Ending The Frustration of Web Searches

Searching for specific technical, academic or legal information on the Internet is increasingly a long, difficult, convoluted and frustrating task. Why is this, and what can be done to make precise searches easier and faster?

Searching for something you want to buy, or for coverage of mainstream academic and technical subjects such as "number theory" or "HTML tag definitions" is easy. For this kind of search, the Web search engines are great.

The problems start when you are looking for something more specific such as "What is the legally necessary and sufficient information a vendor must supply to a prospective purchaser of a piece of rural land within the jurisdiction of the state of Minas Gerais in Brazil?"

The latter is a search I have been attempting on the Web for a while now without success. I am not involved professionally in legal research. It is simply something that I wanted to look into for personal reasons. Naturally, I formulated the question and conducted the search in Portuguese, the language of Brazil. One would think my search should have been easy since Brazil publishes all its laws on the Web for anybody to freely view.

To a human reader my question is precise. It homes in on exactly what I want to know. However, entering it into a Web search engine gets me precisely nowhere. What I need to do is abstract the relevant keywords and, starting with the most general, put them in order of increasing specificness as follows: "law land rural obligations vendor minas-gerais". I then enter these into an Internet search engine, but again, I get precisely nowhere. I get articles about tax payable on land, articles about reformation policies regarding rural land and articles on lots of other topics relating to rural land in Brazil. But nothing about what I want to know and what I have specifically asked for.

Why is this? The problem is to do with how Web pages are indexed. Information search and retrieval systems rely on indexing. And web pages generally are not indexed in a way that best serves the searcher.

The major web search engines today compile their indexes by extracting keywords from the text contained in web pages. The more a given keyword appears within a page, the higher is the ranking given to that page with respect to that keyword. When a searcher enters several keywords into a search engine, the web pages appearing higher in the list will be those in which the submitted keywords appear in closer proximity to each other. There are so many pages on the Web that many will have the same ranking in a search. To try to be fair, some search engines use a scheme called churning. This means that in one search one relevant page will be listed first whereas for another search on the same keywords another equally-ranking page will be listed first.

This process of building an index from keywords appearing in the text began with books. Probably the first instance of this is Cruden's Concordance of the King James Bible, first published in 1737. It was this in fact that triggered my interest in indexing. What struck me was that during a 7-year intense study of the Bible many years ago, Cruden's Concordance proved to be nowhere near as helpful as I expected it to be. And I wondered why.

One day, I remembered a very interesting passage I had read and I wanted to refer to it in the light of another passage I had read in an entirely different part of the Bible. I knew clearly the ideas, notions and concepts embodied within this lost passage. I knew exactly what it was about. I thought of relevant keywords and wrote them down. Then I used Cruden's Concordance to look up the references to these words. However, none of these keywords led me to the passage I was seeking.

Many months later while reading the Bible, I came across my lost passage. I read it very carefully again. Yes, I had understood the ideas within it. Yes, I had selected completely relevant keywords in order to search for it through Cruden's Concordance. But not one of these highly relevant keywords appeared in the text of the passage. Instead, the ideas, notions and concepts within the passage had been expressed far more powerfully in terms of small common words arranged into well tuned phrases. Furthermore, not a single one of those small common words alone conveyed anything specific about any of the ideas, notions or concepts contained in the passage.

Having read and referred to books whose indexes were effective, I gather that somewhere during the intervening centuries some authors and publishers became aware of this problem and solved it by including words and phrases in their indexes that did not appear in their texts. Unfortunately, the advent of the computer turned the clock back again in this regard. This was especially so with books on computer subjects. Perhaps some of the most notoriously useless indexes ever produced are those that appear at the backs of computer software manuals. I have read libraries full of these and was nearly always forced to find what I was looking for via the contents list at the front.

The reason for this retrogression is that, with the text of a book on a computer, it is so easy to write a little program to extract all the significant words that appear in the text, sort them into alphabetical order and list them together with their relevant page numbers. How much faster and cheaper this is than to have a human indexer compile an index. The spiders or web crawlers used by search engines to compile their indexes of web pages are essentially very sophisticated versions of the index extraction programs used to compile book indexes automatically. However, they all leave us with the legacy of Cruden's Concordance. They are all considerably less effective than one should rightfully expect.

It was not always this way. I mentioned that I have read and referred to many books with effective indexes. I wrote a comprehensive 400-page manual for a software package I created in the early 1980s. I compiled an index for this manual. I also wrote a little index extraction program to generate this index. However, my little program did not get its keywords from the text.

I divided my manual into what I called *semantic units*. A *semantic unit* is the smallest piece of content - a paragraph, a group of paragraphs or a section - that a reader may want to look up and access directly from the index. I then read each *semantic unit* carefully and asked myself the following question: "What keywords or phrases are likely to spring to a reader's mind when this is the information he is after?" Then I placed each of these words and phrases on a separate line immediately before the *semantic unit* of text to which they referred. I prefixed each of them with a special escape character. This enabled my indexing program to tell them apart from the text. It also indicated to the manual's print and display programs that they were not part of the text and should therefore not be printed or displayed as part of the content. My indexer program then collected all these words and phrases from the book, sorted them into alphabetical order and stored them as an index file, which could be printed as part of the manual.

The pivotal observation from this exercise was that less than half of my keywords appeared in the text of the manual. This index proved to be very effective. Because it works by referencing *semantic units*, I refer to this method of indexing as *semantic indexing*. The crude method - that of merely extracting keywords from the text - I refer to as *syntactic indexing* because it simply looks at the syntax of the text: not the meaning in the text.

A facility enabling a writer to list relevant keywords for a Web page separately from its text has, almost from the outset, been provided by the Hyper-Text Markup Language (HTML) used to encode Web pages. It is known as the Keywords Metatag and has the form:

<meta name="keywords" content="eggs, bacon, cheese">

When, in the mid 1990s I had constructed a web site of over 680,000 words in over 2,300 files, I indexed all my pages in the same way I had indexed my software package's user manual those many years before. I treated each of my Web pages as a separate *semantic unit.* In fact, in some instances I divided a large Web page into several *semantic units,* delimiting and naming them using HTML Anchor Tags.

At first this worked fine. My web site became effectively indexed by Web search engines. But then something happened to the World-Wide Web that changed everything. It started to be invaded and subsequently became almost completely dominated by business and commerce.

The objective of the Web had been to provide a means for people to publish information which others could access quickly and freely. The webmaster of each site simply wanted to make his information available only to whomsoever genuinely wanted it. With the invasion of business and commerce the Web changed to become a medium for buying and selling. The webmaster of a commercial site had the objective of attracting anybody and everybody in order that their eyes may be dazzled by flamboyant artwork and animated gismos and thereby wooed into buying the products being offered through the site.

The search engine companies seized on the commercial opportunity of offering to rank commercial websites higher in search results lists in return for payments. This insidious practice distorted the relevancies of different web sites and frustrated searchers who were looking for academic and other non-commercial sources of information on the Web. Fortunately this practice seems to have died an appropriate death.

However, the commercial motive led an all-too-large a number of commercial webmasters to abuse the use of the Keyword Metatag. One classic example was where the webmaster of a car sales company inserted pornographic words into the Keywords Metatag so that when web surfers put such words into a search engine they would be taken to his site where he hoped they would be dazzled into buying a new car while on their way to finding a genuine porn site. Another classic case was when I had a passing interest in making geometric shapes in basket-work. I entered the words "ellipsoidal basket" into a major search engine. It only found one website: a Russian child-pornography site. Happily, if you enter those same words today, this same search engine does find websites relevant to ellipsoidal baskets.

This practice of putting what are termed *false attractors* into web page indexes became such an epidemic in the late 1990s that major search engines began to penalise Web pages whose Keyword Metatags contained words that did not appear in the text. This caused web sites like mine that had been painstakingly semantically indexed to all but disappear from search engine indexes. Later, some of the major search engines opted simply to ignore the Keywords Metatag, reverting to abstracting significant words solely from the text content of each Web page.

The bad boys of commerce fought back. They placed all their *false attractors* - naughty keywords that had nothing to do with the content of their page - into the text of the page. However, they made this part of the text the same colour as its background so that a person viewing the page would not see the naughty words. The search engines counter-attacked by looking for the coding that made any part of the text the same colour as the background. If a web page were found to contain text like this, the search engine would down-grade or reject that page from its index.

Many other nasty little techniques have been exploited by commercial webmasters and eventually countered by search engines. However, these drastic countermeasures employed by search engines collaterally penalise Web pages that use the same coding techniques for perfectly genuine purposes.

This is why it is now so difficult to find specific non-commercial information on the Worldwide Web. So, what is the solution?

Probably the majority of pages on the Web do not include Keyword Metatags. Search engines must always therefore default to extracting keywords from text for such pages. In fact the major search engines need to - and probably will continue to - confine themselves to this method Cruden used for the Bible way back in 1737. I think the only way to enable people to find specific non-commercial information on the Worldwide Web is for custodians of specific information to semantically index their web pages and provide their own search engines which use manually-built semantic indexes instead of indexes abstracted from the content text. I have included a semantically indexed search engine applet on this web site.

In the case of my search, I want to find specific clauses within specific laws that provide the answer to my question. The government web site www.planalto.gov.br hosts the laws of Brazil. Looking in the source code of some of these pages I found no Keyword Metatags. Consequently, search engines can only build indexes of these laws by abstracting significantly large words within their texts. This explains why I have not been able to find what I am looking for.

In my opinion, under present technology, it is fundamentally impossible for any automated device to make semantic judgements or associations. Consequently, to make the law really accessible, www.planalto.gov.br/ would have to form a team of legal professionals, schooled in the art of semantic indexing, to semantically index each web page containing a law. This is no small task. However, I think that the cost-benefit to the country in providing fast accurate targeting of applicable law would be tremendous in terms of time saved by lawyers and judicial functionaries throughout the whole country in preparing cases and formulating agreements. I suggest that this would be a very worthwhile project to help speed the economic development of Brazil.

In the case of my own website, I have long since given up trying to engineer some means by which such search engines as Google could be persuaded to index it. Instead, I have created PDF versions of many of my major pages. I have renamed each page with a filename comprising very well-chosen keywords joined by underscore characters. I have placed all these PDF files in a directory called "uploads". I have then pointed my FTP, aMule and gtk-gnutella servers and my Kademlia searcher to this directory as a permanent source of uploadable files. I have also placed them as both PDF files and “freesites” within FreeNet.

This way, my major web pages get indexed in the distributed hash tables of various non-web networks. The secret, however, is that in each PDF file are links that refer back to the relevant pages on my website. A person who chooses to download such a file can thereby find, and gain access to, the proper web-version of the page. I have also incorporated these key web pages into Webrings. Between them, these alternative methods have once again made my website visible to the world.

Parent Document | Robert John Morton, Belo Horizonte 09 Feb 2006, Aug 2012