Searching for Information in Databases

Finding Information in Databases

- Different ways this can be done
- Hierarchical
  - Narrow down information by selecting categories
- Search Queries
  - Input search query that describes what you want to find
Hierarchical Search

- Browsing for a book in Amazon
- Want a Business Ethics textbook
- Our question: How is Amazon’s browsing organized?

Choose Books

34,348,676 possibilities
Choose **Textbooks**

# possibilities probably > 1,000,000

Choose **Business & Finance**

319,768 possibilities
Choose Business Ethics

1,877 possibilities

Find the Book
Hierarchy Example

All Departments > Books > Textbooks > Business & Finance > Business Ethics

? > 34,348,676 > ? > 319,768 > 1,877

Hierarchies

- Group in categories
  - Then subcategories
    - Then another subcategory
      - And so forth
  - Often shown as a tree (like a family tree)
Overlapping Categories

- In an actual database, such as Amazon, categories in a hierarchy often overlap.
- More like a network than a tree.

Search Queries

- Most common way of searching.
- Enter query describing what want to find.
  - Query: combination of words and phrases.
- Google searching is an example.
  - Google is an example of a Search Engine.
Search Engines

- Three parts
  - Web crawler
    - Searches Internet for web pages
  - Database
    - Web pages and keywords from these pages
  - Query processor
    - Searches the database to match a query

Web Crawlers: Searching the WWW

- Automatically browse the WWW in a methodical manner
  - Controlled by crawler servers
  - Copy all visited web pages
  - Identify URLs not yet visited
  - Prioritize web pages for future visits
- Retrieve copies of visited web pages
**Database: Storing Search Results**

- Contents of each page analyzed
  - Keywords extracted from titles, headings, links, etc.
- Stored in Index Database
  - Keywords and URLs from pages
  - All or part of the page (the cache)

**Query Processor: Searching the Database**

- User enters a query
- Query Processor
  - Compares query with keywords in Index Database
  - Provides a listing of web pages that best match the query
    - Search results usually ranked by relevancy

- Note: When you are searching with Google, you are searching Google’s Index Database, not the actual web.
Google: Ranking Pages

- Google uses PageRank to determine relevancy
  - Partially based on link popularity and quality
  - Premise: a good page will frequently be linked by other good pages

Search Engines Aren’t Perfect

- Their page ranking is not perfect
  - Ranking is based on popularity, not your particular interest
  - Some sites try to use tricks to increase their page ranking
- They don’t find information from all web sites
  - The web is growing too fast to be indexed completely
  - Many sites cannot be indexed by search engines (the “deep” or “invisible” web)
The Deep Web

- Why can’t all web pages be indexed by conventional search engines?
  - Several sites, such as commercial or private databases, limit access.
  - Some databases dynamically generate web pages
    - Banner Web is an example
    - No static web pages
    - When a user submits a query, a web page is generated to match the data that has been retrieved
    - Nothing permanent for webcrawler to find.
  - Some pages use unusual coding

Search Queries

- Online searches generally begin with a query

  The effectiveness of the search depends on the quality of the query
Simple Searches

- People generally do simple searches
  - Just one or two words

Problems with Simple Searches

- May not be broad enough
  - A search for “kidney disease” may miss references to diseases of the renal system
  - It also might miss references to “kidney diseases”
- May be too broad
  - A Google search for “digital cameras” returned 20,500,000 results
  - This is way too many to be useful
- How can we do better?
Effective Queries

- Effective queries generally use two features
- Truncation
  - Search for the stem of a word
- Boolean operators
  - Combine search terms to narrow or broaden a search

Truncating Words

- It is useful to search for several words by using a stem in a query
  - “gen*” to find gene, genes, genetics, genetically, etc.
- Google provides this automatically
  - Doesn’t use an asterisk *
  - Not very useful
  - Too many matches for a stem
Truncating Words

- In most databases you can truncate words with truncation symbols.
- Frequently * is used to substitute for any number of letters.
  - A search for “gen*” will find gene, genes, genetics, genetically, etc.
  - You may need to consult the Help or Search Tips section of a database to find out what truncation symbols it uses.

Boolean Operators

- **AND**
  - Search for Term#1 AND Term #2
  - Find results only if both terms are present
- **OR**
  - Search for Term #1 OR Term #2
  - Find results if either term is present
- **NOT**
  - Finds results if a term in not present
  - Not as useful
The OR Operator

- Searching for keyword1 OR keyword2
  - Finds results that contain one or both of the words.

- Example
  - Example 1: A lemon is sour.
  - Example 2: Apple pie is good.
  - Example 3: Lemon pie is even better than apple.

- What will search for lemon OR apple find?

The AND Operator

- Searching for keyword1 AND keyword2
  - Finds only results that contain both of the words.

- Examples
  - Example 1: A lemon is sour.
  - Example 2: Apple pie is good.
  - Example 3: Lemon pie is even better than apple.

- What will search for lemon AND apple find?
The NOT Operator

- Searching for keyword1 NOT keyword2
  - Finds results that contain keyword 1 but do not contain keyword2.

- Examples
  - Example 1: A lemon is sour.
  - Example 2: Apple pie is good.
  - Example 3: Lemon pie is even better than apple.
  - What examples will a search for lemon NOT apple find?

- This frequently narrows too much

Combining Boolean Operators

- Queries involving multiple Boolean operators may need parentheses
  - Example 1: Dogs can get a disease.
  - Example 2: Dogs get hair on the furniture.
  - Example 3: Cats also can get a disease.

- What will (dogs OR cats) AND disease find?
- What will dogs OR (cats AND disease) find?
Example

- Database
  - #1: A chicken lays an egg
  - #2: A chicken crosses the road to get an egg
  - #3: A rooster crosses the road to see the hen
  - #4: An egg is on the other side of the road
  - #5: A car hit the chicken on the road

- What will these queries find?
  - Chicken OR egg
  - Chicken AND egg
  - Chicken NOT egg
  - Chicken AND (egg OR road)
  - (Chicken AND egg) OR road

Preparing for a Search

- Formulate the search question
  - What is my topic?
  - What is the purpose of my search?

- Example: History of stock car racing
Identify Important Concepts

- Break your topic up into important concepts
- List terms that are likely to be associated with each concept
- Example
  - Concept 1: history, historical
  - Concept 2: stock car, NASCAR
  - Concept 3: race, races, racing

Prepare Initial Search Query

- Use truncation and Boolean operators
  - Concept 1: histor*
  - Concept 2: “stock car” OR nascar
  - Concept 3: rac*

- Combine concepts with AND operator
  - Query: histor* AND (“stock car” OR nascar) AND rac*
Assess the Results

- Look at the pages in your search results
- Do you see search terms that you should have included?
  - You may want to try another search including these terms as well
- Did you return too many irrelevant pages?
  - You may want to narrow the search