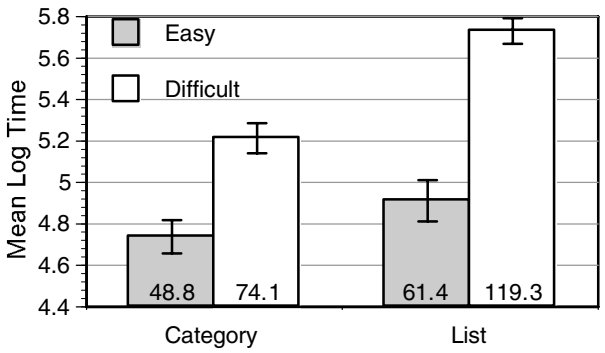


Figure 4. Mean log time to complete tasks for easy and difficult queries for each interface type.

faster than the List interface, $F(1,142)=16.1$, $p<0.01$. Unsurprisingly, we also found that users were much faster for queries with answers in the top 20 than for answers further down in the rank list, $F(1,142)=54.4$, $p<0.01$. However, there was also a borderline significant interaction between the interface used and the difficulty of the query, $F(1,142)=3.5$, $p<0.06$. For easy queries (answers in the top 20), the effect of the interface is somewhat muted. But when the answer was further down the list, the List interface was dramatically slower (see Figure 4). There were a few easy queries where the List interface was also bad, and these were typically associated with poor page titles. For example, one query asked participants to find *the home page for the band They Might Be Giants*. Although this page was the second ranked in the List interface, the target page was titled "TMBG" and many users skipped over it. The List interface was particularly susceptible to this kind of problem. There were surprisingly few queries that were more difficult with the Category interface. These were typically due to ambiguous categorization. For example, a query asking for *the Seattle Weekly's web site* (a local news and entertainment weekly) proved more difficult in the Category interface than the List interface probably because participants looked in two potentially relevant cate-gories (*Entertainment & Media*, and *Society & Politics*).

Interaction style

We measured the number of hovering, page viewing and expansion actions participants performed in the course of finding the answers. We found that significantly more *hover summaries were viewed* in the List than the Category conditions (4.6 vs. 2.8 summaries per query, $t(54)=4.7$, $p<0.01$, Figs. 5a & 5d). Participants also viewed significantly more summaries in the Category without Category Names condition than in the normal Category interface (4.2 vs. 2.8 summaries per query, $t(54)=3.2$, $p<0.01$, Figures 5d & 5f). This suggests that participants were using the summary to add contextual information when they did not have good category information. The *number of links followed* showed this same pattern of effects. More pages were viewed in the List than the Category condition (1.41 vs. 1.23, $t(54)=2.16$, $p<.02$) and in the Category without Category Names than the Category condition (1.58 vs. 1.23, $t(54)=3.30$, $p<0.01$). Additional viewing of hover summaries and link following appears to compensate for the lack of context. However, these simple low-level operations alone do not predict search time. There were, for example, the same number of links followed in the List Inline and Category Inline conditions, but search time was reliably faster in the Category Inline condition. We believe this is due to the perceptual grouping of related results, but more detailed measurements involving eye movements would be needed to verify this.

Subjective questionnaire measures

After the experiment, participants completed a brief online questionnaire. The questionnaire covered prior experience with Web searching, ratings of the two interfaces (on a 7-point scale), and open-ended questions about the best and worst aspects of each interface. Participants almost unanimously preferred the Category to List interface, mirroring their performance data. Mean ratings about the overall quality of the interface (averaged over 5 individual questions) were significantly higher for the Category conditions than the List conditions (6.00 and 4.26 respectively, $p<<0.01$). There was one interesting dissociation between subjective preference and search time data. Search times for the Category without Names and Category Browse conditions were roughly the same, but users disliked the Category without Names interface (mean ratings, 4.56 and 5.86, respectively).

SUMMARY AND CONCLUSION

We developed and evaluated seven different interfaces for structuring search results using category information. The results provide a better understanding of the perceptual and cognitive factors underlying the advantage of some category organizations, compared with linear lists for presenting search results.

In all cases, Category interfaces were faster than List interfaces. This was true even when we added Category Names and Inline Summaries to the List presentation, and when we degraded the Category presentation by removing Category Names or Page Titles. Interestingly, the List with Category Names interface contains the same information as the Category interface (all the individual results along with a category name for each), but performance is much slower with the list. How the category information is presented is the key to its success. The Category with Inline Summaries and the List with Category Names interfaces both contain focus (search results) *plus* context (category names). However, only the Category condition contains the focus *in* the context, and this appears to be critical for success in this search task. Nygren's work [9] suggests that spatial grouping is an important cue used by skilled searchers, and our Category interfaces provide this. It is interesting to note that many web directories present category information for each result in a list, but do not show the results in the context of the category structure. Nor do they present the same kind of high-level view we do showing the distribution of search results across category.

The best performance in the Category interfaces is achieved when both category names and page titles are available. Either alone works better than any list presentation, but the combination of focus in context is the most effective. It seems intuitive that category names can help users quickly focus in on areas of interest without having to examine individual page titles. What may be less apparent is that individual page titles can help disambiguate category names in a browsing interface. Are newspapers classified under *Society & Politics* or under *Entertainment & Media*? The answer is quite clear if specific results are shown in the context of the category names. This result is like that reported some time ago by Dumais and Landauer [4] where they found that both names and examples were the best way to describe Yellow Page categories. Interestingly, while many web directory services show category matches, none show examples of pages in each.

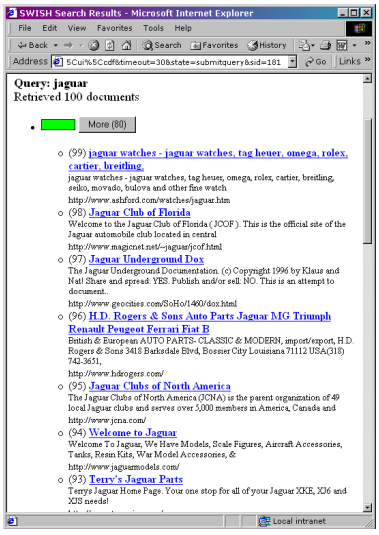
Another finding of interest for design is that Inline summaries were more effective than summaries presented as Hover text. This effect held for both the List and Category interfaces. In spite of the fact that more scrolling was required and some category context was missing when summaries were presented inline, participants were still faster than when they were required to hover to see more details. Apparently the cognitive costs of deciding which title to examine in more detail and the physical costs of pointing to it outweigh the additional scrolling required.

There are a number of interesting directions for future research. One direction involves how best to describe and present categories. In our experiments, categories were ordered by the number of matching pages but perhaps a consistent order would be better. Also, search results within a category were presented in the best match order, but perhaps presenting prototypical instances of each category (as determined by text classifier output) would help users to more quickly understand what is in each category. Another direction of interest would be to explore alternative techniques for visual grouping. We know that spatial grouping works, and that simple visual category descriptors do not, but what about iconic or color coding? Finally, one could explore techniques for explicitly refining queries (our grouping of results is a kind of implicit refinement).

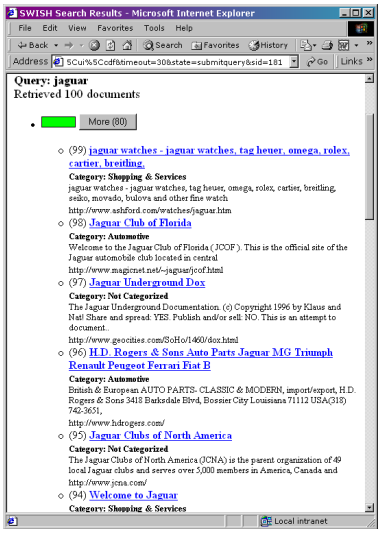
REFERENCES

1. Chakrabarti, S., Dom, B., Agrawal, R., and Raghavan, P. Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies. *The VLDB Journal 7*, 1998, 163-178.
2. Chen, H. and Dumais, S. T. Bringing order to the Web: Automatically categorizing search results. In *Proceedings of ACM SIGCHI Conference on Human Factors in Computing Systems (CHI),* 2000, 145-152.
3. Dumais, S. T. and Chen, H. Hierarchical classification of web content. In *Proceedings of SIGIR'2000*, 256-263.
4. Dumais, S. T. and Landauer, T. K. Describing categories of objects for menu retrieval systems. *Behavioral Research Methods, Instruments and Computers*, 1984, *16(2),* 242-248.
5. Dumais, S. T., Platt, J., Heckerman, D. and Sahami, M. Inductive learning algorithms and representations for text categorization. In *Proceedings of ACM-CIKM98*, Nov. 1998, 148-155.
6. Egan, D. E., Lesk, M. E., Ketchum, R. D., Lochbaum, C. C., Remde, J. R., Littman, M. and Landauer, T. K. Hypertext for the electronic library? CORE sample results. In *Proceedings of Hypertext'91*, 299-312.
7. Furnas, G. W. Generalized fisheye views. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI),*1986, 16-23.
8. Green, S., Marchionini, G., Plaisant, C., and Shneiderman, B., Previews and overviews in digital libraries: Designing surrogates to support visual information-seeking, *Journal of the American Society for Information Science 51*(3), March 2000, 380-393.
9. Mackinlay, J. D., Robertson, G. G., and Card, S. K. The perspective wall: Detail plus context smoothly integrated. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI),* 1991, 173-179.
10. Nygren, E. "Between the clicks": Skilled users scanning of pages. In *Proceedings of Designing for the Web: Empirical studies*, October 1996.
11. Pratt, W., Hearst, M. and Fagan, L. A knowledge-based approach to organizing retrieved documents. In *Proceedings of AAAI/IAAI, 1999, 80-85.*
12. Stata, R, Bharat, K. and Maghhout, F. The term vector database: Fast access to indexing terms for web pages. In *Proceedings of WWW9, 2000, 247-256.*
13. http://search.dmoz.org/
14. http://www.looksmart.com/
15. http://www.northernlight.com/
16. http://www.snap.com/
17. http://www.yahoo.com/

a. List with hover summary.

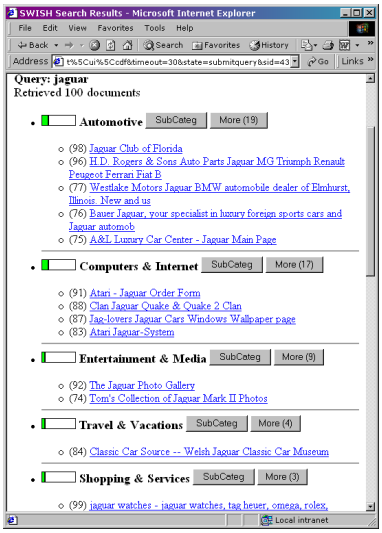


b. List with summary inline.

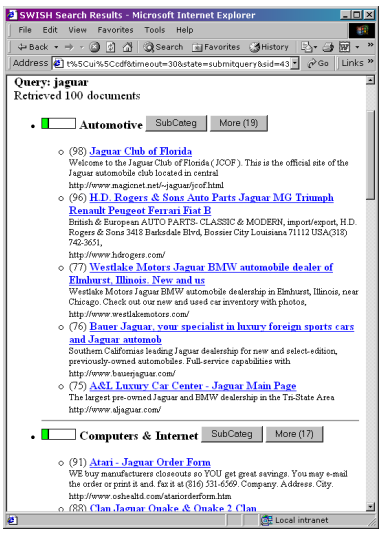


c. List with category names.

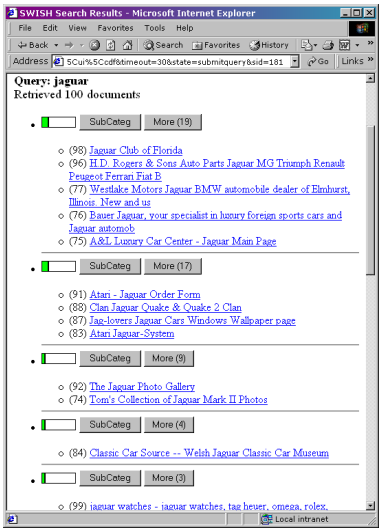
d. Category with hover summary.



e. Category with summary inline.



f. Category with no category names.



g. Category with no page titles (browse).

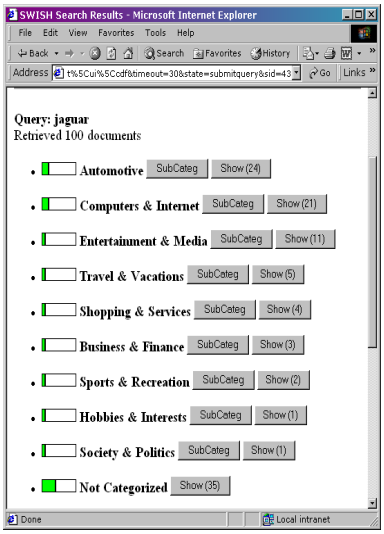


Figure 5. Screen shots of each UI condition.