How do you find information on the World Wide Web? The need for tools to make locating information easier was recognized very early in the development of the Internet. In fact Archie, the program credited with being the world's first Internet search engine, was created in 1990, a year before Tim Berners-Lee created the World Wide Web. The first website, created by Berners-Lee at http://info.cern.ch/ and made available online on 6 August 1991, was also the first web directory, as Berners-Lee kept an ever-growing, categorized list of the websites that other people created. That list became the WWW Virtual Library, a volunteer-maintained catalogue still available at http://vlib.org/.

Web directories and search engines

A web directory is a human compiled list of links to web pages, typically organized into a hierarchical structure of subject categories. After Berners-Lee created the Web in 1991, general directories that attempted to cover the whole of the contents of the Web vied for popularity with search engines, which relied on automated software to provide most of their information. In 1994 two Stanford University PhD students, David Filo and Jerry Yang, created probably the best-known general web directory, Yahoo! Directory (http://dir.yahoo.com/). Although the original basis of the directory was the contents of the two students' personal lists of websites, when Yahoo! became a commercial venture they soon hired information professionals to rationalize, maintain and expand the hierarchy of web links.

But in 1994 there were less than 10,000 websites. By 1998 there were approximately 3.5 million websites (O'Neill et al, 2003), and the World Wide Web has continued to grow exponentially – in 2006, its size was estimated at over 100 million websites (Netcraft, 2006). 1998 was the year when some website creators were frustrated enough with the time taken for Yahoo! to include links to their sites that they set up their own rival Open Directory Project (http://dmoz.org/). It was also the year that another pair of Stanford University graduates, Larry Page and Sergey Brin, launched the Google search engine.

Clay Shirky, a commentator on the Web and emerging technologies, argues in his article ‘Ontology is overrated’ (2005a) that Yahoo! Directory's decline in popularity and the rise of Google demonstrates the failure of traditional human classification techniques when applied to the Web. Human classification and indexing works optimally with a small corpus and a restricted set of topics; when the text is fixed and unchanging; when categories are formally defined; and when both users and classifiers are experts and have a shared culture and vocabulary. The further the problem domain deviates from this ideal, the more expensive, unmaintainable and unsuccessful human indexing will be. The World Wide Web, with its huge corpus on a vast range of topics written by many authors, its constantly shifting and changing content, its lack of central authority and its host of inexpert, untrained users, combines all of the characteristics that make human indexing a strategy least likely to succeed.

How do search engines work?

Imagine the World Wide Web as a vast library of books. The books are not classified and ordered on shelves; instead, they are heaped in a huge pile on the floor. In addition, none of the books have indexes. The search engine is the self-appointed librarian of this library. While the users have the option to sift through the books themselves, they can instead choose to submit a query to the librarian, who promises to give them in return a list of page references that will enable them to easily find information on the chosen subject. (The librarian also gives the user a list of adverts on the same subject, and it is the money from the advertisers that pays the librarian's salary.)

The quality of service provided by the search engine can be measured using the following factors:

- The size of the corpus. Search engines assume that their users will want to tap the range of information offered by the whole of the Web. So the more books the librarian can search through, the better.
- The speed of the answer. Users expect the World Wide Web to provide results quickly, and if their expectations are not met, they will go elsewhere.
- The availability of the service. Maximizing the time that the search service is available increases the number of customers served. Again, if the service is unavailable when customers have need of it, they will go elsewhere, and may transfer their loyalty to a rival organization.
- The accuracy of the results. Usually measured in terms of precision and recall, this factor is the one most people will think of first, and of course it is vitally important; but I deliberately placed it last to emphasize that even if your results are extremely accurate, if you cannot deliver on
the other factors as well, then your overall service quality will not make the grade.

Search engines require their users to submit their search queries via a web page, usually by means of a single search box into which the user enters terms. One of the challenges that they face is that they, like back-of-the-book indexes, constrain users to express their search query solely in terms of the headings and vocabulary chosen for them by the indexer whereas the search engine allows searchers to choose whatever terms they like. In addition, the web page interface is available at all times of the day and night, and can accept search queries from huge numbers of users simultaneously – more than could ever be handled by an army of helpdesk staff in a traditional library.

Having received the search query from the user, the librarian could begin to fulfil the request by taking the first book from the vast unordered pile, scanning through each of its pages in turn to see whether any of the material matches the search query, looking through the next book, and so on, until the search has exhausted all of the books in the pile. The time taken to search in this way through the billions of pages that make up the World Wide Web, even when the searcher is a computer program rather than a human, would stretch into months – clearly unacceptable to the users waiting at their computer for the results to their query.

A better solution would be for the librarian to hire a second person – let us call them an ‘indexer’ – who can spend all of his or her time classifying and indexing the contents of the library and producing a master log – or index – of results. Then, whenever the librarian receives a search query, he or she can fulfil it not by going through the contents of the library itself but by searching the index, and returning to the user the results gleaned from the information there. This is the approach taken by real search engines like Google. A software agent (that is, a computer program that runs continuously without human user interaction), often referred to as a ‘spider’ or ‘robot’, accesses web pages, analyses their contents and records the results in a database. The database is organized so that the computer program that receives users’ search terms via the web interface can quickly query the database records and return search results that can be presented back to the user in a human-readable format. Not surprisingly, this database is referred to as an ‘index’, because its function for the computer is equivalent to the function of a back-of-the-book index for a document reader – it enables fast access to sought information, eliminating the need for the time-consuming and laborious process of scanning the source documents page by page.

### The Google search engine

Google’s software agent, called ‘Googlebot’, continually locates billions of web pages, analyses them and saves the results in Google’s web index. The algorithms it uses for analysis are obviously proprietary and their details a company secret, but common practice for search engines is to break down the page into its individual elements, usually down to word level, and to examine and cross-reference each of those elements. Significance is usually attached to the position of the element, with words in titles and subheadings often given more weight than those in ordinary body text. Words in META tags – HTML elements that allow web page authors to provide an indication of the key words and important content of their page – used to be rated highly by search engines, but are now often ignored because of the potential for abuse by unscrupulous authors. Page contents are often changed and pages added or deleted by authors, so the Googlebot must revisit websites often to keep its information up to date. It takes several months for the agent to complete a sweep of its chosen portion of the Web, but it will revisit pages on a more frequent schedule if they are changing often and they are deemed suitably important.

How does the Googlebot judge the importance of a web page? This is achieved by means of Google’s famous PageRank algorithm, developed by Google founders Page and Brin at Stanford University. PageRank is used by Google to determine the relevancy of search results, and it is the reason that Google quickly overtook established search engines such as AltaVista and Lycos in popularity when it was launched in 1998.

PageRank relies on the uniquely democratic nature of the web by using its vast link structure as an indicator of an individual page’s value. In essence, Google interprets a link from page A to page B as a vote, by page A, for page B. But, Google looks at more than the sheer volume of votes, or links a page receives; it also analyzes the page that casts the vote. Votes cast by pages that are themselves ‘important’ weigh more heavily and help to make other pages ‘important.’

PageRank provides a way for Google to put more relevant pages at the top of its lists of search results. It makes use of thousands of individual human decisions without requiring direct human input into the relevancy calculations.

### Beyond Google – alternatives to search engines

The assignment of subject keywords to web content by means of an automated algorithmic process, rather than by human indexing, has become a standard and accepted approach – a ‘good enough’ solution in the face of the huge, constantly changing corpus that is the World Wide Web. I have already mentioned one alternative approach – the assignment of subject keywords by the web content creators, specified in META tags. This approach is discredited because it is open to abuse by those who assign deliberately misleading or inaccurate keywords to their content for financial gain. One other alternative which is being experimented with is the assignment of subject keywords by web content consumers – the people browsing the Internet. This has led to the rise of the ‘tagging’ phenomenon, or ‘folksonomy’ – a neologism implying the creation of a taxonomy by the collective actions of ordinary folk on the Web. Sites such as del.icio.us, the social bookmarking site, and flickr,
the photo management and sharing application, invite their users to assign keywords, or ‘tags’, to web content, which can then be used for searching by other people.

Those who put their faith in the power of collective intelligence see folksonomies as a major advance that will revolutionize searches. One commentator writes, ‘the obvious disadvantage of this approach is its lack of precision (synonym/antonym control, related terms, context, etc.), but in most practical usage scenarios the trade-off between simplicity and precision makes sense’ (Headshift, 2004), and Clay Shirky summarizes the main advantage: it is ‘cheap cheap cheap’ (2005b). It is possible that tagging may have some part to play in the future of searching, but the drawback to the current tagging approaches are obvious.

Another view is that the current aim of search engines to index the whole Web (or at least, as much of the Web as they can with the resources available to them) is unsustainable. There are large commercial advantages to having your web page appear high on a list of search results produced by a popular search engine such as Google. ‘Spamdexing’ (spamming the search engine’s web index) is a real problem for Google. Automated software can create junk web pages whose sole target audience is the Googlebot, and the increased use of web pages whose content can be edited by third parties, such as wikis and blogs, provides another way with links, artificially boosting the PageRank scores of the web pages that those links point to. A vast twilight zone of junk web pages, created by computers for consumption by third parties, such as wikis and blogs, provides another way with links, artificially boosting the PageRank scores of the web pages that those links point to. A vast twilight zone of junk web pages, created by computers for consumption by third parties, provides another way with links, artificially boosting the PageRank scores of the web pages that those links point to.

In this environment, there is increasing benefit to limiting the scope of the search facility to high-quality resources. General web directories that attempt to classify the contents of the entire Web have given way to focused directories that take advantage of their human editorial input to cherry-pick high-quality resources in a particular subject area. Web-based question-answering facilities, where you pay a small fee for researchers to answer your question, offer an alternative to wading through irrelevant search results to find the information you are seeking. And there is evidence of decreased search engine use among people in the 18–24 age bracket, as they turn instead to social networking sites such as MySpace to find what they want on the Web (Orlowski, 2006). Given these trends, what changes can be made to make the search engine technology of the future more appealing to its users?

**Beyond Google — better search engines**

Analysis of web content by search engine spiders has traditionally been limited to a consideration of the text as a series of characters and words, where ‘word’ is loosely defined as a set of letters or numbers bounded by spaces or punctuation symbols. The search interface, in turn, invites the user to enter a set of words, which are then matched with those found in the search engine’s index. This means that while Google’s search box may seem to be asking the question, ‘What subject do you want information on?’, the key to a successful search is often instead to consider, ‘What word or combination of words will be most likely to appear on web pages that address the subject I am interested in, and least likely to appear on pages that are irrelevant to me?’ This complicated question is not one that many users, unfamiliar with the internal workings of search engines, will consider, or indeed are willing to address. Search engines are beginning to employ increasingly sophisticated techniques to narrow this gap between user expectations and the actual function of the software.

The problem with word-based search is that there is no one-to-one correspondence between words and meanings. In fact, a single word may have multiple meanings, and a single meaning may be represented by more than one word (or phrase; for example the same meaning may be represented by the word ‘equivocate’ and the phrase ‘beat around the bush’). Linguists use the term ‘lexeme’ to refer to an individual unit of meaning or sense, and some work has been done to create search engines based on lexemes rather than words. This would mitigate both the problem of missing synonyms (searching for ‘cars’ and failing to return pages containing ‘automobiles’, for example) and the problem with irrelevant synonyms (for example, searching for ‘Java’, meaning the island, and receiving results relating to the computer programming language). Dealing with this second issue would, of course, require enhancements to the user interface, to enable users to choose the meaning they are interested in when selecting a polysemous word.

David Crystal (2006a, 2006b) and his team of lexicographers at Crystal Semantics have developed a classification system that is encyclopaedic in breadth, and hooked up that system to a series of software components using the product name *Textonomy*. These software components compare the text content of web pages against a taxonomic database that identifies proper nouns, abbreviations and common noun senses and assigns them to one of 2,500 categories. This taxonomic database has been built and populated using the encyclopaedias that David Crystal has edited for Cambridge University Press and Penguin Books as a basis, and supplemented by eight years of development and refinement. The search engines of the future will have computer-generated indexes, but the data contained in those indexes may well be driven by datasets produced by human indexing techniques and human linguistic research.

**References**


Misbehaving computers

Jon Jermey

Computers are a tremendous boon to indexers, but – like any other complex equipment – they do occasionally misbehave. If freelance indexers are to meet their deadlines and keep completed work safe they must be able to deal with misbehaving computers. Time spent on keeping a computer up and running is time that cannot be spent indexing. Money spent on computer repairs comes out of indexing income. This article looks at ways to keep your computer affairs running as smoothly as possible.

Nearly all computer problems are soluble with the application of patience, experience and picking other people’s brains. This article lists some of the tactics used by Windows-based computer users, organized under grand strategic plans. If you use a Mac or some other platform the strategies should be the same but specific tactics will vary.

Plan A: Fortify before the battle

It shouldn’t need saying, but it always does: back up, back up, back up. Back up to another hard disk, back up over a network, back up to a memory card, back up to a CD or DVD, back up on to the Web via a program like Mozy (www.mozy.com), back up religiously every hour or so, and store the copies well out of harm’s way! Check the back ups now and then to make sure they are what you think they are. Back up essential programs as well as data. Back up properly and you can laugh at disaster. Use System Restore (see below) to back up system files before making radical changes. And make sure your firewalls and virus protection programs are switched on and up to date. I use and recommend PC-Cillin from Trend Micro (www.trend.com): a good program that keeps getting better.

Plan B: Build up your skills and resources

I use a space-saving computer utility that sometimes offers to clean up my disk by – among other things – removing all the Help files! A sad reflection on how seldom these files get used. Nearly all modern programs come with comprehensive help systems, usually linked to online material, and representing days of work by the program designers and their consultants – often including indexers. Help desk staff estimate that half the calls they get could have been avoided if people had used their built-in help systems. Help is an integral part of the package, and you have paid for it; spend a few minutes learning how to use it before disaster strikes.

The same applies to the big textbook that these days takes the place of a manual. The time to buy that is when you buy the program, and then you can explore the two together, not six months later when you have already got used to using the program in a bumbling, idiosyncratic, half-baked way and cannot be bothered to learn any better. Optimists take pride in learning what their software is capable of – and what it isn’t.

There are also free Web-based tutorials for many programs, where expert users have the fun of showing off their skills to newbies. Take advantage of natural human vanity and brush up your own knowledge with these.

Plan C: Retreat to a safe place

The aim of this strategy is to get the computer back to the way it was when everything was working. Tactic One is to turn the whole thing off and on again. Turning off attachments like the printer and monitor sometimes helps as well. Tactic Two is to use Windows’ built-in System Restore, which recreates the important system files from earlier copies. You will find it under Accessories/System Tools. Tactic Three is