MT834

Unit 4

Web indexing and search engines
Course team
Developer: Jenny Lim, Consultant
Designer: Chris Baker, OUHK
Coordinator: Dr Li Tak Sing, OUHK
Member: Dr Andrew Lui Kwok Fai, OUHK

External Course Assessor
Prof. Mingshu Li, Institute of Software, Chinese Academy of Sciences

Production
ETPU Publishing Team

Copyright © The Open University of Hong Kong, 2004.
All rights reserved.
No part of this material may be reproduced in any form by any means without permission in writing from the President, The Open University of Hong Kong.

The Open University of Hong Kong
30 Good Shepherd Street
Ho Man Tin, Kowloon
Hong Kong
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>1</td>
</tr>
<tr>
<td>Objectives</td>
<td>2</td>
</tr>
<tr>
<td>Introduction</td>
<td>3</td>
</tr>
<tr>
<td>How search engines work</td>
<td>4</td>
</tr>
<tr>
<td>Components of a search engine</td>
<td>6</td>
</tr>
<tr>
<td>Registering your site</td>
<td>16</td>
</tr>
<tr>
<td>Submitting to search engines</td>
<td>17</td>
</tr>
<tr>
<td>Submitting to search directories</td>
<td>18</td>
</tr>
<tr>
<td>Optimizing website rankings</td>
<td>20</td>
</tr>
<tr>
<td>Difficulties in indexing the Web</td>
<td>22</td>
</tr>
<tr>
<td>Nature of Web content</td>
<td>23</td>
</tr>
<tr>
<td>The deep Web</td>
<td>23</td>
</tr>
<tr>
<td>Providing a search facility</td>
<td>26</td>
</tr>
<tr>
<td>Local or remote</td>
<td>26</td>
</tr>
<tr>
<td>Parts of a local search engine</td>
<td>27</td>
</tr>
<tr>
<td>Search gateway programs</td>
<td>30</td>
</tr>
<tr>
<td>Databases</td>
<td>32</td>
</tr>
<tr>
<td>Summary</td>
<td>35</td>
</tr>
<tr>
<td>Feedback to activities</td>
<td>36</td>
</tr>
<tr>
<td>Suggested answers to self-tests</td>
<td>38</td>
</tr>
<tr>
<td>References</td>
<td>42</td>
</tr>
</tbody>
</table>
Overview

Search engines and directories are the tools most frequently used to locate and retrieve information on the Web. With their user-friendly, graphical, point-and-click interfaces, it’s no wonder that they are among the most frequently visited websites.

As Web information providers, it is important for us to understand how search engines and directories work: how they collect documents, index these documents, process search terms and generate the results. We also need to understand the criteria used for ranking search results. We can use this knowledge to increase the chance that our pages will appear near the top of the rankings.

As Web users, we are often frustrated by the inaccuracy and irrelevancy of the results that we get from search engines, despite the high number of results that are returned to us. We will look at the difficulties that are inherent in indexing the Web due to its massive size, lack of cohesiveness and the nature of the content itself.

We will also talk about the so-called ‘deep Web’ — the portion of the Web that is not covered by search engines. According to some estimates, as much as half of all Web documents are inaccessible to search engines (BrightPlanet 2001).

Finally, you will gain hands-on experience in providing search services on your own website. Most e-commerce sites nowadays offer online searching of their product catalogue or website content in order to facilitate navigation by visitors, and we will implement a similar service for ABC Books’ online catalogue.

This unit should take you about four weeks or 30–35 hours to complete. Please plan your time carefully.
Objectives

1. *Explain* how Web documents are indexed by search engines, and *explain* why it is inherently difficult to index the Web completely and efficiently.

2. *Describe* the components of a Web search application.

3. *Discuss* the steps in listing a website with a search engine and *recommend* ways to optimize search engine rankings.

4. *Demonstrate* how search engines can be implemented using gateway programs and databases.

5. *Implement, install* and *configure* a local Web search facility.
Introduction

When the Web first appeared in early 1990s, it introduced a new and convenient way of distributing information to a global audience. Suddenly, anyone who knew how to create and upload pages to a Web server could become a Web publisher.

However, one consequence of this massive increase in Web publication activity is that we are overwhelmed with all sorts of information. Retrieving high-quality, timely and accurate information from the Web is a challenging task.

Unless users know the exact location or URL they’re visiting, they often rely on a directory or search engine to find the information they want. In this case, users will go to a search website, submit a query that is typically a list of keywords, and then receive a list of relevant webpages that contain the keywords entered. Directories and search engines are analogous to the cataloguing and indexing services that are available in physical libraries.

**Figure 4.1** A typical entry for a book in a library card catalogue


Search engines do not search the actual documents on the Web every time a query is submitted. For the sake of speed and efficiency, they go through index files that contain stored information about the documents being searched. Therefore, the performance of a search engine is determined by the quality and freshness of its index files.

A Web index is like a card catalogue in a physical library. Unlike a physical library, however, the Web is a decentralized information resource where content can be added, updated and taken offline by individual owners at will. One source even states that ‘The current state of search engines can be compared to a phone book which is updated irregularly, and has most of the pages ripped out’ (Lawrence and Giles 1998). On the other hand, although Web index files do not always
contain the latest information, they are frequently updated in increments so that their contents are reasonably accurate and current.

Algorithms for search, indexing and ranking results are constantly being improved by search providers in order to increase the accuracy and quality of search results. More and more information providers are also implementing local search services on their website, in addition to menu-based searching and hyperlinking. It’s quite clear that search technologies are now vital and necessary for assisting users as they navigate and access information on the Web.

How search engines work

The World Wide Web is currently estimated to have over 3 billion webpages stored on 44 million domains (Netcraft 2003). Search engines provide users with a single starting point where they can begin searching for information online. When you begin a Web search, however, search engines do not actually search all the pages on every individual Web server for the information you want. Instead, search engines search through a database that contains information taken from webpages. This database is called an index, and webpages that are included in a search engine’s database are said to be indexed by that search engine. A Web index performs the same function as the keyword index at the back of a textbook. In the same way that an index in a textbook contains a list of words from the textbook, along with the page numbers where these words can be found, a Web index contains a list of words extracted from webpages, along with the URLs of these pages.

There are two main ways that a Web index is built: through machine indexing and human indexing. Search engines are, therefore, mainly classified according to whether they use automated computer programs or humans to compile their indexes. The following reading describes the different types of search engines and gives a brief description of how they work.

Reading 4.1


Note: You don’t need to click on the ‘For more information’ links at the bottom of the page.

From Reading 4.1, you can see that the main distinction between different types of search engines is whether they use an automated program (e.g. crawler, spider or robot) to collect pages for indexing or whether they rely on humans to catalogue and classify the pages that are submitted to them.
Crawler-based search engines are simply known as search engines, while human-powered search engines are also known as portal directories, subject directories or simply search directories.

Google is the best-known search engine nowadays. As the amount of human input required to gather the information and index the pages is minimal for search engines, most are capable of covering many more webpages than human-powered search directories. For example, Google claims it has collected information about more than 2 billion webpages in its database. It also claims that its entire index is refreshed every 28 days (ComputerWire).

Yahoo is probably the most famous portal directory. Since human effort is required to review and catalogue sites in portal directories, they usually only cover a very small portion of the Web. This limitation has forced Yahoo to include results from a search engine — Google.

In fact, you’ll find that most portal directories also provide search engine services, and most search engines also provide portal directory services. We say that Yahoo is a portal directory because it is well-known for providing this service. Similarly, we say that Google is a search engine because it is famous for providing a search engine service.

To better understand how portal directories and search engines function, let’s compare them with indexes in a library. Most libraries maintain at least the following indexes for the books they collect:

- an index that sorts according to the names of authors; and
- an index that sorts according to the categories of books.

A library index that classifies materials according to subject categories is very similar to a portal directory. Just like a category index, users of portal directories must choose the appropriate category from which they will begin drilling down in order to find the website they want.

On the other hand, a library index that sorts according to the names of the authors is closer to a search engine in which the index can be generated automatically. Users must think of a keyword or list of keywords that will return the correct result. In this case, the user must provide the author’s full name or a portion of it in order to begin the search.

Now let’s try out a search engine and a search directory in the next activity and compare the two.
Activity 4.1

Look for relevant materials on the following topic using Yahoo’s directory (http://dir.yahoo.com) and Google (http://www.google.com):

hiking and trekking vacations

Describe the actions you took on both Yahoo and Google in order to arrive at your results. You can also evaluate the quality and relevance of the results returned by both search tools.

Note: There is feedback on this activity at the back of this unit.

From the previous activity, you may have noticed that search engines and search directories also organize and present information differently. Search directories aim to organize Web information resources in a hierarchy of categories that can be browsed, while search engines present a list of relevant URLs after processing the keywords entered. Now do the following self-test to assess your understanding of how these tools should be used.

Self-test 4.1

Describe when you think it would be suitable to use a search engine for a particular search, and when it would be suitable to use a search directory.

Search engines are one of the most popular means of finding websites, with approximately 85% of users using a search engine to find information (GVU Center 1998). It’s important for information providers to know how search engines work since a lot of their website traffic will probably come from a search engine referral. The next section will focus on the components that make up a typical crawler-based search engine.

Components of a search engine

Reading 4.1 has given you a basic overview of how a search engine works, and now we’ll take a closer look at how the different parts of a search engine work together. Here are the search engine components that we will look at:

• a gatherer which retrieves webpages for indexing;
• a page repository where retrieved pages are stored (perhaps temporarily);
• an indexer which creates indexes based on the words extracted from the visited pages; and
- a query engine and interface which receives and fulfills search requests from users.

Figure 4.2 shows how these different components may be put together in a typical search engine.

![General search engine architecture](image)

**Figure 4.2** General search engine architecture

*Source: Based on Arasu et al. 2001.*

**Gatherer module**

The main task of this module is to fetch webpages for inclusion into a search index. As mentioned in Reading 4.1, the program responsible for visiting and collecting different webpages is called a robot, spider or crawler. Crawlers are small programs that browse the Web on the search engine’s behalf, in the same way that a human user surfs the Web by following a series of links.

Crawlers are given a starting set of URLs whose contents they should retrieve (figure 4.2). These URLs may have been submitted to the search engine, or they can be heavily used servers and popular pages. In this manner, the crawler quickly begins to travel across the most widely used portions of the Web.
A crawler usually begins its visit from a website’s homepage, and then selects the next document to traverse by following a hypertext link. The crawler keeps following successive links until an exit condition occurs. The exit conditions are usually the time elapsed during the traversal or the number of levels of hyperlinks that have been visited.

In some cases, crawlers may extract URLs appearing in the retrieved pages and submit this information to a separate, crawler control module (Figure 4.2). The control module determines what links to visit next and informs the crawler accordingly.

It’s possible for your website to be included in a search engine even if you have never submitted your pages before. A crawler may arrive at your site by following links from another page. This is good news for websites that do want their site to be listed, but there may be cases when website owners may not want their pages to be indexed just yet. For example, there may be pages that:

- are ‘under construction’ and are not yet ready for public viewing;
- contain sensitive or confidential information meant for a limited audience only; or
- exist in a format which is unsuitable for a robot (such as audio or video).

Fortunately, website owners can communicate with the robots or spiders that come a-visiting using the Robots Exclusion Protocol. With this protocol, they can specify which pages should or should not be indexed. There are two ways to do this: via META tags within the HTML and via a special file called robots.txt that is placed on the Web server. In the next activity, you will use both these methods on your ABC Books’ website.

---

**Activity 4.2**

Read the following two articles before you answer the questions in this activity. These documents describe how you can give specific instructions to robots that may visit your site:


Based on these readings, write the indexing rules that you should specify to the robots that come to your ABC Books’ website:

1. Use robots.txt to restrict spiders from indexing the files in the author and books directory.
2. Use the robots META tag to restrict spiders from crawling any files that authors.html links to.

Note: There is feedback on this activity at the back of the unit.

When a robot visits a webpage, it may visit all the other pages on the same server by following links from the first page visited. This action is called a deep crawl. If a search engine deep crawls your site, then it’s enough for you to submit your site’s homepage. Otherwise, you have to explicitly submit all the pages that you want indexed. As of this writing, All the Web, Google and Inktomi deep crawl the sites that they visit, while AltaVista does not (Sullivan 2002).

Robots must also observe a policy for limiting the traversal of a website. Most search engine companies treat their crawling strategy as a trade secret and do not normally reveal detailed information about it. However, there are two traversal strategies taken from graph theory that may be used: breadth first and depth first. These algorithms are well-suited to the Web due to its graph-like structure which consists of nodes (i.e. webpages) and links (i.e. hypertext).

A breadth-first strategy will first traverse all the hypertext links found in the initial URL, gather up these documents, then examine this group of gathered documents for further links to follow. In other words, pages are crawled in the order that they are discovered. Breadth-first gathering results in a wide and shallow traverse.

![Figure 4.3 Breadth-first traversal](image)

The depth-first strategy starts from the initial URL, then keeps following successive links at ever-increasing depths. Usually there is a limit to the number of links followed using depth first. The result is that the depth-first gatherer does a narrow but deep traverse.
Activity 4.3

Compare the results of breadth-first and depth-first traversal on ABC Books’ website, given that the traversal must stop after five links have been explored. The following site diagram shows the links that appear on the homepage.

Which method is more likely to generate a better quality index for the site?

Note: There is feedback on this activity at the back of the unit.

Given the enormous size of the Web and the frequency of updates of information on the Web, here are some of the questions that arise when setting a crawling policy for a search engine (Arasu et al. 2001):
1 What pages should the crawler download?

The most comprehensive search engines can only cover a fraction of the entire Web. Crawlers must have a policy for selecting and prioritizing the URLs that they visit, so that the portion of the Web that they select for indexing is kept more meaningful and up-to-date.

2 How should the crawler refresh pages?

Because webpages change at different rates, the crawler must carefully decide which pages to visit and which pages can be skipped so as to avoid wasting time and resources. For example, if a certain page rarely changes, the crawler may want to revisit this page less often and concentrate its time on visiting other, more frequently updated pages.

3 How should the load on the visited websites be minimized?

When a crawler gathers pages on the Web, it consumes resources belonging to other organizations, such as memory, CPU and bandwidth. Responsible and well-designed crawlers must minimize their impact on these resources as they go about their work.

4 How should the crawling process be parallelized?

Due to the enormous size of the Web, crawlers often run on multiple machines and download pages in parallel. This allows search engines to download a large number of pages in a reasonable amount of time. However, parallel crawlers must be coordinated properly so that different crawlers do not visit the same website multiple times.

As you can see, responsible crawlers must observe ethical and responsible behaviour as they traverse the Web. They should ensure that their visits have minimum impact on website performance. They can also avoid unnecessary visits in the first place, by keeping track of dead links and pages which are seldom updated.

The next reading offers some useful guidelines for robot designers. Although there is no way to enforce these guidelines, they can be used to identify what is acceptable behaviour and what is not.

---

**Reading 4.2**


Now do the following self-test in order to test your understanding of the Web crawling process and the tags that are used to issue instructions to crawlers.
Self-test 4.2

1 List six guidelines that should be observed by a responsible robot writer.

2 What are the differences between the directives nofollow and noindex in the robots META tag?

3 When should you use robots.txt or the robots META tag?

4 Can robot.txt or the robot META tag actually deny access to robots if they still insist on accessing ‘forbidden’ directories?

Page repository

The page repository is a storage system for managing large collections of webpages. The repository performs two functions: (1) it allows the crawler to store the collected pages; and (2) it allows the indexer to retrieve pages for indexing.

Pages may be stored in the repository only temporarily during the crawling and indexing process. They may also be used to cache collected pages so that the search engine can serve out result pages very quickly. Due to the vast quantities of documents that must be kept in the page repository, special consideration must be given to scalability and storage distribution issues during its design and implementation.

Indexer module

The gatherer fetches documents and submits them to an indexer. The indexer then assigns each document a unique identifier (called the primary key) and creates a record for it. This record contains the unique identifier, the URL of the document, and a set of values or related terms describing the document.

The indexer also extracts words from each page and records the URL where each word occurs, along with its location within the page. The result is generally a very large ‘lookup’ table, also called an inverted index or text index (see figure 4.2). The text index can provide all the URLs where a given word occurs.

Some indexers may index every single word on the page (i.e. full-text indexing), while some may select words that occur in important areas, such as the title, headings, subheadings, links, and the first few lines of text on a page. The selection criteria vary from one search engine to another, which explains why they return different results.

For example, Lycos keeps track of the words in the title, subheadings, links, and words that appear in the first few lines of the text. It also looks
at the most frequently used words on a page. Altavista indexes every single word on a page (SearchEngineWatch.com).

Using META tags, the owner of a page can also specify the keywords for which the page should be indexed. However, not all search engines give the same importance to META tags, because webpage authors may be tempted to manipulate META tags in the hopes of getting a higher ranking.

```
<HEAD>
<TITLE>Stamp Collecting World</TITLE>
<META name="title" content="It's A World of Stamps">
<META name="description" content="Information about collecting stamps, from prices to history.">
<META name="keywords" content="stamps, stamp collecting, stamp history, prices, philatelists, buying and selling stamps">
</HEAD>
```

**Figure 4.6** Possible META tags for a page on stamp collecting

Keywords in META tags have declined in importance over the years. They are still included in webpages out of habit and perhaps out of fear that search engines may start noticing them again. Today, they are mainly used to specify the description which should be returned in search engine result listings (figure 4.7). Some authors have even recommended just using the title and description META tags, and doing away with the keywords entirely.

```
Coin and Stamp Collecting

Comprehensive resources on coin and stamp collecting, paper currency collecting and numismatics and philately in general, including rare and ancient coins and ...
```

**Figure 4.7** How the description attribute of a META tag appears in a search engine listing

Aside from the text index, the indexer module can also build other types of indexes. These other indexes are used to enhance the quality and relevance of search results beyond what can be achieved through text-based indexing alone. For example, Google keeps information about the links between pages in its structure index, because this information may be used to rank search results later on. Utility indexes may also be used to provide access to pages of a given length, pages of a certain ‘importance’, or pages with some number of images in them (figure 4.2).

Indexing is the key component of a search engine. An effective indexing process will yield a high-quality index that accurately represents the collection of information resources. Searching a high-quality index is more likely to result in the precise identification and retrieval of the correct resources.

Due to the transient nature of most Web content, indexes must be constantly updated in order to maintain the freshness and relevance of
content. Different search engines have different crawling schedules, and it’s possible that there will be a delay between the time that new pages are added to the Web or old pages are modified and the time when these pages are re-indexed by various search engines.

**Self-test 4.3**

List some of the ways in which the indexes of different search engines may vary.

---

**Query engine and interface**

The query engine is the component of the search engine that users see and interact with. It performs two major tasks: (1) it searches through the index to find matches to a search; and (2) it ranks the retrieved records in the order that it believes is the most relevant.

The criteria for selection (or rejection) of search terms and assigning weight to them depends on the policy of the search engine concerned, as does the specific information that is stored along with each keyword — such as where in a given webpage it occurred, how many times it occurred, the attached weight, and so on. Each search engine has a different formula for assigning weight to the words in its index.

Search engines often use exact matching when processing keywords. However, there may be situations when exact matching is inadequate. For example, an exact search for ‘lion’ would miss those documents that contain ‘lions’. This is why most search engines now implement stemming as well. Stemming searches for a search term along with its variations. For the search term ‘think’, documents containing ‘think’, ‘thinks’, ‘thought’, and ‘thinking’ may also be returned.

Common words such as ‘where’, ‘how’ and ‘and’, as well as certain single digits and single letters, may be ignored because they add to the search engine’s workload without improving the results.

Now, let’s talk about the query interface. This is the portion of the search engine which is visible to users. Aside from a basic search box, most search engines also offer an advanced search interface, which is basically an online form which accepts more detailed information about the results you want.

Search engine users are often frustrated when they are not able to find the right answers, even though they are inundated with pages and pages of results. Users can help search engines do a better job by asking more precise questions through the advanced search interface.
In the next activity, you will try refining your searches using Google’s Advanced Search form and observe whether it succeeds in making your search results more focused and in filtering out irrelevant results.

**Activity 4.4**


The features available in Google’s Advanced Search are also available in basic search, but you must know how to type in the operators into the search box along with your keywords. A very common example of an operator is ‘OR’.

For example, if you’re looking for information on Bangkok or Shanghai vacation packages, you can type **vacation packages Bangkok or Shanghai** directly into the search box without going through the Advanced Search form.

The next two optional readings can provide you with more information on what other operators are available in Google:


Now that we understand how a search engine works, it’s time for us to put this knowledge to practical use in the next section.
Registering your site

Search engines are one of the most common ways to locate websites. Acquiring a good position within a search engine’s listings can lead to a dramatic increase in a website’s traffic. In this section, we will discuss the steps for getting your sites listed in a search engine (i.e. crawler-based) and a search directory (i.e. human-powered).

Search engine registration refers to the act of getting your website listed with search engines. Merely getting listed, however, does not mean that your website will be ranked highly for the search terms that you want to be associated with. Your website may never get listed at all. The only way to secure a guaranteed listing is by paying for it.

Search engine optimization refers to the ongoing process of refining and improving your search engine listings in order to achieve better rankings. This is something that website owners must do on a regular basis.

Although there are thousands of search engines, you should concentrate your efforts on the top ten search destinations online. These are more likely to send the most traffic to your site. The next reading lets you know the audience measurements for the most well-known search engines. It also describes the relationships between the search engines, since some of them actually use the services of a third party site to provide them with search results.

Reading 4.3


Note: Only read the section ‘Top 5 search destinations’.

Do the following reading to find out the percentage of searches handled by each search engine.

Reading 4.4


These two readings illustrate that there is more than one way to measure the reach and popularity of a search engine.
Self-test 4.4

Describe at least two situations which may result in over-counting when popularity is measured by audience reach (i.e. percentage of unique visitors who use a search engine).

Submitting to search engines

The next reading describes the steps for getting listed in crawler-based search engines.

Reading 4.5


Reading 4.5 stresses the importance of getting other sites to link to your site. It’s no longer enough to submit your website through the Add URL form in a search engine. Many search engines now consider the quality and quantity of the links pointing to your site when they determine how your pages should be ranked within their results.

Some search engines (such as Inktomi and Teoma) do not even provide the option to submit websites directly to them anymore. They rely completely on link analysis to determine if your site will get listed or not. Another option is to pay a fee in order to get listed on major directories and search engines. This increases the chances that your site will be picked up by other crawlers.

When building links to your site, you should concentrate on getting links from webpages whose content is closely related or similar to yours. For example, it’s a good idea for ABC Books to exchange links with websites that also deal with classic literary works.

Self-test 4.5

1 Describe the steps for submitting your website to a crawler-based search engine.

2 Visit the Add URL form for Google and Altavista. What information do you need to provide in order to get your website listed?

3 List some ways to increase the number of links pointing to your site.
4 What suggestion does Reading 4.5 give to newly established websites who wish to get listed despite the lack of links pointing to them?

It takes several weeks after you’ve submitted a site before it gets listed, if at all. Unless you’ve paid a fee in order to get listed, you cannot expect search engine companies to give you a timetable for when you’ll get listed, or why they have ignored your submission. Information providers must themselves check on their own whether their website has been indexed by the search engines they’ve submitted to.

**Activity 4.5**

The following reading shows you the best ways to confirm whether your webpages have been indexed by the major crawler-based search engines:


Use the URLs in the reading above to check whether ABC Books’ competitors are listed in the following search engines. You can also note how many pages from these competitors’ sites got listed.

**Table 4.1 Checking the listings of ABC Books’ competitors**

<table>
<thead>
<tr>
<th>ABC Books’ competitors</th>
<th>Search engines</th>
</tr>
</thead>
</table>

*Note: There is feedback on this activity at the back of the unit.*

In the end, it’s important to remember that search engines are not the only way that visitors will find your site. It’s estimated that search engines are only able to index 15% of all websites. There are other effective ways for promoting your website, such as email, advertising banners and link exchanges. You should never rely entirely on search engines to direct traffic to your site.

**Submitting to search directories**

The next reading describes the procedure for getting listed in crawler-based search engines.
When you submit your site to a directory, you can suggest an appropriate category for your site to be listed under. However, human editors will still evaluate your request and ultimately decide whether you will be listed under your desired category or somewhere else.

In the next activity, you will prepare the information that will be submitted to a search directory for the ABC Books website.

**Activity 4.6**

Here is a screen shot of Yahoo’s website submission form.

1 Prepare a suitable site title and description of ABC Books for submission to this search directory, keeping in mind the recommendations from Reading 4.6.
Aside from the subject category in figure 4.8, what other appropriate
categories could ABC Books be listed under? (Hint: Try locating
‘independent booksellers’ in Yahoo’s directory, since this category
may also be suitable for ABC Books.)

Note: There is feedback on this activity at the end of the unit.

Optimizing website rankings

Submitting your website to search engines is only the initial step in
achieving good listings. Information providers must monitor their
rankings regularly, because new submissions happen all the time and
may affect the rankings of other websites.

For example, your site may start out with a good ranking but later on get
pushed down, or even drop out entirely from the list. You should also
resubmit your site whenever you make substantial content or design
changes in order to get the crawler to refresh your website information in
its index.

There are many automated tools that can perform these tasks for you —
from generating keywords and META tags, to website registration and
search engine optimization. However, I’d like to focus on search engine
optimization as a manual process. After all, no one knows your site as
well as you do. You are more likely to do a better job than any software
when it comes to monitoring and optimizing your rankings.

Since most search engines now create a full-text index for the pages they
have visited, there will be an enormous number of documents returned
for a particular keyword search. For example, typing in a popular
keyword such as ‘flowers’ or ‘textbooks’ will return hundreds of pages
of results.

It’s important to understand how various search engines rank the returned
documents so that you can build webpages that perform well according
to their ranking criteria. Here are the general characteristics of
documents which are more likely to receive a better ranking:

• have more occurrences of the keywords;

• have the keyword in the title (which may result in a higher ranker
  than just having it in the body of the text);

• have the keyword in the description attribute of the META tag;

• have the keyword in their URL, e.g. when using the keyword ‘mp3’,
greater weight would be given to documents with the domain name
  http://www.mp3.com;

• have more of the keywords occurring close to each other within the
  same document (i.e. when searching on multiple keywords);
• have more webpages linking to them, also known as link popularity (the quality of the links is also evaluated, which means that a link from Yahoo counts more than a link from a less important page);

• are themselves listed in important sites like Yahoo;

• are often clicked by users when they are returned as search results, also known as click through popularity (click through rates are generally accepted as a measure of success in getting visitors to a site, but nowadays, higher traffic does not always translate into profitability); and

• belong to website owners who have paid a fee in order to receive a better ranking.

Google makes use of link structure information when ranking its search results. Using this scheme, pages that have more links pointing to them are considered more relevant and will therefore appear higher in the search results. The importance of the links themselves is also ranked, so results are ranked higher depending not just on quantity but on the importance of the pages that link to them. For example, a page might be given more importance if Yahoo points to it rather than if some unknown page points to it.

The location and frequency of keywords on a webpage may also affect its ranking. A search engine may analyse how often a keyword appears in relation to other words in a webpage. It may also check if the search keywords appear in certain areas, such as near the top of a webpage, within a heading or in the first few paragraphs of text (SearchEngineWatch.com).

**Self-test 4.6**

Discuss the benefits and problems of using the following strategies for ranking search results:

• link popularity;

• click through popularity; and

• paying for a higher rank.
Difficulties in indexing the Web

The information retrieval algorithms used on the Web today are based on well-known techniques that were originally used for smaller and more coherent collections of documents, such as news articles and library book catalogues.

Compared to a physical library, however, the Web is a lot more challenging to index and search. Here are the major differences between indexing the books in a library and indexing the documents on the Web:

- A librarian knows exactly how many books are in a library. No one knows the total number of webpages on the Web.
- A librarian knows exactly when a new book arrives or an old book is withdrawn from the collection. Search engines cannot confirm whether a website exists unless they have actually visited it.
- The books in the library do not change their contents independently. When a newer edition arrives, the library changes its indexes accordingly. However, the contents of a webpage can change anytime. Unless the author reports the changes to different search engines, or the search engines revisit the page, the index may contain out-of-date information about the webpage.
- All multimedia materials in a library, such as CD-ROMs, audio and microfiche, are indexed. However, search engines have great difficulty in handling multimedia materials. We will discuss this later in the unit when we look at the ‘invisible web’. Multimedia materials on the Web can only be indexed correctly if they are accompanied by adequate text descriptions.

The next table summarizes the effects of the factors listed above on the quality of library and Web indexes.

**Table 4.2  Comparison of library and Web indexes**

<table>
<thead>
<tr>
<th>Index of a library</th>
<th>Index of a Web</th>
</tr>
</thead>
<tbody>
<tr>
<td>The system is <em>closed</em>.</td>
<td>The system is <em>open</em>.</td>
</tr>
<tr>
<td>The index <em>always contains</em> information about all the library’s resources.</td>
<td>The index is not capable of containing information about all the webpages.</td>
</tr>
<tr>
<td>The index always contains <em>up-to-date information</em> about all the library’s resources.</td>
<td>It is difficult, if not impossible, to have up-to-date information about all the webpages in the index.</td>
</tr>
</tbody>
</table>

Search engines have an enormous influence in routing traffic to websites, and as such, there are Web developers and even entire companies who deliberately manipulate search engines so that they can maximize their
traffic, even under false pretenses. This is another problem that does not exist in traditional, closed information retrieval systems.

**Nature of Web content**

Changes to information retrieval techniques are needed in order to make them work within the context of the Web. Search engines must take into account that Web content may:

- exist in different file types or formats — text, HTML, Portable Document Format (PDF), images, audio, video;
- contain hyperlinks to the same document or to external documents;
- be in different languages — English, Japanese, Simplified Chinese, Traditional Chinese, etc.;
- use different vocabularies — phone numbers, product numbers, stockkeeping units (SKUs);
- be static (pre-built) or dynamically generated (constructed in real time);
- be spread over geographically separate and distributed computers; and
- have external metadata, or information which can be inferred about the document but is not stored within it (e.g. page update frequency, reputation of the source, popularity or usage, and citations).

Another major characteristic of the Web is that no one knows exactly how much content is available on it. Studies suggest that a major portion of the Web is not even indexed by search engines or directories. You’ll find out more about this in the next section.

**The deep Web**

Web content can be divided into two areas. The ‘surface’ Web consists of static, publicly available webpages which can be crawled and indexed by search engines. The ‘deep’ Web consists of specialized, Web-accessible databases and dynamic websites which are not readily found through search engines. Many average Web users do not even know that the deep Web exists, even though it is estimated that it contains 400 to 550 times more information than the surface Web. The total quality of content on the deep Web is also estimated to be 1,000 to 2,000 times greater than that of the surface Web (BrightPlanet 2001).

The deep Web is also called the ‘invisible’ Web, because this portion of the Web is invisible to search engines. The next reading describes the types of content that form the deep or invisible Web. It also gives some good advice on how users can expand their search to include both the surface and the deep Web.
OUHK’s Electronic Library is a very good example of invisible Web content. Although the library is accessible on the Web, its contents are only meant for the use of OUHK students, tutors, academics and staff and cannot be indexed by search engines. Many public libraries also offer Web access to online databases containing full-text journals and news articles, including our very own Hong Kong Public Library. All you need to access these Web-enabled databases is a valid library card number.

Activity 4.7

Visit the following deep websites and compare the quality of the information on these sites with what you can get on the surface Web:

1. The Internet Movie Database — comprehensive collection of movie information, including Asian movies:
   http://www.imdb.com

2. Hong Kong Public Library (Electronic Resources) — provides access to full-text journals, images and citations covering various disciplines:
   http://www.hkpl.gov.hk/01resources/1_3electronic_internet.htm

3. IBM’s Resource for Developers — provides technical information that can be used by developers when building and deploying information systems:

4. Alibaba.com — directory of suppliers, manufacturers, importers and exporters:
   http://www.alibaba.com

This concludes our section on Web searching and indexing. The remaining sections of this unit will concentrate on how information providers can provide search services on their own websites. Take the following self-test to assess how well you understand the deep Web and how users can take advantage of the information available within it.
$\textbf{Self-test 4.7}$

List four examples of information that belongs to the invisible Web, and explain why this information is inaccessible to search engines.
Providing a search facility

No matter how well-designed your website navigation is, some users may still fail to find the information they need. If you do not offer an alternative way of accessing your site, most users will simply give up and move on after a few failed attempts.

Usability studies show that more than half of all users are search-dominant (Nielsen 1997). Search-dominant users will usually go straight for the search button when they enter a website. They are very task-focused and want to find specific information as fast as possible without spending time to get used to your website navigation and structure.

A local search tool can provide a powerful and familiar alternative for accessing a website. Visitors can just type their search terms, press the search button in a form, and get a list of all the documents on your site that match their search terms. Local search works in a similar way to Web search engines, except that the index is confined to a single website.

As a rule of thumb, sites with more than about 200 pages should offer a search facility (Nielsen 1997). Even the best website with a good menu system can present a faster, friendlier interface to its information by offering a supplementary search tool. The good news is that more and more good search tools are becoming available.

In this section, you’ll learn more about:

• information to help you choose from the many search tools available;

• installing a search engine on your site using local and remote methods; and

• server-side programming examples for implementing local search.

Local or remote

The good news is that you rarely have to create your own search engine. There are many search tools available for almost any platform and Web server you can imagine. They range from free to very expensive, from user-friendly, graphical interfaces to compile-it-yourself. No matter which option you choose, though, you should know that there are two ways of providing a search service for a website:

1 Local — the search engine runs locally on your Web server and conducts searches against a Web index stored on your local machine.

2 Remote — the search engine runs on a remote server and conducts searches against a Web index stored on a remote machine. Remote search services are also known as hosted search services, since the index and software are hosted on a third-party site.
Note that the remote option (e.g. option 2) is not the same as submitting local pages to search engines such as Google or AltaVista. When you submit your pages to these Web search engines, the index entries are added to their global index made up of search terms from pages taken all over the Web. The remote option we are talking about will confine the search only to index entries built from the pages on your website.

Activity 4.8

The following URL discusses whether remote or local search should be used. Read pages 2 and 3 of the article ‘Adding search to your site: remote vs. local search’ at

http://hotwired.lycos.com/webmonkey/00/09/index2a_page2.html

After you have finished reading, answer the following questions:

1 State whether the following conditions apply to local or remote search services:
   • It’s better to use this if the data is not open to the public.
   • It saves effort in maintaining the search engine.
   • You can have more control over the indexing process.
   • You may have to pay on a regular basis.
   • You save money in buying software.

2 What do you have to do before registering for a remote server to index your pages?

Note: There is feedback on this activity at the back of the unit.

Parts of a local search engine

Next, let’s look at the different components of a search tool. Since you are basically integrating a search application provided by an external organization or individual, you should take some time to understand how the whole thing works before you install it on your server or integrate it with your webpages. The search engine concepts that you’ve previously learned are also applicable to local search tools, as you’ll see from the following figure.
Here's a more detailed description of these components:

1. Search engine — the program (CGI, server module or separate server) that accepts the request from the form or URL, searches the index, and returns the results page to the server.

2. Search index file — created by the search indexer program, this file stores the data from your site in a special index or database, designed for very quick access. Depending on the indexing algorithm and size of your site, this file can become very large. It must be updated often, or it will become unsynchronized with the pages and provide obsolete results.

3. Search forms — the HTML interface to the site search tool, provided for visitors to enter their search terms and specify their preferences for the search. Some tools provide pre-built forms.

4. Search results listing — an HTML page listing the pages which contain text matching the search term(s). These are sorted according to relevance, usually based on the number of times the search terms appear, and whether they're in a title or header. Most results listings include the title of the page and a summary (the metadata data, the first few lines of the page, or the most important text). Some also include the date modified, file size, and URL. The format of this is often defined by the site search tool, but may be modified in some ways.

Whether you decide to implement a local or remote search service, these components are generally the same.
Now that we know how a local search service works and what its different components are, we will implement a remote indexing and search service for ABC Books in the following activity.

Activity 4.9

1. From the course website, download the version of ABC Books’ website (abcbooks_v2.zip) which shall be used in this activity. This version contains 43 static HTML pages (with one page for each author and for each book title), as well as images and style sheets. There is a subdirectory called authors containing the Author pages, and another subdirectory called books containing the Book pages.

2. Upload the website to your Web server at OUHK. The website needs to be online for the remote search service to access and index its pages.

3. FreeFind is the remote search service that we will use. You must first register for an account at http://www.freefind.com/

Our goal is to use FreeFind to create a remote search index for ABC Books, and then to try some searches on it.

4. Here are the general steps for implementing this:

   • After registering, you will receive an email containing a password which you can use to administer your account. Log in to your account at FreeFind and request that your website be indexed.

   • Another email will be sent to let you know that your site has already been indexed. Log in to your account at Freefind and copy the HTML that is needed to include a search box on your website. Freefind offers a number of styles for you to choose from. Figure 4.11 shows one of the available options.

   ![Search this site powered by FreeFind](Figure 4.11 One option at FreeFind)

   • Once you’ve included a search box on your site, enter some search terms and see if it retrieves the correct pages! (Note: FreeFind displays the results in a new browser window, using its own page template. Since we are using the free service, there will be advertisements shown along the top of the results page. The
advertisements can only be removed by upgrading to a paid account.)

- You can further customize the search results page by adding your own logo, changing the background and text colour, and specifying which fields (e.g. title, description and URL) will be shown. This can be done when you log in to your account as well.

5 For detailed instructions on how these steps should be done, you can refer to the following page on the FreeFind site:

http://www.freefind.com/library/tut/pagesearch/

The important sections to read are: ‘Setup overview’, ‘Indexing your site’, ‘Adding your panel to your site’ and ‘Customizing your search results’.

Note: You can view the model answer in action on the course website.

---

**Search gateway programs**

Next, we’ll implement a local search service for ABC Books using a search gateway program. This is basically a server-side application which allows a user to search all the files on a Web server for specific information. The program runs on the Web server, and the Web index is also stored on the local machine.

In this section, we will use SWISH-E (Simple Web Indexing System for Humans — Enhanced) to provide a local search service on ABC Books’ website.

---

**Activity 4.10**

1 We will continue using the ABC Books’ webpages which you downloaded in Activity 4.9. You will install and implement the local search service on your own local Web server this time.

2 SWISH-E is the local search software that we will use. You can download the UNIX version from:

   http://www.swish-e.org/download

3 Next, you should unzip the downloaded file and install it from the directory which is created for you. Here are the instructions to follow:

   http://www.swish-e.org/current/docs/INSTALL.html
   #Building_Swish_e
4 After installing SWISH-E, create a configuration file in your HTML folder which contains the instructions for indexing your site. You can write this file in any text editor, or you may download it from the course website (the file is called swish-e.conf). The lines starting with # are comments which explain the indexing directions:

```
# Example Swish-e Configuration file
# Define *what* to index
# IndexDir can point to a directories and/or a files
# Here it's pointing to two directories - authors and books.
# Swish-e will also recurse into sub-directories.

IndexDir ./authors
IndexDir ./books

# Only index the .html files
IndexOnly .html

# Show basic info while indexing
IndexReport 1
# Specify words to ignore, called stopwords.
IgnoreWords www http a an the of and or
```

5 Now you can build the search index based on the instructions in this configuration file. Enter this in the command line:

```
$swish-e -c swish-e.conf
```

If all goes well, SWISH-E will display a series of messages telling you how many files were indexed successfully:

Indexing Data Source: "File-System"
Indexing "./.authors"
Indexing "./.books"
Removing very common words...
no words removed.
Writing main index...
Sorting words ...
Sorting 42 words alphabetically
Writing header ...
Writing index entries ...
Writing word text: Complete
Writing word hash: Complete
Writing word data: Complete
42 unique words indexed.
4 properties sorted.
6 files indexed. 727 total bytes. 50 total words.
Elapsed time: 00:00:00 CPU time: 00:00:00
Indexing done!

6 SWISH-E also creates an index file called swish-e.index in the current directory. You can try searching the index directly from the command line. The following example will search the index for the keyword ‘Sun Tzu’:

```
swish-e –w "Sun Tzu"
```
Now that you’ve verified the search engine works, we need to build a server-side program which accepts keywords from the user and sends them to SWISH-E for processing. You can download a pre-written PHP search script from the course website (search_swish.php). Edit the section within the code titled ‘User-defined configuration variables’ and place the values specific to your own installation.

```bash
# Absolute path and command to execute the SWISH searcher
$swish = "/usr/local/bin/swish-e";
# Name of configuration file
$swishconf = "swish-e.conf";
```

Now link to `search_swish.php` from the menu label called **Search** from your homepage. Test that the search engine works from your Web browser.

*Note: You can view the model answer in action on the course website.*

It’s possible to create several indexes which cover different parts of a website. ABC Books could have built two indexes, one for the Authors directory and another one for the Books directory. When specifying the search terms, users can also indicate which section of the site they want to search. The SWISH-E program could be instructed to use different configuration files depending on the section being searched.

**Databases**

The search tools from the two previous activities (4.9 and 4.10) are applicable to websites that contain only static, pre-built pages. However, there are now many websites which draw their content from databases in real-time. In this section, we will discuss how full-text indexing can be implemented in database-generated pages.

Databases will be covered more extensively in *Unit 5*, but here’s a brief overview. You can think of databases as ‘a collection of information organized in such a way that a computer program can quickly select desired pieces of data’ (Webopedia.com). Databases can be accessed via Structured Query Language (SQL), which is basically a standard way of issuing queries against a database without any need to know what its underlying structure is.

Right now, ABC Books’ website consists of static pages only. This means that whenever changes are made to their book catalogue (e.g. title, author or price updates), the same change must also be made to all webpages where that information is found. This method may be feasible now that their product catalogue only consists of 25 titles, but what if their collection expands beyond a hundred titles? Creating and updating all these pages by hand will definitely be a time-consuming and potentially error-prone exercise.
One alternative is to store ABC Books’ catalogue in a database. Here’s a basic list of fields which could make up a record in their book catalogue table:

1. ISBN — unique identifier for the book;
2. page URL — the URL of the page;
3. page title — the title of the webpage (e.g. ‘The Art of War’ by ‘Sun Tzu’, or ‘The Life of Jane Austen’); and
4. page content — the book summary (if this page is for a book) or the author’s biography (if this page is for an author).

Databases can allow full-text searching on record fields. We can write a server-side application which uses Structured Query Language (SQL) to do a full-text search on the page title and page body fields above. If a match is found, then we will display a link to the corresponding URL from the list of search results.

The next reading describes the steps for doing this using the mySQL relational database.

---

**Reading 4.8**

‘Using mySQL full-text searching’,

*Note:* The section on ‘Advanced boolean searching’ is optional.

---

Now that you’ve seen how full-text searching can be done against a relational database using a PHP script, let’s try this out for ourselves in the next activity. You will need to have mySQL installed on your local Web server before you proceed.

---

**Activity 4.11**

1. You should download mt834_act4-11.zip from the course website, which contains the following files:
   - A mySQL script called init_data.sql which will create the ABCBooks database, create the Book_Catalog table within it, and then load records into this new table.
   - A PHP script called search_mysql.php which accepts a search keyword via an HTML textbox and performs a full-text search of this keyword against the Page_Title and Page_Content fields of the Book_Catalog table. This script is very similar to the script described in Reading 4.7.
2 Execute the mySQL script `init_data.sql` to create and initialize the database:

```bash
$mysql -i init_data.sql
```

3 Log in to mySQL and verify that the database and table were created successfully. We will use the `select` SQL command to view the records loaded into the table:

```bash
$mysql ABCBooks
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 54 to server version: 3.23.58
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql> select * from Book_Catalog
```

4 Now link to `search_mysql.php` from the menu label called **Search** from your homepage.

*Note:* A username and password is not required in order to access the ABCBooks database. But if you change the access levels for this database later on, you should remember to change the username and password within `search_mysql.php`.

5 Test your full-text search and fix any problems you find. You can view the model answer in action on the course website.

The database search application created in Activity 4.11 can be further improved by allowing users to specify whether they are searching only on the title or on the page body. This is done by modifying the search criteria in the SQL `select` statement. You will get more hands-on practice on database searching in the next unit.

You’ve now seen a number of ways of providing a search facility on your websites. First, we used a remote indexing and search tool from a third party provider (FreeFind). Second, we installed our own search engine (SWISH-E) and implemented a server-side script to query the index created by this search engine. Third, we used the full-text searching capabilities of a database in order to search its contents. This was also done through a server-side script which issued SQL commands against the database.
Summary

The World Wide Web has brought about dramatic changes in the way people access information. Information resources that were formerly locked away in libraries, CD-ROMs, print journals and proprietary databases can now be searched conveniently and efficiently by anyone with a desktop computer, a Web browser and an Internet connection.

In this unit, you learned that creating an index for the Web is much more difficult than creating an index for a library. The main problem in indexing the Web is that its contents are open-ended. No one knows exactly how many webpages there are on the Web or exactly when a page will cease to exist.

You were introduced to the components of search engines, namely the robot, the indexer, the query system and the retrieval system. We then looked at the different characteristics of robots as well as the mechanisms used by website owners to communicate with robots.

You also studied the so-called ‘deep Web’, the portion of the Web that is not covered by search engines. Examples are audio, movies, framed pages, password-protected pages, dynamic websites and online databases. We also discussed ways to make our searches more effective by including content from both the surface Web and the deep Web.

Finally, we discussed how to implement local search facilities on our own websites. You learned that there are two ways of doing this: (1) installing and running your own local search engine; or (2) having your pages indexed by a remote server. You were presented with the pros and cons of the two methods. Finally, you installed a search engine on your ABC Books website using both options.
### Feedback to activities

#### Activity 4.1

<table>
<thead>
<tr>
<th>Topic</th>
<th>Google</th>
<th>Yahoo</th>
</tr>
</thead>
<tbody>
<tr>
<td>hiking and trekking vacations</td>
<td>I typed in <strong>hiking trekking vacations</strong> in their search engine (<a href="http://www.google.com">http://www.google.com</a>).</td>
<td>I clicked on <strong>Recreation</strong> from the directory (<a href="http://dir.yahoo.com">http://dir.yahoo.com</a>).</td>
</tr>
<tr>
<td></td>
<td>More than 35 pages of results were returned. The few links I clicked on were relevant to the search, but it was a bit difficult figuring out which were the best links to select based on the descriptions shown after each link.</td>
<td>I clicked on <strong>Outdoors</strong> from the <strong>Recreation</strong> subdirectory.</td>
</tr>
<tr>
<td></td>
<td>I clicked on <strong>Recreation</strong> from the <strong>Recreation</strong> subdirectory.</td>
<td>I clicked on <strong>Hiking</strong>, which returned a concise, one page list of hiking-related websites that could be scanned easily by the human eye. The quality of the listings was very good, and the brief description for each listing was very informative.</td>
</tr>
</tbody>
</table>

#### Activity 4.2

1. The robots.txt file should contain the following:

   ```
   User-agent: *
   Disallow: /authors
   Disallow: /books
   ```

2. The following META tag should be included in authors.html to restrict the spider from indexing any pages that it links to:

   ```html
   <META name="robots" content="nofollow">
   ```

#### Activity 4.3

A breadth-first search will gather these five pages: `index.html` → `books.html` → `authors.html` → `news.html` → `tarzan.html`.

A depth-first search will gather these five pages: `index.html` → `books.html` → `tarzan.html` → `kingdoms.html` → `mansions.html`.

A breadth-first search will result in all the author names and book titles getting indexed, because the spider can extract them from the Authors and Books pages, respectively. A depth-first search will index the book titles only, and so users who search on author names via search engines may not find ABC Books’ website.
Activity 4.5

Here are the pages listed in each search engine:

For hkupress.org, Google has 46 results while Altavista has 17.

For paddyfields.com, Google has one result while Altavista has none.

Activity 4.6

1 Here’s a suggested title and description for ABC Books:
   Site title: ABC Books — Hong Kong’s Literary Book Store
   Description: Hong Kong bookseller specializing in classic literary works in English and Chinese

2 Here is another possible category where ABC Books may be able to reach international buyers:

   Directory → Business and Economy → Shopping and Services → Books → Bookstores

Activity 4.8

1 • It’s better to use this if the data is not open to the public — local.
   • It saves effort in maintaining the search engine — remote.
   • You can have more control over the indexing process — local.
   • You may have to pay on a regular basis — remote.
   • You save money in buying software — remote.

2 Steps to be done before registering with a remote server include:
   • checking that all pages can be accessed from the homepage directly or indirectly;
   • checking the contents of robots.txt to see if robots are allowed to visit the pages; and
   • if the pages contain frames, including <noframes> tags to ensure that they can still be indexed.
Suggested answers to self-tests

Self-test 4.1

It is more suitable to use a search engine when you know which keywords to use and you want to search for them in a large, comprehensive database.

Example: searching for materials about thermonuclear reaction.

It is more suitable to use a search directory when you have an idea of which categories the information may be listed under, but you do not have specific keywords. Search directories are also suitable if you want to search through a smaller database that has entries that are handpicked.

Example: searching for any French restaurant in the Sai Kung area.

Self-test 4.2

1 Here are just a few of the guidelines that responsible robot designers should abide by:
   • set a limit on how many pages should be retrieved from a particular site;
   • make a list of places that should not be visited before starting the traversal;
   • identify the robot to the website that is being visited;
   • remember the places that have already been visited so the same page is not retrieved multiple times;
   • scan the URLs and verify that the spider can handle the content types on these pages before visiting them; and
   • test your robot on a number of local servers before launching it onto the Web.

2 nofollow tells a robot not to crawl to pages that are pointed to by the current page. noindex tells a robot not to index the current page.

3 Only the Web maintainer can use robot.txt as it is placed in the root of the document directory. Thus, if you are a common user, your only option is to use a META tag. In addition, all credible search engines honour robot.txt while some of them ignore the robot META tag.

4 It is entirely up to robot if it will honor any restrictions communicated to it by the information provider through the use of robots.txt and the robots META tag. These two methods do not actually prevent the robot from accessing restricted files or directories. If the Web maintainer really wants a robot to stay away from certain files or directories, then he/she should consider other options such as password protection.
**Self-test 4.3**

- Size of the index — How many documents are included in the index, and what percentage of the total Web is it able to search?
- Freshness or ‘up-to-dateness’ — How often is the index refreshed?
- Completeness of text — Is full-text indexing used, or are keywords extracted from certain areas of the page only?
- Types of documents offered — Are only HTML documents included, or are other file formats such as PDF (Portable Document Format), DOC (Microsoft Word) and images (GIF and JPEG) also searchable?

**Self-test 4.4**

Some situations may result in over- or under-counting when popularity is measured by audience reach.

A ‘unique’ visitor is actually a single computer used to access a search engine, which leads to the following situations:

- If the same person uses a search engine from two different computers (e.g. home and work), they are counted twice.
- If several people use the same computer to access a search engine, they are counted once.

**Self-test 4.5**

1. Steps for submitting your website to a crawler-based search engine:
   - choose keywords which are relevant to your page content;
   - add META tags to your pages containing the title, keywords and description;
   - submit your site to your chosen search engines;
   - optimize your listings by building links to your site and refining your webpage content; and
   - verify and maintain your listings periodically.

2. As of the time of writing, here is the necessary information in order to get listed:
   - Google — URL of homepage plus some comments about your site; and
   - Altavista — up to five URLs plus email address.

3. A good way to build links to your site is to find other websites which are complementary to yours. You can look for these websites by entering your target keywords in major search engines. You can approach the owners of websites which appear in the top results and establish a linking partnership with them. You can also establish
links with specialized portals, online merchants, and industry and business websites that are related to your website’s topic.

4 Newly established sites can consider paying a fee in order to get listed on major directories such as Yahoo and major search engines such as Google. This increases the possibility that they will get picked up by other crawlers. They should then invest time and effort in building reciprocal links with other related sites. In time, they may remain listed on search engines without paying a fee as long as they have enough links pointing to them.

**Self-test 4.6**

Here are some benefits and problems associated with various strategies for ranking search results:

- Link popularity assumes that a link from website A to website B is a vote or recommendation by A for B. This method gives a lot of importance to the opinions of website owners, which may not always be equal to the relevancy of the link. Some sites may even cooperate by including links to each other on their pages. This would defeat the purpose of link popularity.

- The benefit of click-through popularity is that it considers how many search engine users actually visited the page when it was displayed within the search results. This may result in less opportunity for cheating and collusion among website owners compared with the link popularity strategy. However, just because a user clicks on a result doesn’t mean that they will spend a lot of time on the site or that they will contribute towards a site’s profitability.

- There is still some discussion on the Internet regarding whether it is ethical to pay for a higher rank. The argument for this method is that if a non-relevant site pays for a higher rank, people will probably be disappointed by the site’s content and the Web maintainer will not gain any benefit from this higher rank. Smaller websites whose owners cannot afford the fee may find it difficult to get listed at all.

**Self-test 4.7**

Some examples of information on the ‘invisible’ Web:

- Dynamically generated pages — These are generated by server-side scripts whose URLs may include the question mark (?). Spiders may be instructed to avoid visiting such URLs for fear of getting stuck in an infinite loop or running faulty scripts.

- Password protected pages — Many websites contain sections which require users to log in with a username and password. These pages cannot be accessed by a spider.

- Searchable databases — Pages generated from a database require search parameters to be passed to them. Unless there are static pages
which link to the database content, database content cannot be listed in search engines.

- Non-HTML formats — Search engines are primarily designed to index HTML pages, so pages which have large portions of non-HTML content (e.g. large graphics, animation or multimedia) may be difficult to index.

- Framed pages — Not all search engines can handle framed pages, while some search engines index the contents of each frame within a page separately. This may result in a framed page being returned out of the context of its surrounding frameset.
References


