Information Hierarchies [*]

Ellen Spertus
Concurrent VLSI Architecture Group
NE43-630
ellens@ai.mit.edu

Introduction

The advent and growth of the World-Wide Web present a new challenge to information providers: building structures that allow users to easily find the data they want. The problem is different from that of sorting through printed information. While libraries and organizations have standards such as the Dewey Decimal System or the Mathematics Subject Classification, none exist for the Web, which is too decentralized and growing too quickly for any such standard to exist. With the huge amounts of information on the Web, efficient mechanisms are clearly needed.

Hierarchies are popular on the Web for providing subject classifications. I will use as an example the highly-popular hierarchical index Yahoo[†]. Containing almost 40,000 entries, Yahoo would be useless if users were unable to efficiently find what they wanted. The tremendous success of Yahoo (over a million accesses per day) suggests that it has been well-structured.

In this paper, I discuss the following:

1. Problems with currently-available log data
2. Optimization criteria for information hierarchies
3. A quantitative definition for the semantic information content of a hierarchy

Log Data

Each Yahoo page has a name, which describes its concepts and its location in the hierarchy. For example, `Science: Computer Science` contains information about computer science and is the child of `Science`, which is a child of the top level. Each page can contain links to other Yahoo pages and to pages outside of Yahoo.

Currently, a great source of available information is unused: the access logs. For each file retrieval over the Web, the log at the server holds the name of the page, the time of the transfer, and the machine requesting the data. By examining the information in the log, one can infer user access patterns to see how effectively they find the information they want. For example, if many users were seen to walk down the tree based at `Computer Science`, return to the very top level, then eventually reach `Science: Information Technology`, at which time they left Yahoo (presumably to follow the links), that would suggest that users would more easily find `Information Technology` if it were a child of `Computer Science` rather than (or in addition to) being a child of `Science`. Additionally, if most visitors to `Science` also visited `Science: Computer Science` but not `Science`'s other children, this would suggest that `Computer Science` be raised in the hierarchy.

There are several difficulties with the log data currently available:

1. Log entries indicate the machine that requested a page, not the user. If many users access Yahoo from the same machine, it is difficult to sort out patterns of usage. For example, all Prodigy requests are made from one of four hosts.

2. Because Yahoo pages may be cached by clients, it is impossible to tell by which link a user reaches a page. For example, if a user views the Yahoo top page, then a child, then backs up to the top page, there will only be two data transfers, one for each unique page. The top page is cached locally by the client and is not retransmitted. If the next page visited is pointed to by both the top page and the first child, it is impossible to tell what link was followed.

3. The time stamps on the file requests do not necessarily indicate how long it took a user to find the item he or she was seeking. The user could have been distracted, or network delays may make it impossible to infer user time from the difference in times between requests.

I predict that browsers will eventually be augmented to solve many of these problems, recording user data that would be sold back (either with names or anonymously in bulk) to service providers or rating agencies. For the rest of my analysis, I will assume that complete access information is available for analysis.

[*] This research was performed while the author was employed by Microsoft Research, supervised by Dr. Daniel Weise. The author is grateful to Jerry Yang and Dr. Tom Knight for their encouragement and to Michael Ernst and Bjarne Steensgaard for their comments.

[†] Yahoo can be found at `http://www.yahoo.com`. Its creators and maintainers are David Filo and Jerry Yang.

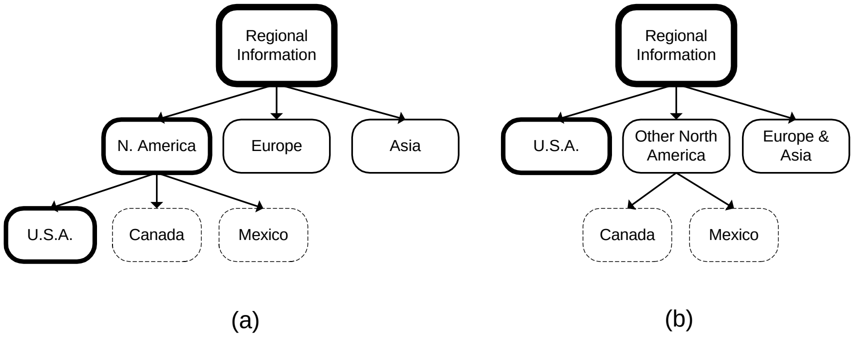

Figure 1: Heavier solid borders indicate more frequent access

Optimization Criteria

The cost of finding a page is equal to the sum of such terms as the following, all parameterized per user:

1. The number of mouse clicks (or keystrokes) times[‡] the cost to the user of the action (low for some people, high for those with RSI)
2. The amount of data transferred times the cost of transmission
3. The amount of time the user must wait for data times how much the user dislikes waiting
4. The total time taken times the user's value of his or her time
5. The amount of reading done times the cost of reading (which is high for poor readers, people with sight handicaps, and those with bad monitors)
6. The amount of time spent thinking (about obscure subject headings) times how much the user dislikes this sort of thought

That the user parameter values may be unmeasurable does not imply they are unimportant. The success of VCR+, for example, shows that people are willing to pay money to avoid thinking and entering data. While a user probably cannot directly give accurate information about his or her preferences, one may be able to infer the information from the user's behavior. Even though the values of the user parameters vary (perhaps depending on time of day, network load, and the user's mood), they are still real. If we instantiate values for the user parameters, we get an optimization function that can be used with a set of goals to determine which of two hierarchies is better.

Semantic Information Content

Assume there exists a Yahoo-like hierarchy in which links to the outside only occur at the leaves. By analyzing log and browser data, we can determine how much time, bandwidth, and mouse-clicking is required for users to reach each of the leaves.

Figure 1a shows a portion of the hierarchy containing regional information. The most frequently-accessed pages have darker borders. A mechanical tree rebalancing that just considered access counts ant not semantics would convert the tree into that shown in 1b. The U.S.A. link is moved up, since it is frequently accessed, and Europe and Asia are moved down (combined into a single node at the level of detail shown). While the second tree is better balanced, the first tree is conceptually simpler and presumably easier to navigate. Let us assume that the two trees are equally good for the given data set and optimization function. Let us call the balancedness of the trees B_a and B_b, respectively, for the trees in 1a and 1b. These values can be determined objectively from the structure of the trees and the access counts. We can now define the *semantic information content* (S) of the trees:

$$S_a + B_a = S_b + B_b$$
$$S_a - S_b = B_b - B_a$$

This gives us an objective quantitative statement of the difference in semantic information in each tree.

References

[1] Blahut, Richard E. *Principles and Practice of Information Theory*. Addison-Wesley, 1987.

[2] Knuth, Donald E. *The Art of Computer Programming: Vol. 3/Sorting and Searching*. Addison-Wesley, 1973.

[‡] This is a simplification, since the total cost might not be linear in the number of clicks.