WEB INDEXING: AN EXERCISE IN HYPERTEXT NAVIGATION

Dwight Walker 2/1 Nelson Street, Randwick 2031 email dwight@zip.com.au

Web indexing has become a sought-after skill as the Web grows dramatically. To enable indexers to grapple with Web indexing, I will first give an explanation of hypertext documents followed by a tutorial of a new tool that was developed for the inaugural AusSI Web indexing prize. It is called WEBIX and comes in DOS and Windows versions. Although aimed at Web site or electronic journal indexes, it can however be used to produce potted indexes of the Internet. These could be scaled up to the larger whole-of-Internet indexes like the W3 Virtual Library.

The Challenge of Indexing the Web

For the first time since Gutenberg, publishing has grown exponentially with the invention of the World Wide Web in the early 1990s. Everyone with Internet access has been able to publish on a small scale what interests them. This can be as simple as a home page (a summary of a person's interests and views) to million page electronic journal projects in the United States of America and the United Kingdom. The quality ranges wildly from rushed nondescript personal notes to structured academic treatises. Initially the Web grew without check. The trouble was that information retrieval became more and more difficult. Spiders such as Lycos and Webcrawler were invented to help locate information. They are software robots that roam the Web collecting millions of resources (or links), key words and outlines to enable users to select the appropriate resource. These databases of harvested links and information can be searched using search engines. Some groups built customised subject-oriented indexes such as Yahoo! Others built gangling lists like the Mother-of-all BBS. Users are producing their own classified indexes of resources on their home pages, organised around their own views and interests. (Middleton, 1995)

Virtual Libraries, Metadata and Bookbots

Currently indexers mainly index books, magazines, newspapers and databases. Very few have ventured onto the Web. Several libraries around the world have tried to classify Web sites according to Dewey, Library of Congress Subject Headings (LCSH) or a specific discipline, e.g. Mathematics on the Web (American Mathematical Society), Edinburgh Engineering Virtual Library (EEVL) (Heriot-Watt University) and Cyberstacks (Iowa State University).

Mathematics on the Web is a well organised specialised index of mathematics related links. The classification is based on academic lines and is easy to peruse. EEVL is a specialised classified UK engineering index. In its pilot stage, the six UK universities using EEVL have been extremely positive about it. Issues for the indexers were limiting the index to quality UK links and criteria for describing resources. Its classification is based on broad disciplines but is not traditional allowing room for modification - a good feature. Being open ended users can suggest valuable links.

Cyberstacks is built on LCSH so ties the freedom of hypertext to a book shelving scheme. It is hard to browse as there are very few cross-references and it cannot be searched for key words. A cross-classified index is going to be added to improve access but overall Cyberstacks is not a very useful system. A more flexible indexing process would be better.

AusSI's Web Indexing Starting Point

The idea for producing Web indexes came from a mathematics academic named Steve Hunt at Ormond College, Melbourne University. When I was setting up the Australian Society of Indexers Web page, I posted an entry on the Australian Electronic Publishing electronic mailing list to which he replied by phone from Melbourne. He was interested in what an indexer did, how they organised knowledge and how they went about producing an index. He then highlighted to me the need for "virtual libraries" to be constructed. These would be nothing less than large lists of well-researched links to other parts of the Web which held the content. Meta-indexes, i.e. lists of lists, were also ripe for production by indexers.

Since most indexers understood what a back-of-the-book index was I attempted (Walker, 1995; 1996a; 1996b) to capture a Web index which would look like one and describe it below. After sending messages on the indexers' INDEX-L mailing list in the USA, Jonathan Jermey, Tim Craven and I created a tool to convert an index into HTML (HyperText Markup Language), the lingua franca of the Web.

McKinley Internet Directory

In illustrating the complexity of the Internet, Maxwell (1995) from the McKinley Internet Directory described the Web as a zoo, where if you wanted to see the monkeys, you will be offered a thousand different paths to get to them and will often see the same one hundred monkeys all the time! The Internet is not geared for the user as links to information are often organised like a "laundry list" with no description or subject cues to enable the user to decide their relevance. On the other hand, the McKinley Internet Directory has produced value-added information: description, contents evaluation, up-to-dateness, organisation, ease of access, and a Star Rating. With a full-text search engine to enable targeted browsing, "the McKinley Group will be setting a standard for the organization, description, and evaluation of metadata about Internet resources."

Bookbots

The indexer's skills of analysing and sorting information can be used to leverage the unattached pieces of information on the Web and incorporate them into bibliographies of new electronic books. As described below this gives the reader the ability to step off into other information worlds through links or contacts given in an electronic book. An indexer can produce custom bibliographies of hypertext links to satisfy a reader's desires, a mixture of bibliographical searching and indexing. Fillmore (1995) envisaged where a book is no longer a single sequential piece of information but chunks of information floating in a sea free of moorings to books and page numbers:

> Cut loose from pages they've been affixed to for 500 years, the ideas and bits of information spin out of the books containing them and call for a new

organisation, promise to spin off into chaotic babble unless readers find and recognize their worth through transferable (named) organization. These first readers, those who identify, classify and name, as in the age of paper, may be indexers. In the dynamic environment of the web, indexers are asked . . . also to serve as 'link editors', associative thinkers in a recorded environment, with power to name their links, . . . What this means is a basic shift in the direction of indexing, and the creation of a kind of kinetic indexing or `bookbot', which, instead of a kind of back door and pointing from the reader into the book, points from the reader outward to the universal online bookshelf, with the goal of creating a customized index to a bookshelf area.

The Web as a Maze of Links

What is Hypertext? It is a way of navigating information. It involves creating links between associated pieces of information so that the reader can browse in many different related directions when searching for information. Hypertext is essentially non-linear in comparison with the linear layout of a book or periodical. It is very open-ended and is ideal for the customisation of an index as the data is fluid and chunky and can be cross-linked any way the reader, author or indexer wants. Bush describes "trails" below which are personalised indexes.

Vannevar Bush described hypertext in a seminal paper in Atlantic monthly in 1945. He called it "Memex" and envisaged a desk with panels which could be interacted with to guide the reader into different documents. Associations between chunks of information were named and formed indexes which were used by the reader to browse selectively.

> Bush's memex machine, which was designed but never built, was meant to personalize libraries. Physically the device looked like a work desk with viewing screens, a keyboard, sets of buttons and levers. Storage of printed materials of all kinds was accomplished using microfiche. Pages of books were selected for viewing by typing an *index code* [italics mine] to control a mechanical selection device or by moving levers to turn page images of the selected item. Any two items in the memex could be coded for permanent association. Bush called this coded association a trail, analogous to the trail of mental associations in the user's mind . . . Documents were stored on microfiche which had to be mechanically brought into position to be viewed on a screen. In one screen the user could view documents and in another screen make annotations on the same document. (Rada, 1991)

The idiosyncratic associations in Vannevar Bush's Memex system and other hypertext systems like the Web need support through classification systems. As the general link in hypertext is unlabelled (like a "goto" in computer programming) it leaves a feeling of "lost in space" or "spaghetti tangle" to the reader if one follows it into unfamiliar territory. The effort in filling the need for the large lack of supporting processes in hypertext is being done by the creator and accessor. The index can be the reader's personal librarian. (Rada, 1991) One advantage for indexers and librarians is that virtual structures created using hypertext make it more flexible to adjust or improve classification.

Navigating through hypertext documents

In this new method of information navigation, instead of page numbers and pages to thumb through we have hypertext links to follow, so-called surfing. In the Web, these links are represented by highlighted text or icons. When you click on them with your mouse, your browser opens the page that is linked to that highlight, a bit like opening a door. Once you have selected the link to follow, clicking on it lands you where that link goes to. It is reversible and the browser can back out of the tunnel you have crawled down.

The browser, whether Netscape, Mosaic or Internet Explorer, moves through the document where you decide to move. You can start anywhere by "opening" a "page". A page may be as long as you wish but for this exercise about a screenful. To keep a track of where you are in hyperspace, one uses bookmarks. These keep a record of a "place" in the sea of documents and give it a name. These can be used later to construct an index. In your surfing, you collect valuable links or bookmarks for later use, creating a knowledge base.

Hypertext building blocks: locators, chunks and links

Hypertext information is organised in chunks. Unlike books or magazines, you do not have a page number for a hypertext fragment. Horn (1989) discusses hypertext navigation and chunks. "A chunk is any familiar pattern." An information block is the basic subdivision of a subject matter. Horn's method of hypertext creation includes chunking in small, manageable units and selecting information only relevant to one basic point. Information blocks are grouped into information maps with order not specified.

For users, computers only display about one third of a page so documents are split into "chunks". Each chunk should be assigned at least one keyword that a reader will search for. In this respect, it is more important that indexers will anticipate readers' needs. Readability and ease of use is very important in online documents as resolution is less and fatigue sets in earlier. Readers' attention spans are shorter when reading static displays so an index is an invaluable retrieval tool in online documents. (Lathrop, 1996).

Hence the way to access or locate a piece of hypertext is to name it. With the Web's version of hypertext, the names are file names or URLs (see below). So to create a simple index, you first create several small files of information corresponding roughly to a chapter of a book. Name the files in a logical way, e.g. publicat.htm for publications, affiliat.htm for affiliates. You could develop this much further if you wanted to. One can also have links to a particular part of a document by placing anchors or tags inside the document. For the rest of the article I will be dealing only at the document level so will ignore anchors. This collection of small files together forms a "family of documents". They all relate to one another and to the home page of the document (a table of contents pointing to the other major documents in the collection) illustrating the Web's hypertext nature.

Web indexing tools

I have created with others' assistance an array of tools to index the Web.

Editor or Indexing Software: For those not used to indexing, a simple text editor can be used to create some simple indexes. For the more advanced indexer, there are several large professional packages used to produce lengthy indexes, namely MACREX and CINDEX. They can all print an index to disk which can then be run through WEBIX.

Index to HTML converters: (a) CINDEX add-on HTML/Prep Leverage Technologies have produced an add-on for CINDEX called HTML/Prep including the files HTML.COD and HTML.FMT.

- First OPEN or CREATE an index
- Type GET HTML (done once)
- Type SET TYPE then change "Field Type?" to HTML (done once)
- Text an index is produced with file names as locators (see below).
- The view is changed to formatted using VIEW/FORM
- The index is printed to file using PRINT/FILE.
- This disk file is converted into HTML using the converter HTML/Prep, viz. htmlprep webindex

Figure 1: Creating an index entry

```
+-----------c:\cindex\aussi.ndx----------+
8  Australian Society of Indexers
   http://www.zeta.org.au/\~aussi
1  objectives of AusSI objectiv.htm
2  pricing of an index pricing.htm
3  publications publicat.htm
4  1st International Conference Proceedings
   proceed.htm
Newsletter
6  abstracts newslett.htm
7  index publicat.htm
5  sample (Jan/Feb 95) janfeb95.htm
+-----------------------------------------+
Rec 8 [06:04:96, 01:14PM; ]              37
> Australian Society of Indexers
P http://www.zeta.org.au/\~aussi

Command  edit
ALL Sorted Formatted  8 of 8 records (0 new),
size 100  0:15        Help F11
```

Figure 2: Formatted view ready to be printed to disk

```
+-----------c:\cindex\aussi.ndx----------+
A                                        |
Australian Society of
Indexers<c>http://www.zeta.org.au/~aussi
O                                        |
objectives of AusSI<c>objectiv.htm       |
P                                        |
pricing of an index<c>pricing.htm        |
publications<c>publicat.htm
!1st International Conference
Proceedings<c>proceed.htm
!Newsletter
abstracts<c>newslett.htm
index<c>publicat.htm
sample (Jan/Feb 95)<c>janfeb95.htm
+-----------------------------------------+
Command
ALL Sorted Formatted  8 of 8 records (0
new), size 100  0:15        Help F11
```

Index to HTML converters: (b) MACREX macros For MACREX, macros can be setup to convert the text index to HTML.

Index to HTML converters: (c) WEBIX There is also a program dubbed WEBIX which was written to convert ASCII text files from the index format into the HTML format. This was written by two volunteers - Jonathan Jermey and Tim Craven. One wrote the Windows version (Weblinkr) and the other the DOS version (INDTOHTM) respectively. These can be used with any editor or indexing software that follows the layout below.

The Web Index

Specifications for a Web Index: What follows is a simple explanation of what the index needs to look like to be able to be converted to HTML by WEBIX:

```
header URL1.htm
 subheader URL2.htm
  subsubheader URL3.htm
```

How can I create a hypertext index? Once you have the file names (or documents) and their subjects in a list:

subject	file name (or document)
affiliates	affiliat.htm
publications	publicat.htm

you can construct an index in alphabetical order using your CINDEX or MACREX program or a simple text editor like EDIT in DOS or NOTEPAD in Windows.

Here is a sample from the AusSI Web site index:

subject	file name or link (on end of line)
objectives of AusSI	objectiv.htm
pricing of an index	pricing.htm
publications	publicat.htm
1st International Conference Proceedings	proceed.htm
Newsletter	
sample (Jan/Feb 95)	janfeb95.htm
abstracts	newslett.htm
index	publicat.htm

Each level is indented by 2 spaces or a tab.

WEBIX's function: WEBIX takes the file of subject headers and links and converts it into hypertext suitable for the Web. Specify the names of the text index file (input) and the Web index file (output). Here is how to use the DOS version of WEBIX:

```
indtohtm webindex.txt webindex.html
    ^          ^          ^
WEBIX    text index    Web index
```

Viewing the Web Index

Load the file into Netscape, say, by choosing File | Open File. This will load the file into your browser as though you were out there moving around the Web. The hypertext files on your hard disk are your Web world. This is what Netscape will show:

objectives of AusSI
pricing of an index
  publications
    1st International Conference Proceedings
    Newsletter
      sample (Jan/Feb 95)
      abstracts
      index

Any word that is underlined has a page linked to it. This is a live part of your Web now. The index has a finger pointing to all the relevant pages on those topics.

Overall Cycle: The cycle is:

1. edit the index in a text editor or CINDEX/Macrex
2. print the index to disk or save it as a text file
3. run INDTOHTM, Weblinkr or HTMLPrep
4. (re)load it into Netscape

More advanced indexing: bibliographic web indexing

WEBIX has the capability to be extended for use on the Internet itself. Instead of using local file names like publicat.htm, you can use full Universal Reference Locators (URLs) such as http://www.zeta.org.au/~aussi. A URL is the address of the Web page on the Internet. As long as you know the URL you can access data anywhere on the Internet with a click of a mouse button.

So, the index entry "Australian Society of Indexers http://www.zeta.org.au/~aussi" will produce "Australian Society of Indexers", which when clicked on will produce this:

Figure 3: Top of Australian Society of Indexers Home Page

Australian Society of Indexers

Formed in 1976, the Society developed from the Society of Indexers in Australia, which had been in existence for quite some time.

"Mirrors within mirrors"

What is an Indexer?
Indexers are information professionals. They create back-of-book indexes, database indexes, magazine indexes. They work closely with publishers and authors to create an index for their publication. It is a fairly specialised job which requires lots of concentration and a quick hand at editing and choosing good terminology for a topic. Often they work alone. AusSI helps create links between these professionals in the field and conferences. Browse what we offer...

Touching up the HTML

I use HTML Writer to edit my HTML pages such as entering a new page header or a cross reference in my index file. To create a cross reference first put in local "anchors" then create local jumps to them. The anchor is the word you are referring to and the jumps are the see and see also references. So, it is possible to refer to or jump within a page and not just to a page.

Some setup considerations

If you do not have an Internet connection, install MOZOCK.DLL, a dummy Winsock. (Winsock is used by Windows programs to communicate with the Internet). Later when you have Internet access, use Trumpet Winsock. When you then use Netscape and it tries to dial out to connect to the Internet through Winsock it will not cause problems. This allows you to build Web pages on your PC without having an Internet connection.

You will need some dummy HTML files to work on. An easy way is to start up HTML Writer and create a few, or if you have access to the Internet, save a few pages from your Internet travels. These form good material to index. There are also WordPerfect to HTML and RTF to HTML converters on the Internet to enable you to create some HTML documents. Microsoft Word's Internet Assistant and Adobe PageMaker 6.0 also produce HTML pages.

Finally, you will want to incorporate this index into your Web page family. Put a link to it from your home page and at least the main part of each section of the family of documents. HTML Writer can be used for this touching up.

Conclusions

So with WEBIX, a Web index can be created using basic tools like your existing indexing software, Netscape and HTML Writer. Like any index, it will go out of date and because electronic publishing is so fluid it will go out of date even more quickly than a paper-based one! Whenever the documents you are indexing change their name, your index is out of date. There are tools (like Adobe's Site Mill for Macintosh) that keep track of changing file names and update all pages that refer to them including the index. WEBIX can be used with these tools to automatically update the locators in the index you are working on. That will certainly be a semi-automated way of maintaining indexes of the Web.

There's still the need to add new entries though as the Web site grows. A database of file names with the date you last indexed it could be used as a check list to see if the index is out of date. I am sure some of the CINDEX and MACREX tools could be tweaked to keep track of changing Web pages and indexes - telling when an index entry was last updated and by whom. These all point toward better indexing for the Web.

"They [the phone company or networking giant] need publishing people, specifically people who understand how to chunk up, weight, and value information... [Hot Lists] are nothing but index entries begging to be customized to the reader, by the reader." (Fillmore, 1995)

So there will be great demand for the flexible computer literate Web indexer.

References

Adobe. http://www.adobe.com

American Mathematical Society. "Mathematics on the Web" http://www.ams.org/mathweb/mi-mathbyclass.html

Australian Society of Indexers, "The Art of Indexing the Internet" http://www.zeta.org.au/~aussi/inetindx.htm

Bush, Vannevar, 1945. "As we may think". *Atlantic monthly* 176 (1, July): 101-108. (http://www.theAtlantic.com/atlantic/atlweb/flashbks/computer/bushf.htm)

Cyberstacks. http://www.public.iastate.edu/~CYBERSTACKS

December, John, 1994. "New spiders roam the Web". *Computer-mediated communication magazine* 1 (5):3. http://sunsite.unc.edu/cmc/mag/1994/sep/spiders.html

Edinburgh Engineering Virtual Library (EEVL). http://www.eevl.ac.uk/

Fillmore, Laura, 1995. "Beyond the back of the

book: indexing in a shrinking world". *Key words* 3 (3, July-August): 16-20. http://www.obs-europa.de/obs/english/papers/mont2.htm

Horn, Robert E., 1989. *Mapping Hypertext: an analysis, organization, and display of knowledge for the next generation of on-line text and graphics.* Lexington MA, Lexington Institute.

HTML Writer. http://lal.cs.byu.edu/people/nosack/

Lathrop, Lori, 1996. "Considerations in indexing online documents". *Key words* 4 (1, January-February): 1, 29.

Maxwell, Christine, 1995. "Cyberspace: The newest indexing frontier". *Key words* 3 (3, July-August): 13-15. http://www.mckinley.com

Middleton, Michael, 1995. "Indexing the Internet," in *Indexers - Partners in Publishing, Proceedings from the 1995 International Conference of the Australian Society of Indexers edited by Max McMaster. Australian Society of Indexers.* ftp://ftp.fit.qut.edu.au/InfoSys/papers/asindex.wp5

Netscape. http://www.netscape.com

Rada, Roy, 1991. "HYPERTEXT: from text to Expertext". London, McGraw-Hill. http://www.eecs.wsu.edu/~rada)

Walker, Dwight, 1996. "AusSI Web indexing prize". *Indexer* 20 (1, April): 6-7.

Walker, Dwight, 1995. "Web Indexing Prize". *Australian Society of Indexers newsletter* 19 (10): 7-8.

Walker, Dwight, 1996a. "Web Indexing Prize". *Australian Society of Indexers newsletter* 20 (1): 6-7.

Walker, Dwight, 1996b. "Web Indexing Prize". *Australian Society of Indexers newsletter* 20 (2): 4-6.

AUTOMATIC INDEXING AND ABSTRACTING

Glenda Browne PO Box 307, Blaxland 2774

This paper will examine developments in automatic indexing and abstracting in which the computer creates the index and abstract, with little or no human intervention. The emphasis is on practical applications, rather than theoretical studies. This paper does not cover computer-aided indexing, in which computers enhance the work of human indexers, or indexing of the Internet.

Research into automatic indexing and abstracting has been progressing since the late 1950s. Early reports claimed success, but practical applications have been limited. Computer indexing and abstracting are now being used commercially, with prospects for further use in the future. The history of automatic indexing and abstracting is well covered by Lancaster (1991).

Database indexing

The simplest method for indexing articles for bibliographic databases is extraction indexing, in which terms are extracted from the text of the article for inclusion in the index. The frequency of words in the article is determined, and the words which are found most often are included in the index. Alternatively, the words which occur most often in the article compared to their occurrence in the rest of the database, or in normal language, are included. This method can also take into account word stems (so that "run" and "running" are recognised as referring to the same concept), and can recognise phrases as well as single words.

Computer extraction indexing is more consistent than human extraction indexing. However, most human indexing is not simple extraction indexing, but is assignment indexing, in which the terms used in the index are not necessarily those found in the text. For assignment indexing, the computer has a thesaurus, or controlled vocabulary, which lists all the subject headings which may be used in the index. For each of these subject headings it also has a list of profile words. These are words which, when found in the text of the article, indicate that the thesaurus term should be allocated.

For example, for the thesaurus term "childbirth", the profile might include the words "childbirth, birth, labor, labour, delivery, forceps, baby" and "born". As well as the profile, the computer also has criteria for inclusion instructions as to how often, and in what combination, the profile words must be present for that thesaurus term to be allocated.

The criteria might say, for example, that if the word "childbirth" is found ten times in an article, then the thesaurus term "childbirth" will be allocated. However if the word "delivery" is found ten times in an article, this in itself is not enough to warrant allocation of the term "childbirth", as "delivery" could be referring to other subjects such as mail delivery. The criteria in this case would specify that the term "delivery" must occur a certain number of times, along with one or more of the other terms in the profile.

Database indexing in practice

In practice in database indexing, there is a continuum of use of computers, from no computer at all to fully automatic indexing. Options include no computer, computer clerical support (e.g. for data entry), computer quality control (e.g. checking that all index terms are valid thesaurus terms), computer intellectual assistance (e.g. helping with term choice and weighting); and automatic indexing. (Hodge, 1994)

Most database producers use computers at a number of different steps along this continuum. At the moment, however, automatic indexing is only ever used for a part of a database, for example, for a specific subject, access point, or document type.

Automatic indexing is used by the Defense Technology Information Center (DTIC) for the management-related literature in its database; it is used by FIZ Karlsruhe for indexing chemical names; it was used until 1992 by the Russian International Centre for Scientific and Technical Information (ICSTI) for its Russian language materials; and it was used by INSPEC for the re-indexing of its backfiles to new standards. (Hodge, 1994)

BIOSIS (Biological Abstracts) uses computers at all steps on the continuum, and uses automatic indexing in a number of areas. Title keywords are mapped by computer to the semantic vocabulary of 15,000 words. The terms from the semantic vocabulary are then mapped to one of 600 concept headings, which