infinitesimal percentage would be a fruitless exercise," the judge concluded. It probably would not be effective to issue a broader injunction, and even if it were, "the risk of unlimited inhibitions of free speech should be avoided when practicable."

The judge understood the gravity of the issue he was deciding. Fundamentally, he was reluctant to use the authority of the government in a futile attempt to prevent people from saying what they wanted to say and finding out what they wanted to know. Even if the documents had been visible only for a short time period, unknown numbers of copies might be circulating privately among interested parties. Grasping for an analogy, the judge suggested that God Himself had failed in His attempt to enjoin Adam and Eve from their pursuit of the truth!

Two sponsored links appeared when we did the search for "zyprexa documents." One was for another lawyer offering his services for Zyprexa-related lawsuits against Lilly. The other, triggered by the word "documents" in our search term, was for Google itself: "Online Documents. Easily share & edit documents online for free. Learn more today. docs.google.com." This was an ironic reminder that the bits are out there, and the tools to spread them are there too, for anyone to use. Thanks to search engines, anyone can find the information they want. Information has exploded out of the shells that used to contain it.

In fact, the architecture of human knowledge has changed as a result of search. In a single decade, we have been liberated from information straightjackets that have been with us since the dawn of recorded history. And many who should understand what has happened, do not. In February 2008, a San Francisco judge tried to shut down the Wikileaks web site, which posts leaked confidential documents anonymously as an aid to whistleblowers. The judge ordered the name "Wikileaks" removed from DNS servers, so the URL "`Wikileaks.org`" would no longer correspond to the correct IP address. (In the guts of the Internet, DNS servers provide the service of translating URLs into IP addresses. See the Appendix.) The publicity that resulted from this censorship attempt made it easy to find various "mirrors"—identical twins, located elsewhere on the Web—by searching for "Wikileaks."

The Fall of Hierarchy

For a very long time, people have been organizing things by putting them into categories and dividing those categories into subcategories. Aristotle tried to classify everything. Living things, for example, were either plants or animals. Animals either had red blood or did not; red-blooded animals were either live-bearers or egg-bearers; live-bearers were either humans or other mammals; egg-bearers either swam or flew; and so on. Sponges, bats, and whales all presented classification enigmas, on which Aristotle did not think he had the last word. At the dawn of the Enlightenment, Linnaeus provided a more useful way of classifying living things, using an approach that gained intrinsic scientific validity once it reflected evolutionary lines of descent.

Our traditions of hierarchical classification are evident everywhere. We just love outline structures. The law against cracking copyright protection (discussed in Chapter 6, "Balance Toppled") is Title 17, Section 1201, paragraph (a), part (1), subpart (A). In the Library of Congress system, every book is in one of 26 major categories, designated by a Roman letter, and these major categories are internally divided in a similar way—B is philosophy, for example, and BQ is Buddhism.

If the categories are clear, it may be possible to use the *organizing* hierarchy to *locate* what you are looking for. That requires that the person doing the searching not only know the classification system, but be skilled at making all the necessary decisions. For example, if knowledge about living things was organized as Aristotle had it, anyone wanting to know about whales would have to know *already* whether a whale was a fish or a mammal in order to go down the proper branch of the classification tree. As more and more knowledge has to be stuffed into the tree, the tree grows and sprouts twigs, which over time become branches sprouting more twigs. The classification problem becomes unwieldy, and the retrieval problem becomes practically impossible.

The system of Web URLs started out as such a classification tree. The site `www.physics.harvard.edu` is a web server, of the physics department, within Harvard University, which is an educational institution. But with the profusion of the Web, this system of domain names is now useless as a way of finding anything whose URL you do not already know.

In 1991, when the Internet was barely known outside academic and government circles, some academic researchers offered a program called "Gopher." This program provided a hierarchical directory of many web sites, by organizing the directories provided by the individual sites into one big outline. Finding things using Gopher was tedious by today's standards, and was dependent on the organizational skills of the contributors. Yahoo! was founded in 1994 as an online Internet directory, with human editors placing products and services in categories,

"Gopher" was a pun—it was software you could use to "go for" information on the Web. It was also the mascot of the University of Minnesota, where the software was first developed.

making recommendations, and generally trying to make the Internet accessible to non-techies. Although Yahoo! has long since added a search window, it retains its basic directory function to the present day.

The practical limitations of hierarchical organization trees were foreseen sixty years ago. During World War II, President Franklin Roosevelt appointed Vannevar Bush of MIT to serve as Director of the Office of Strategic Research and Development (OSRD). The OSRD coordinated scientific research in support of the war effort. It was a large effort—30,000 people and hundreds of projects covered the spectrum of science and engineering. The Manhattan Project, which produced the atomic bomb, was just a small piece of it.

From this vantage point, Bush saw a major obstacle to continued scientific progress. We were producing information faster than it could be consumed, or even classified. Decades before computers became commonplace, he wrote about this problem in a visionary article, "As We May Think." It appeared in the *Atlantic Monthly*—a popular magazine, not a technical journal. As Bush saw it,

> The difficulty seems to be, not so much that we publish unduly ... but rather that publication has been extended far beyond our present ability to make real use of the record. The summation of human experience is being expanded at a prodigious rate, and the means we use for threading through the consequent maze to the momentarily important item is the same as was used in the days of square-rigged ships. ... Our ineptitude in getting at the record is largely caused by the artificiality of systems of indexing.

The dawn of the digital era was at this time barely a glimmer on the horizon. But Bush imagined a machine, which he called a "memex," that would augment human memory by storing and retrieving all the information needed. It would be an "enlarged intimate supplement" to human memory, which can be "consulted with exceeding speed and flexibility."

Bush clearly perceived the problem, but the technologies available at the time, microfilm and vacuum tubes, could not solve it. He understood that the problem of finding information would eventually overwhelm the progress of science in creating and recording knowledge. Bush was intensely aware that civilization itself had been imperiled in the war, but thought we must proceed with optimism about what the record of our vast knowledge might bring us. Man "may perish in conflict before he learns to wield that record for his true good. Yet, in the application of science to the needs and desires of man, it would seem to be a singularly unfortunate stage at which to terminate the process, or to lose hope as to the outcome."

A Futurist Precedent

In 1937, H. G. Wells anticipated Vannevar Bush's 1945 vision of a "memex." Wells wrote even more clearly about the possibility of indexing everything, and what that would mean for civilization:

> *There is no practical obstacle whatever now to the creation of an efficient index to all human knowledge, ideas and achievements, to the creation, that is, of a complete planetary memory for all mankind. And not simply an index; the direct reproduction of the thing itself can be summoned to any properly prepared spot. … This in itself is a fact of tremendous significance. It foreshadows a real intellectual unification of our race. The whole human memory can be, and probably in a short time will be, made accessible to every individual. … This is no remote dream, no fantasy.*

Capabilities that were inconceivable then are commonplace now. Digital computers, vast storage, and high-speed networks make information search and retrieval necessary. They also make it possible. The Web is a realization of Bush's memex, and search is key to making it useful.

It Matters How It Works

How can Google or Yahoo! possibly take a question it may never have been asked before and, in a split second, deliver results from machines around the world? The search engine doesn't "search" the entire World Wide Web in response to your question. That couldn't possibly work quickly enough—it would take more than a tenth of a second just for bits to move around the earth at the speed of light. Instead, the search engine has *already* built up an index of web sites. The search engine does the best it can to find an answer to your query using its index, and then sends its answer right back to you.

To avoid suggesting that there is anything unique about Google or Yahoo!, let's name our generic search engine Jen. Jen integrates several different processes to create the illusion that you simply ask her a question and she gives back good answers. The first three steps have nothing to do with your particular query. They are going on repeatedly and all the time, whether anyone is posing any queries or not. In computer speak, these steps are happening in the *background*:

1. **Gather information.** Jen explores the Web, visiting many sites on a regular basis to learn what they contain. Jen revisits old pages because their contents may have changed, and they may contain links to new pages that have never been visited.
2. **Keep copies.** Jen retains copies of many of the web pages she visits. Jen actually has a duplicate copy of a large part of the Web stored on her computers.
3. **Build an index.** Jen constructs a huge index that shows, at a minimum, which words appear on which web pages.

When you make a query, Jen goes through four more steps, in the *foreground*:

4. **Understand the query.** English has lots of ambiguities. A query like "red sox pitchers" is fairly challenging if you haven't grown up with baseball!
5. **Determine the relevance of each possible result to the query.** Does the web page contain information the query asks about?
6. **Determine the ranking of the relevant results.** Of all the relevant answers, which are the "best"?
7. **Present the results.** The results need not only to be "good"; they have to be shown to you in a form you find useful, and perhaps also in a form that serves some of Jen's other purposes—selling more advertising, for example.

Each of these seven steps involves technical challenges that computer scientists love to solve. Jen's financial backers hope that her engineers solve them better than the engineers of competing search engines.

We'll go through each step in more detail, as it is important to understand what is going on—at every step, more than technology is involved. Each step also presents opportunities for Jen to use her information-gathering and editorial powers in ways you may not have expected—ways that shape your view of the world through the lens of Jen's search results.

The background processing is like the set-building and rehearsals for a theatrical production. You couldn't have a show without it, but none of it happens while the audience is watching, and it doesn't even need to happen on any particular schedule.

Step 1: Gather Information

Search engines don't index everything. The ones we think of as general utilities, such as Google, Yahoo!, and Ask, find information rather indiscriminately throughout the Web. Other search engines are domain-specific. For example, Medline searches only through medical literature. ArtCylopedia indexes 2,600 art sites. The FindLaw LawCrawler searches only legal web sites. Right from the start, with any search engine, some things are in the index and some are out, because some sites are visited during the gathering step and others are not. Someone decides what is worth remembering and what isn't. If something is left out in Step 1, there is no possibility that you will see it in Step 7.

Speaking to the Association of National Advertisers in October 2005, Eric Schmidt, Google's CEO, observed that of the 5,000 terabytes of information in the world, only 170 terabytes had been indexed. (A *terabyte* is about a trillion bytes.) That's just a bit more than 3%, so 97% was not included. Another estimate puts the amount of indexed information at only .02% of the size of the databases and documents reachable via the Web. Even in the limited context of the World Wide Web, Jen needs to decide what to look at, and how frequently. These decisions implicitly define what is important and what is not, and will limit what Jen's users can find.

How *often* Jen visits web pages to index them is one of her precious trade secrets. She probably pays daily visits to news sites such as `CNN.com`, so that if you ask tonight about something that happened this morning, Jen may point you to CNN's story. In fact, there is most likely a master list of sites to be visited frequently, such as `whitehouse.gov`—sites that change regularly and are the object of much public interest. On the other hand, Jen probably has learned from her repeated visits that some sites don't change at all. For example, the Web version of a paper published ten years ago doesn't change. After a few visits, Jen may decide to revisit it once a year, just in case. Other pages may not be posted long enough to get indexed at all. If you post a futon for sale on `Craigslist.com`, the ad will become accessible to potential buyers in just a few minutes. If it sells quickly, however, Jen may never see it. Even if the ad stays up for a while, you probably won't be able to find it with most search engines for several days.

Jen is clever about how often she revisits pages—but her cleverness also codifies some judgments, some priorities—some *control*. The more important Jen judges your page to be, the less time it will take for your new content to show up as responses to queries to Jen's search engine.

Jen roams the Web to gather information by following links from the pages she visits. Software that crawls around the Web is (in typical geek

How a Spider Explores the Web

Search engines gather information by wandering through the World Wide Web. For example, when a spider visits the main URL of the publisher of this book, `www.pearson.com`, it retrieves a page of text, of which this is a fragment:

```
<div id="subsidiary">
<h2 class="hide">Subsidiary sites links</h2>
<label for="subsidiarySites" class="hide">Available
sites</label>
<select name="subsidiarySites" id="subsidiarySites" size="1">
<option value="">Browse sites</option>
<optgroup label="FT Group">
<option value="http://www.ftchinese.com/sc/index.jsp">
  Chinese.FT.com</option>
<option value="http://ftd.de/">FT Deutschland</option>
```

This text is actually a computer program written in a special programming language called HTML ("HyperText Markup Language"). Your web browser renders the web page by executing this little program. But the spider is retrieving this text not to render it, but to index the information it contains. "FT Deutschland" is text that appears on the screen when the page is rendered; such terms should go into the index. The spider recognizes other links, such as `www.ftchinese.com` or `ftd.de`, as URLs of pages it needs to visit in turn. In the process of visiting those pages, it indexes them and identifies yet more links to visit, and so on!

A spider, or web crawler, is a particular kind of *bot.* A bot (as in "robot") is a program that endlessly performs some intrinsically repetitive task, often an information-gathering task.

irony) called a "spider." Because the spidering process takes days or even weeks, Jen will not know immediately if a web page is taken down—she will find out only when her spider next visits the place where it used to be. At that point, she will remove it from her index, but in the meantime, she may respond to queries with links to pages that no longer exist. Click on such a link, and you will get a message such as "Page not found" or "Can't find the server."

Because the Web is unstructured, there is no inherently "correct" order in which to visit the pages, and no obvious way to know when to stop. Page A may contain references to page B, and also page B to page A, so the spider has to be careful not to go around in circles. Jen must organize her crawl of the Web to visit as much as she chooses without wasting time revisiting sections she has already seen.

A web site may stipulate that it does not want spiders to visit it too frequently or to index certain kinds of information. The site's designer simply puts that information in a file named robots.txt, and virtually all web-crawling software will respect what it says. Of course, pages that are inaccessible without a login cannot be crawled at all. So, the results from Step 7 may be influenced by what the sites want Jen to know about them, as well as by what Jen thinks is worth knowing. For example, Sasha Berkovich was fortunate that the Polotsky family tree had been posted to part of the genealogy.com web site that was open to the public—otherwise, Google's spider could not have indexed it.

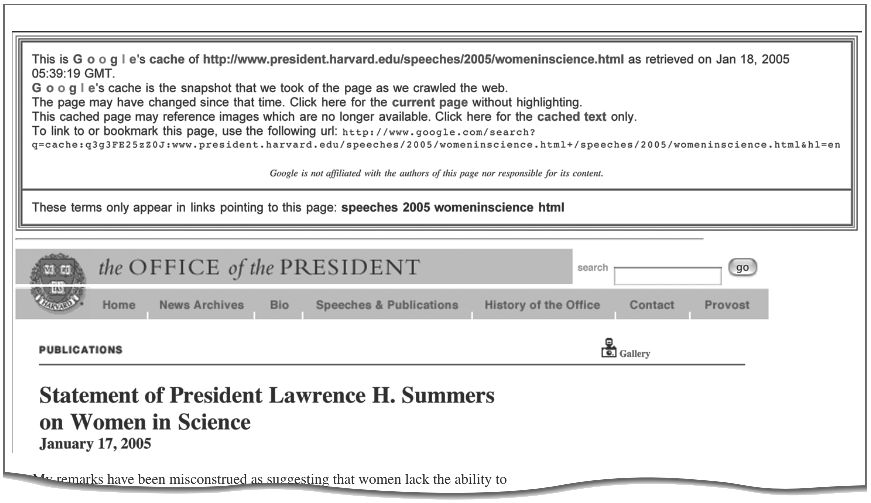
Finally, spidering is not cost free. Jen's "visits" are really requests to web sites that they send their pages back to her. Spidering creates Internet traffic and also imposes a load on the web server. This part of search engines' background processing, in other words, has unintended effects on the experience of the entire Internet. Spiders consume network bandwidth, and they may tie up servers, which are busy responding to spider requests while their ordinary users are trying to view their pages. Commercial search engines attempt to schedule their web crawling in ways that won't overload the servers they visit.

Step 2: Keep Copies

Jen downloads a copy of every web page her spider visits—this is what it means to "visit" a page. Instead of rendering the page on the screen as a web browser would, Jen indexes it. If she wishes, she can retain the copy after she has finished indexing it, storing it on her own disks. Such a copy is said to be "cached," after the French word for "hidden." Ordinarily Jen would not do anything with her cached copy; it may quickly become out of date. But caching web pages makes it possible for Jen to have a page that no longer exists at its original source, or a version of a page older than the current one. This is the flip side of Jen never knowing about certain pages because their owners took them down before she had a chance to index them. With a cached page, Jen knows what used to be on the page even after the owner intended it to disappear.

Caching is another blow to the Web-as-library metaphor, because removing information from the bookshelf doesn't necessarily get rid of it. Efforts to scrub even dangerous information are beyond the capability of those who posted it. For example, after 9/11, a lot of information that was once available on the Web was pulled. Among the pages that disappeared overnight were reports on government vulnerabilities, sensitive security information, and even a Center for Disease Control chemical terrorism report that revealed industry shortcomings. Because the pages had been cached, however, the bits lived on at Google and other search engine companies.

Not only did those pages of dangerous information survive, but anyone could find them. Anytime you do a search with one of the major search engines, you are offered access to the cached copy, as well as the link to where the page came from, whether or not it still exists. Click on the link for the "Cached" page, and you see something that looks very much like what you might see if you clicked on the main link instead. The cached copy is identified as such (see Figure 4.3).



This is Google's cache of **http://www.president.harvard.edu/speeches/2005/womeninscience.html** as retrieved on Jan 18, 2005 05:39:19 GMT.
Google's cache is the snapshot that we took of the page as we crawled the web.
The page may have changed since that time. Click here for the **current page** without highlighting.
This cached page may reference images which are no longer available. Click here for the **cached text** only.
To link to or bookmark this page, use the following url: `http://www.google.com/search?q=cache:q3g3FE25zZ0J:www.president.harvard.edu/speeches/2005/womeninscience.html+/speeches/2005/womeninscience.html&hl=en`

Google is not affiliated with the authors of this page nor responsible for its content.

These terms only appear in links pointing to this page: **speeches 2005 womeninscience html**

Statement of President Lawrence H. Summers on Women in Science
January 17, 2005

My remarks have been misconstrued as suggesting that women lack the ability to

FIGURE 4.3 Part of a cached web page, Google's copy of an official statement made by Harvard's president and replaced two days later after negative public reaction. This copy was retrieved from Google after the statement disappeared from the university's web site. Harvard, which holds the copyright on this once-public statement, refused to allow it to be printed in this book (see Conclusion).

This is an actual example; it was the statement Lawrence Summers released on January 17, 2005, after word of his remarks about women in science became public. As reported in *Harvard Magazine* in March–April 2005, the statement began, "My remarks have been misconstrued as suggesting that women lack the ability to succeed at the highest levels of math and science. I did not say that, nor do I believe it." This unapologetic denial stayed on the

The digital explosion grants the power of both instant communication and instant retraction—but almost every digital action leaves digital fingerprints.

Harvard web site for only a few days. In the face of a national firestorm of protest, Summers issued a new statement on January 19, 2005, reading, in part, "I deeply regret the impact of my comments and apologize for not having weighed them more carefully." Those searching for the President's statement were then led to the contrite new statement—but for a time, the original, defiant version remained visible to those who clicked on the link to Google's cached copy.

Finding Deleted Pages

An easy experiment on finding deleted pages is to search using Google for an item that was sold on craigslist. You can use the "site" modifier in the Google search box to limit your search to the craigslist web site, by including a "modifier":

```
futon site:craigslist.com
```

The results will likely return pages for items that are no longer available, but for which the cached pages will still exist.

The digital explosion grants the power of both instant communication and instant retraction—but almost every digital action leaves digital fingerprints. Bits do not die easily, and digital words, once said, are hard to retract.

If Jen caches web pages, it may be possible for you to get information that was retracted after it was discovered to be in error or embarrassing. Something about this doesn't feel quite right, though—is the information on those pages really Jen's to do with as she wishes? If the material is copyrighted—a published paper from ten years ago, for example—what right does Jen have to show you her cached copy? For that matter, what right did she have to keep a copy in the first place? If you have copyrighted something, don't you have some authority over who can make copies of it?

This enigma is an early introduction to the confused state of copyright law in the digital era, to which we return in Chapter 6. Jen cannot index my web page without receiving a copy of it. In the most literal sense, any time you "view" or "visit" a web page, you are actually copying it, and then your web browser renders the copy on the screen. A metaphorical failure once again: The Web is *not a library*. Viewing is an exchange of bits, not a passive activity, as far as the web site is concerned. If "copying" copyrighted materials was totally prohibited, neither search engines nor the Web itself could work, so some sort of copying must be permissible. On the other hand, when Jen caches the material she indexes—perhaps an entire book, in the case of the

Google Books project—the legal controversies become more highly contested. Indeed, as we discuss in Chapter 6, the Association of American Publishers and Google are locked in a lawsuit over what Google is and is not allowed to do with the digital images of books that Google has scanned.

Step 3: Build an Index

When we searched the Web for "Zyprexa," Jen consulted her index, which has the same basic structure as the index of a book: a list of terms followed by the places they occur. Just as a book's index lists page numbers, Jen's index lists URLs of web pages. To help the search engine give the most useful responses to queries, the index may record other information as well: the size of the font in which the term appears, for example, and where on the page it appears.

Indexes are critical because having the index in order—like the index of a book, which is in alphabetical order—makes it possible to find things much faster than with sequential searching. This is where Jen's computer scientists really earn their salaries, by devising clever ways of storing indexed information so it can be retrieved quickly. Moore's Law also played a big role in the creation of web indexes—until computer memories got fast enough, cheap enough, and big enough, even the cleverest computer scientists could not program machines to respond instantly to arbitrary English queries.

Indexes and Concordances

The information structure used by search engines is technically known as an *inverted index*—that is, an index of the words in a document or a set of documents, and the places where those words appear. Inverted indexes are not a new idea; the biblical concordances laboriously constructed by medieval monks were inverted indexes. Constructing concordances was one of the earliest applications of computer technology to a nonmathematical problem.

When Jen wants to find a term in her index, she does not start at the beginning and go through it one entry at a time until she finds what she is looking for. That is not the way you would look up something in the index of a book; you would use the fact that the index is in order alphabetically. A very simple strategy to look up something in a big ordered index, such as a phone book, is just to open the book in the middle and see if the item you are looking for belongs in the first half or the second. Then you can ignore half the phone book and use the same strategy to subdivide the remaining half. The number of steps it takes to get down to a single page in a phone book with n pages using this method is the number of times you have to divide n by 2 to get down to 1. So if n is 1000, it takes only 10 of these probing steps to find any item using *binary search*, as this method is known.

In general, the number of steps needed to search an index of *n* things using binary search is proportional, not to *n*, but to the number of digits in *n*. That means that binary search is exponentially faster than linear search—searching through a million items would take only 20 steps, and through a billion items would take 30 steps. And binary search is fairly dumb by comparison with what people actually do—if you were looking for "Ledeen" in the phone book, you might open it in the middle, but if you were looking for "Abelson," you'd open it near the front. That strategy can be reduced to an even better computer algorithm, exponentially faster than binary search.

How big is Jen's index, in fact? To begin with, how many terms does Jen index? That is another of her trade secrets. Jen's index could be useful with a few tens of millions of entries. There are fewer than half a million words in the English language, but Jen probably wants to index some numbers too (try searching for a number such as 327 using your search engine). Proper names and at least some words in foreign languages are also important. The list of web pages associated with a term is probably on disk in most cases, with only the information about *where* on the disk kept with the term itself in main memory. Even if storing the term and the location on disk of the list of associated URLs takes 100 bytes per entry, with 25 million entries, the table of index entries would occupy 2.5 gigabytes (about 2.5 billion bytes) of main memory. A few years ago, that amount of memory was unimaginable; today, you get that on a laptop from Wal-Mart. The index can be searched quickly—using binary search, for example—although retrieving the list of URLs might require going to disk. If Jen has Google's resources, she can speed up her query response by keeping URLs in main memory too, and she can split the search process across multiple computers to make it even faster.

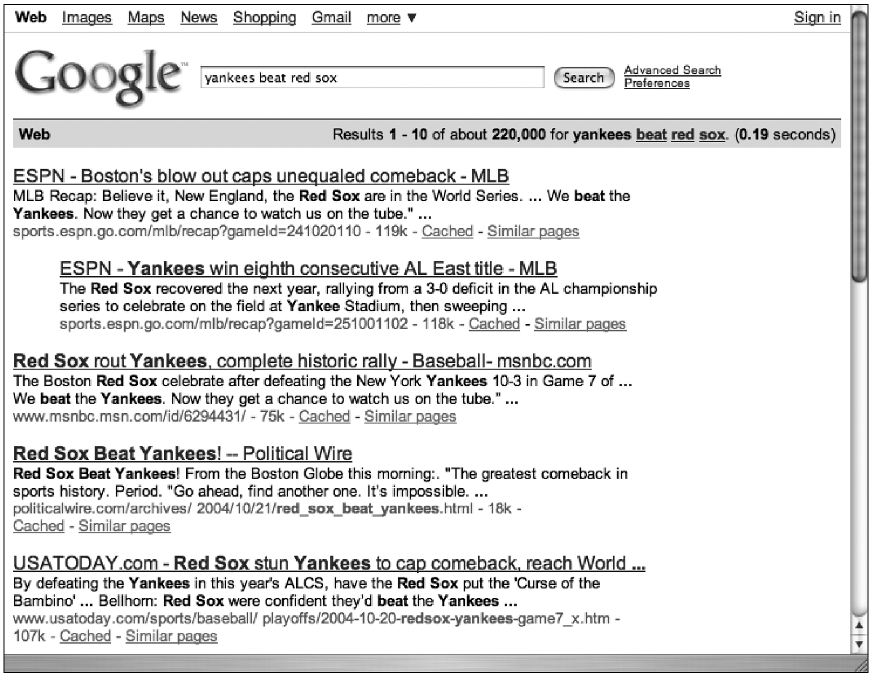
Now that the preparations have been made, we can watch the performance itself—what happens when you give Jen a query.

Step 4: Understand the Query

When we asked Google the query *Yankees beat Red Sox*, only one of the top five results was about the Yankees beating the Red Sox (see Figure 4.4). The others reported instead on the Red Sox beating the Yankees. Because English is hard for computers to understand and is often ambiguous, the simplest form of query analysis ignores syntax, and treats the query as simply a list of keywords. Just looking up a series of words in an index is computationally easy, even if it often misses the intended meaning of the query.

To help users reduce the ambiguity of their keyword queries, search engines support "advanced queries" with more powerful features. Even the simplest, putting a phrase in quotes, is used by fewer than 10% of search engine users. Typing the quotation marks in the query “Red Sox beat Yankees” produces more appropriate results. You can use “~” to tell Google to find synonyms, “-” to exclude certain terms, or cryptic commands such as “allinurl:” or “inanchor:” to limit the part of the Web to search. Arguably we didn’t ask our question the right way, but most of us don’t bother; in general, people just type in the words they want and take the answers they get.

Often they get back quite a lot. Ask Yahoo! for the words “allergy” and “treatment,” and you find more than 20,000,000 references. If you ask for “allergy treatment”—that is, if you just put quotes around the two words—you get 628,000 entries, and quite different top choices. If you ask for “treating allergies,” the list shrinks to 95,000. The difference between these queries may have been unintentional, but the search engine thought they were drastically different. It’s remarkable that human-computer communication through the lens of the search engine is so useful, given its obvious imperfections!



Web Results **1** - **10** of about **220,000** for **yankees** **beat** **red** **sox**. (**0.19** seconds)

ESPN - Boston's blow out caps unequaled comeback - MLB
MLB Recap: Believe it, New England, the **Red Sox** are in the World Series. ... We **beat** the **Yankees**. Now they get a chance to watch us on the tube." ...
sports.espn.go.com/mlb/recap?gameId=241020110 - 119k - Cached - Similar pages

ESPN - **Yankees** win eighth consecutive AL East title - MLB
The **Red Sox** recovered the next year, rallying from a 3-0 deficit in the AL championship series to celebrate on the field at **Yankee** Stadium, then sweeping ...
sports.espn.go.com/mlb/recap?gameId=251001102 - 118k - Cached - Similar pages

Red Sox rout **Yankees**, complete historic rally - Baseball- msnbc.com
The Boston **Red Sox** celebrate after defeating the New York **Yankees** 10-3 in Game 7 of ...
We **beat** the **Yankees**. Now they get a chance to watch us on the tube." ...
www.msnbc.msn.com/id/6294431/ - 75k - Cached - Similar pages

Red Sox Beat Yankees! -- Political Wire
Red Sox Beat Yankees! From the Boston Globe this morning:. "The greatest comeback in sports history. Period. "Go ahead, find another one. It's impossible. ...
politicalwire.com/archives/ 2004/10/21/**red_sox_beat_yankees**.html - 18k -
Cached - Similar pages

USATODAY.com - **Red Sox** stun **Yankees** to cap comeback, reach World ...
By defeating the **Yankees** in this year's ALCS, have the **Red Sox** put the 'Curse of the Bambino' ... Bellhorn: **Red Sox** were confident they'd **beat** the **Yankees** ...
www.usatoday.com/sports/baseball/ playoffs/2004-10-20-**redsox-yankees**-game7_x.htm -
107k - Cached - Similar pages

Google ™ is a registered trademark of Google, Inc. Reprinted by permission.

FIGURE 4.4 Keyword search misses the meaning of English-language query. Most of the results for the query “Yankees beat Red Sox” are about the Red Sox beating the Yankees.

Natural Language Queries

Query-understanding technology is improving. The experimental site www.digger.com, for example, tells you when your query is ambiguous and helps you clarify what you are asking. If you ask Digger for information about "java," it realizes that you might mean the beverage, the island, or the programming language, and helps get the right interpretation if it guessed wrong the first time.

Powerset (www.powerset.com) uses natural language software to disambiguate queries based on their English syntax, and answers based on what web pages actually say. That would resolve the misunderstanding of "Yankees beat Red Sox."

Ongoing research promises to transfer the burden of disambiguating queries to the software, where it belongs, rather than forcing users to twist their brains around computerese. Natural language understanding seems to be on its way, but not in the immediate future. We may need a hundred-fold increase in computing power to make semantic analysis of web pages accurate enough so that search engines no longer give boneheaded answers to simple English queries.

Today, users tend to be tolerant when search engines misunderstand their meaning. They blame themselves and revise their queries to produce better results. This may be because we are still amazed that search engines work at all. In part, we may be tolerant of error because in web search, the cost to the user of an inappropriate answer is very low. As the technology improves, users will expect more, and will become less tolerant of wasting their time sorting through useless answers.

Step 5: Determine Relevance

A search engine's job is to provide results that match the intent of the query. In technical jargon, this criterion is called "relevance." Relevance has an objective component—a story about the Red Sox beating the Yankees is only marginally responsive to a query about the Yankees beating the Red Sox. But relevance is also inherently subjective. Only the person who posed the query can be the final judge of the relevance of the answers returned. In typing my query, I probably meant the New York Yankees beating the Boston Red Sox of Major League Baseball, but I didn't say that—maybe I meant the Flagstaff Yankees and the Continental Red Sox of Arizona Little League Baseball.

Finding all the relevant documents is referred to as “recall.” Because the World Wide Web is so vast, there is no reasonable way to determine if the search engine is finding everything that is relevant. Total recall is unachievable—but it is also unimportant. Jen could give us thousands or even millions more responses that she judges to be relevant, but we are unlikely to look beyond the first page or two. Degree of relevance always trumps level of recall. Users want to find a few good results, not all possible results.

Search Engines and Information Retrieval

Three articles offer interesting insights into how search engines and information retrieval work:

“The Anatomy of a Large-Scale Hypertextual Web Search Engine” by Sergey Brin and Larry Page was written in 2000 and gives a clear description of how the original Google worked, what the goal was, and how it was differentiated from earlier search engines.

“Modern Information Retrieval: A Brief Overview” by Amit Singhal was written in 2001 and surveys the IR scene. Singhal was a student of Gerry Salton and is now a Google Fellow.

“The Most Influential Paper Gerald Salton Never Wrote” by David Dubin presents an interesting look at some of the origins of the science.

The science of measuring relevance is much older than the Web; it goes back to work by Gerald Salton in the 1960s, first at Harvard and later at Cornell. The trick is to automate a task when what counts as success has such a large subjective component. We want the computer to scan the document, look at the query, do a few calculations, and come up with a number suggesting how relevant the document is to the query.

As a very simple example of how we might calculate the relevance of a document to a query, suppose there are 500,000 words in the English language. Construct two lists of 500,000 numbers: one for the document and one for the query. Each position in the lists corresponds to one of the 500,000 words—for example, position #3682 might be for the word “drugs.” For the document, each position contains a count of the number of times the corresponding word occurs in the document. Do the same thing for the query—unless it contains repeated words, each position will be 1 or 0. Multiply the lists for the document and the query, position by position, and add up the 500,000 results. If no word in the query appears in the document, you’ll get a result of 0; otherwise, you will get a result greater than 0. The more frequently words from the query appear in the document, the larger the results will be.

Figure 4.5 shows how the relevance calculation might proceed for the query "Yankees beat Red Sox" and the visible part of the third document of Figure 4.4, which begins, "Red Sox rout Yankees ...." (The others probably contain more of the keywords later in the full document.) The positions in the two lists correspond to words in a dictionary in alphabetical order, from "ant" to "zebra." The words "red" and "sox" appear two times each in the snippet of the story, and the word "Yankees" appears three times.

Lexicon:	ant, …,	beat, …,	defeating, …,	new, …,	patriots, …,	red, …,	sox, …,	Yankees, …,	zebra, …
Doc:	0, …,	1, …,	2, …,	1, …,	0, …,	2, …,	2, …,	3, …,	0, …
Query:	0, …,	1, …,	0, …,	0, …,	0, …,	1, …,	1, …,	1, …,	0, …
Doc × Query	0, …,	1, …,	0, …,	0, …,	0, …,	2, …,	2, …,	3, …,	0, …

Sum of elements of Doc × Query = 1+2+2+3 = 8 = "relevance" of document to query

FIGURE 4.5 Document and query lists for relevance calculation.

That is a very crude relevance calculation—problems with it are easy to spot. Long documents tend to be measured as more relevant than short documents, because they have more word repetitions. Uninteresting words such as "from" add as much to the relevance score as more significant terms such as "Yankees." Web search engines such as Google, Yahoo!, MSN, and Ask.com consider many other factors in addition to which words occur and how often. In the list for the document, perhaps the entries are not word counts, but another number, adjusted so words in the title of the page get greater weight. Words in a larger font might also count more heavily. In a query, users tend to type more important terms first, so maybe the weights should depend on where words appear in the query.

Step 6: Determine Ranking

Once Jen has selected the relevant documents—perhaps she's chosen all the documents whose relevance score is above a certain threshold—she "ranks" the search results (that is, puts them in order). Ranking is critical in making the search useful. A search may return thousands of relevant results, and users want to see only a few of them. The simplest ranking is by relevance—putting the page with the highest relevance score first. That doesn't work well, however. For one thing, with short queries, many of the results will have approximately the same relevance.

More fundamentally, the documents Jen returns should be considered "good results" not just because they have high relevance to the query, but also because the documents themselves have high quality. Alas, it is hard to say what "quality" means in the search context, when the ultimate test of success is providing what people want. In the example of the earlier sidebar, who is to judge whether the many links to material about Britney Spears are really "better" answers to the "spears" query than the link to Professor Spears? And whatever "quality" may be, the ranking process for the major web search engines takes place automatically, without human intervention. There is no way to include protocols for checking professional licenses and past convictions for criminal fraud—not in the current state of the Web, at least.

Even though quality can't be measured automatically, something like "importance" or "reputation" can be extracted from the structure of linkages that holds the Web together. To take a crude analogy, if you think of web pages as scientific publications, the reputations of scientists tend to rise if their work is widely cited in the work of other scientists. That's far from a

What Makes a Page Searchable

No search provider discloses the full details of its relevance and ranking algorithm. The formulas remain secret because they offer competitive advantages, and because knowing what gives a page high rank makes abuse easier. But here are some of the factors that might be taken into account:

- Whether a keyword is used in the title of the web page, a major heading, or a second-level heading
- Whether it appears only in the body text, and if so, how "prominently"
- Whether the web site is considered "trustworthy"
- Whether the pages linked to from within the page are themselves relevant
- Whether the pages that link to this page are relevant
- Whether the page is old or young
- Whether the pages it links to are old or young
- Whether it passes some objective quality metric—for example, not containing any misspellings

Once you go to the trouble of crawling the Web, there is plenty to analyze, if you have the computing power to do it!

perfect system for judging the importance of scientific work—junk science journals do exist, and sometimes small groups of marginal scientists form mutual admiration societies. But for the Web, looking at the linkage structure is a place to start to measure the significance of pages.

One of Google's innovations was to enhance the relevance metric with another numerical value called "PageRank." PageRank is a measure of the "importance" of each a page that takes into account the external references to it—a World Wide Web popularity contest. If more web pages link to a particular page, goes the logic, it must be more important. In fact, a page should be judged more important if a lot of *important* pages link to it than if the same number of unimportant pages link to it. That seems to create a circular definition of importance, but the circularity can be resolved—with a bit of mathematics and a lot of computing power.

This way of ranking the search results seems to reward reputation and to be devoid of judgment—it is a mechanized way of aggregating mutual opinions. For example, when we searched using Google for "schizophrenia drugs," the top result was part of the site of a Swedish university. Relevance was certainly part of the reason that page came up first; the page was specifically about drugs used to treat schizophrenia, and the words "schizophrenia" and "drugs" both appeared in the title of the page. Our choice of words affected the relevance of the page—had we gone to the trouble to type "medicines" instead of "drugs," this link wouldn't even have made it to the first page of search results. Word order matters, too—Google returns different results for "drugs schizophrenia" than for "schizophrenia drugs."

This page may also have been ranked high because many other web pages contained references to it, particularly if many of those pages were themselves judged to be important. Other pages about schizophrenia drugs may have used better English prose style, may have been written by more respected scientific authorities, and may have contained more up-to-date information and fewer factual errors. The ranking algorithm has no way to judge any of that, and no one at Google reads every page to make such judgments.

Sergey Brin and Larry Page, Google's founders, were graduate students at Stanford when they developed the company's early technologies. The "Page" in "PageRank" refers not to web pages, but to Larry Page.

Google, and other search engines that rank pages automatically, use a secret recipe for ranking—a pinch of this and a dash of that. Like the formula for Coca-Cola, only a few people know the details of commercial ranking algorithms. Google's algorithm is patented, so anyone can read a description. Figure 4.6 is an illustration from that patent, showing several pages with links to each other. This illustration suggests that both the documents themselves and the links between them might be assigned varying numbers as measures of their importance. But the description omits many details and, as actually implemented, has been adjusted countless times to improve its performance. A company's only real claim for the validity of its ranking formula is that people like the results it delivers—if they did not, they would shift to one of the competing search engines.

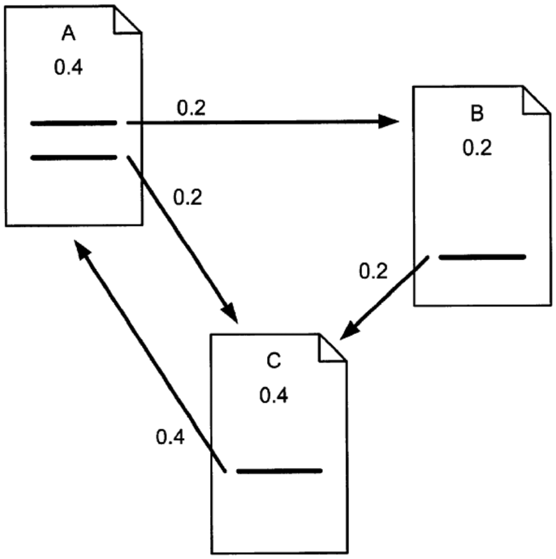


FIGURE 4.6 A figure from the PageRank patent (U.S. Patent #6285999), showing how links between documents might receive different weights.

It may be that one of the things people like about their favored search engine is consistently getting what they believe to be unbiased, useful, and even truthful information. But "telling the truth" in search results is ultimately only a means to an end—the end being greater profits for the search company.

Ranking is a matter of opinion. But a lot hangs on those opinions. For a user, it usually does not matter very much which answer comes up first or whether any result presented is even appropriate to the query. But for a company offering a product, where it appears in the search engine results *can* be a matter of life and death.

KinderStart (www.kinderstart.com) runs a web site that includes a directory and search engine focused on products and services for young children. On March 19, 2005, visits to its site declined by 70% when Google lowered its PageRank to zero (on a scale of 0 to 10). Google may have deemed KinderStart's page to be low quality because its ranking algorithm found the page to consist mostly of links to other sites. Google's public description of its criteria warns about pages with "little or no original content." KinderStart saw matters differently and mounted a class action lawsuit against Google, claiming, among other things, that Google had violated its rights to free speech under the First Amendment by making its web site effectively invisible. Google countered that KinderStart's low PageRank was just Google's opinion, and opinions were not matters to be settled in court:

> Google, like every other search engine operator, has made that determination for its users, exercising its judgment and expressing its opinion about the relative significance of web sites in a manner that has made it the search engine of choice for millions. Plaintiff KinderStart contends that the judiciary should have the final say over that editorial process.

No fair, countered KinderStart to Google's claim to be just expressing an opinion. "PageRank," claimed KinderStart, "is not a mere statement of opinion of the innate value or human appeal of a given web site and its web pages," but instead is "a mathematically-generated product of measuring and assessing the quantity and depth of all the hyperlinks on the Web that tie into PageRanked web site, under programmatic determination by Defendant Google."

Seeing a Page's PageRank

Google has a toolbar you can add to certain browsers, so you can see PageRanks of web pages. It is downloadable from toolbar.google.com. You can also use the site www.iwebtool.com/pagerank_checker to enter a URL in a window and check its PageRank.

The judge rejected every one of KinderStart's contentions—and not just the claim that KinderStart had a free speech right to be more visible in Google searches. The judge also rejected claims that Google was a monopoly guilty of antitrust violations, and that KinderStart's PageRank of zero amounted to a defamatory statement about the company.

Whether it's a matter of opinion or manipulation, KinderStart is certainly much easier to find using Yahoo! than Google. Using Yahoo!, `kinderstart.com` is the top item returned when searching for "kinderstart." When we used Google, however, it did not appear until the twelfth page of results.

A similar fate befell `bmw.de`, the German web page of automaker BMW. The page Google indexed was straight text, containing the words "gebrauchtwagen" and "neuwagen" ("used car" and "new car") dozens of times. But a coding trick caused viewers instead to see a more conventional page with few words and many pictures. The effect was to raise BMW's position in searches for "new car" and "used car," but the means violated Google's clear instructions to web site designers: "Make pages for users, not for search engines. Don't deceive your users or present different content to search engines than you display to users, which is commonly referred to as 'cloaking.'" Google responded with a "death penalty"—removing `bmw.de` from its index. For a time, the page simply ceased to exist in Google's universe. The punitive measure showed that Google was prepared to act harshly against sites attempting to gain rank in ways it deemed consumers would not find helpful—and at the same time, it also made clear that Google was prepared to take *ad hoc* actions against individual sites.

Step 7: Presenting Results

After all the marvelous hard work of Steps 1–6, search engines typically provide the results in a format that is older than Aristotle—the simple, top-to-bottom list. There are less primitive ways of displaying the information.

If you search for something ambiguous like "washer" with a major web search engine, you will be presented with a million results, ranging from clothes washers to software packages that remove viruses. If you search Home Depot's web site for "washer," you will get a set of automatically generated choices to assist you in narrowing the search: a set of categories, price ranges, brand names, and more, complete with pictures (see Figure 4.7).

Alternatives to the simple rank-ordered list for presenting results better utilize the visual system. Introducing these new forms of navigation may shift the balance of power in the search equation. Being at the top of the list may no longer have the same economic value, but something else may replace the currently all-important rank of results—quality of the graphics, for example.

No matter how the results are presented, something else appears alongside them, and probably always will. It is time to talk about those words from the sponsors.