Lecture 39: Search engines and web crawler: Part II

On completion, the student will be able to:

1. Explain the design goals of WebCrawler.
2. Identify the different components of WebCrawler and explain how they function.
3. Understand how WebCrawler discovers new documents to be indexed.
4. Explain the differences between the indexing mode and real-time search mode of WebCrawler.
WebCrawler

Introduction

• WebCrawler is a tool that helps in solving the resource discovery problem in the world wide web.

• Some features:
  - It navigates the web automatically on demand.
    - Because the web is changing continuously.
  - To perform the indexing, it accesses the web one document at a time, sometimes taking decisions based on information available locally.
What is WebCrawler basically?

- A single piece of software, with two different functions:
  - Building indexes of web pages.
  - Navigate the web automatically on demand.
- Key design goals:
  - Content-based indexing.
  - Breadth first search to create a broad index.
  - Crawler behavior to include as many web servers as possible.

Design of the WebCrawler

- World wide web has a dynamic structure, and is decentralized.
  - Any individual can add contents to the web (web servers, web pages, hyperlinks).
- Design implications ....
  - Discovering new documents and keeping the database updated forms a very important component of the design.
  - In the form of uniform resource locators (URLs).
    - Can refer to any retrievable network resource.
• The WebCrawler performs the following actions after “retrieving” a document.
  ➢ It flags the document as being “retrieved”.
  ➢ Looks for any outbound hyperlinks in the document.
  ➢ Indexes the content of the document.
• All the above actions requires some information to be stored in a database.

Components in WebCrawler
• The various components in WebCrawler.

  ➢ Search engine
  ▪ Responsible for deciding which new documents to explore, and for initiating the process of their retrieval.

  ➢ Database
  ▪ Used to store the document metadata, full-text index, and the hyperlinks between documents.

  ➢ Agents
  ▪ Responsible for retrieving the documents from the web under the control of search engine.

  ➢ Query server
  ▪ Responsible for handling the query processing service.

  ➢ libWWW
  ▪ This is the CERN WWW library, used by agents to access several different kinds of contents using different protocols.
Search Engine

• Basic requirement:
  ➢ WebCrawler discovers new documents.
    ▪ Start with a known set of documents.
    ▪ Examine the hyperlinks from them, that point to other documents.
    ▪ Repeat the whole process recursively.
  ➢ Alternate way of looking at the problem.
    ▪ Web is a huge directed graph, with documents as vertices and hyperlinks as edges.
    ▪ Need to explore the graph using a suitable graph traversal algorithm.

Web Viewed as a Graph
• Some points about search engine:
  ➢ The search engine is responsible for:
    ▪ Which documents to visit
    ▪ Types of documents to visit
  ➢ No point in retrieving files that the WebCrawler cannot index.
    ▪ Images, audio clip, postscript file, etc.
    ▪ Even if they are retrieved, they are ignored during indexing.

• WebCrawler basically carries out two kinds of searching:
  ➢ In the indexing mode
  ➢ In real-time search mode
WebCrawler: Indexing Mode

• Basic motivation:
  ➢ Try and build an index of as much of the web as possible.
  ➢ Some heuristics used:
    ▪ Which documents to select if the space for storing indices is limited?
    ▪ A reasonable approach is to ensure that documents come from as many different servers as possible.
  ➢ WebCrawler uses a modified breadth first approach in order to ensure that every server has at least one document that has been indexed.

• The breadth first strategy:
  ➢ Every time a document on a new server is found, the server is marked as “visited”.
  ➢ All the servers being pointed to by hyperlinks are put on an “open” list.
  ➢ The process is repeated.
    ▪ Look for a server in the “open” list which has not yet been “visited”.
WebCrawler: Real-time Search

- Basic motivation:
  - Given a user’s query, try to find documents that most closely matches.
  - A different search algorithm is used here by WebCrawler.
  - Intuitive reasoning:
    - If we follow the links from a document that is similar to what the user is looking for, they will most likely lead to relevant documents.

- Steps followed:
  - Run the user’s query against the WebCrawler index, to get the initial list of “similar” documents.
  - Select the most relevant documents from the list.
    - Follow any unexplored links from these documents.
    - As new documents are retrieved, they are also added to the index, and the query rerun.
The results are sorted by relevance.
The documents highly relevant are possible candidates for further exploration.

• Possible problem with this approach:
  - Links are followed blindly, with no surety that the path being followed is a relevant one.
  - How to tackle this?
    - Use sophisticated pattern matching techniques to find “closest match” based on anchor text.
    - Ensure that you do not miss the context during navigation.
About Agents

- Responsible for retrieving the documents from the web under the control of search engine.
- Interface to an agent:
  - Input:
    - a command “retrieve URL”
  - Output:
    - an object containing the requested document, or an explanation why it could not be retrieved.

About the Database

- The database that is maintained by WebCrawler consists of two pieces:
  - A full-text index
  - Representation of the web as a graph
- The database is stored on disk, and is updated dynamically.
• It implements a search service.
  ➢ Users enter keywords as their query.
  ➢ Page titles and URLs of documents containing some or all of the keywords are retrieved from the index.
  ➢ Results sorted by relevance and presented to the user.
  ➢ How to define relevance?
    ▪ Over all the words in the query, find the sum of the product of the word’s weight in the document and its weight in the query, all divided by the number of words in the query.

• Query processing time:
  ➢ Can process 8 to 10 queries per second.
  ➢ Time includes parsing of the query, calling the database server, perform the query match, retrieve the results, format the search results as a HTML page and send them back to the HTTP server.
• How many agents?
  ➢ During searching, waiting time for accessing the server and network resources is the main bottleneck.
  ➢ Agents are typically run as multiple concurrent processes or threads (10 to 15, say).

End of Lecture 39
1. What is the basic purpose of search engine?

It is a software program that helps in locating information stored on a computer system, typically on the world wide web.
Quiz Solutions on Lecture 38

2. What is the difference between a crawler-based and a human-powered search engine?

Crawler-based search engines explore the web and create the listings automatically. Here the author of the web page has to submit a short description for being included in the listing.

3. In a crawler-based search engine, what is a crawler?

A crawler or spider visits a web page, retrieves it, and follows the hyperlinks to other pages within the site. It visits the site regularly (say, once every month) and look for changes.

4. In a crawler-based search engine, what is the index?

It is a huge book that contains a copy of every web page that the crawler finds.
5. What was the purpose of the software tool Archie?

It downloads directory listings of all files on anonymous FTP sites, and creates searchable database.

6. Do the search engines crawl through the entire web? Explain.

Crawl a small fraction of the web at moderate frequency. Crawl dynamic web sites at higher frequency.

7. How can a search engine automatically generate the keywords from a document?

It tries to pull out and index words that appears to be significant (title of page, words repeated many times, words appearing towards the beginning of page, etc.).
8. How is the page relevance typically measured?

By search term frequency and their position in the document, how many links from other documents, etc.

9. What is a meta tag? How is it specified in a HTML file?

They specify some information, including keywords in the document. They are specified using the “META” tag.

10. How can you prevent a crawler from using the keywords meta tag information for indexing?

By including the following meta tag:

```html
<META name=“ROBOTS” content=“NOINDEX”>
```

11. Why do some very popular search engines do not index using keywords meta tags?

Because some web authors use false and misleading keywords in the meta tag, to increase their page hits.
Quiz Questions on Lecture 39

1. What are the basic functions of WebCrawler?
2. What are the key design goals of WebCrawler?
3. How is the index generation problem mapped into a graph problem?
4. Does WebCrawler retrieves every file being pointed to for the purpose of indexing? Explain.
5. How does WebCrawler function in the indexing mode?
6. What is real-time search?