

INLS 760 – Web Databases  
Lecture 8 – Metadata and Full-Text  
Search

Dr. Robert Capra  
Spring 2007

Copyright 2007, Robert G. Capra III

1

Today's Schedule

- Quiz
- Questions about Project 5
- Finish in-class programming of P3
- Metadata and Full-Text Search

Copyright 2007, Robert G. Capra III

2

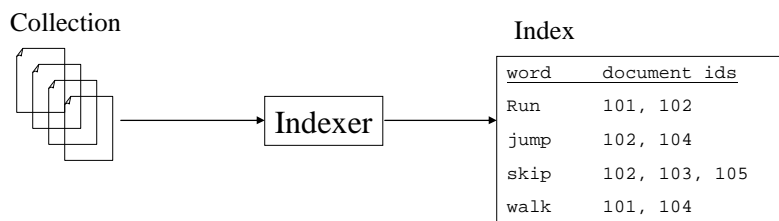
# Search Basics

- Collection
  - A collection of documents
- Document
  - A collection of words
- Word
  - Basic unit
  - Stopwords: a, an, the, of, etc.
  - Stemming: running, runner, ran → run
- Index
  - Mapping of words onto documents (one-to-many)

Copyright 2007, Robert G. Capra III

3

# Search Basics



Search:

```
Results = index['run'] INTERSECT index['jump']
```

Copyright 2007, Robert G. Capra III

4

# Website Search Issues

- Where are the documents stored?
  - Filesystem
  - Database
  - Internet
- Where is the metadata stored?
  - Filesystem
  - Database
  - Internet

|          |     | Documents/Objects   |                  |                        |
|----------|-----|---------------------|------------------|------------------------|
|          |     | FS                  | DB               | Web                    |
| Metadata | FS  | Small apps          | ??               | ??                     |
|          | DB  | Common Digital Libs | Common Ecommerce | Web search engines DLs |
|          | Web | XML                 | XML              | XML                    |

Copyright 2007, Robert G. Capra III

5

# Approaches

- Documents on web
  1. Let a web search engine do all the work
  2. Let a web search engine do some of the work
- Documents on filesystem (or web)
  3. Read text into DB & let the DB do the work
  4. Index documents into DB tables
  5. Index documents using a PHP search engine
  6. Index documents using a search engine package

Copyright 2007, Robert G. Capra III

6

## 1. Let a web search engine do all the work

- Let a web search engine do all the work
  - Ex: Google – site:www.w3.org
  - `http://www.google.com/search?hl=en&q=bgcolor+site:www.w3.org&btnG=Search`
- Pros: easy
- Cons: no control over indexing, presentation

Copyright 2007, Robert G. Capra III

7

## 2. Let web search engine do some of the work

- Web Search Engine + PHP
  - Ex: Yahoo Web Services  
`http://developer.yahoo.com/search/web/V1/webSearch.html`
- 
- `http://api.search.yahoo.com/WebSearchService/V1/webSearch?appid=YahooDemo&query=persimmon&results=7`
- 

```
<?php
$request =
    'http://api.search.yahoo.com/WebSearchService/V1/'
    . 'webSearch?appid=YahooDemo&query='
    . urlencode('inls760')
    . '&site=www.unc.edu';
    . '&results=7';
$response = file_get_contents($request);
echo "<h1>Search Results</h1>";
echo $response;
?>
```

Copyright 2007, Robert G. Capra III

8

### 3. Read text into DB & let the DB do the work

- Idea:
  - Metadata in DB
  - Documents on web or in filesystem
    - Read text of the documents into a field in the DB
    - Read in ONLY the text to be searchable
  - Use MySQL operators LIKE or MATCH
    - LIKE is okay for short metadata fields
    - MATCH does a full-text search
      - Includes stemming, stopwords, etc.
      - <http://dev.mysql.com/doc/refman/4.1/en/fulltext-search.html>

Copyright 2007, Robert G. Capra III

9

### Table to store metadata and body text

```
create table texts (  
    itemnum int unsigned not null  
        auto_increment primary key,  
    title varchar(100),  
    authors varchar(100),  
    publication varchar(100),  
    year varchar(4),  
    url varchar(254),  
    body text,  
    FULLTEXT (authors, title, publication, body)  
);
```

Enables FULL-TEXT searching on these fields

Copyright 2007, Robert G. Capra III

10

## Searching Metadata Fields using LIKE

```
<?php
// should include error checking
$searchby = strip_tags($_GET['searchby']);
$searchstring = strip_tags($_GET['searchstring']);

// connect to db, set up, and execute the query
require_once("/export/home/r/rcapra/dbconnect.php");
$fired = "SELECT * FROM texts where $searchby LIKE '%$searchstring%'";
$result = mysql_query($fired);

// table header
echo '<table border=1 cellpadding=2>';
echo '<th><tr>'; echo '<td>Item #</td>'; echo '<td>Author</td>';
echo '<td>Title</td>'; echo '<td>Publication</td>';
echo '<td>Year</td>'; echo '</tr></th>';

// loop through the results, output them into the table
while ($row = mysql_fetch_array($result,MYSQL_ASSOC)) {
    echo '<tr><td>' . $row['itemnum'] . "</td>";
    echo '<td>' . $row['authors'] . "</td>";
    echo '<td><a href="' . $row['url'] . '">' . $row['title'] . '</a></td>';
    echo '<td>' . $row['publication'] . "</td>";
    echo '<td>' . $row['year'] . "</td></tr>";
}
echo '</table>';

mysql_close($db);
?>
<a href="load-body.php">Load body text into DB</a>
```

lect8/browse2.php

Copyright 2007, Robert G. Capra III

11

## Cleaning and loading the body text

```
<?php
// connect to db, set up, and execute the query
require_once("/export/home/r/rcapra/dbconnect.php");
$fired = "SELECT * FROM texts";
$result = mysql_query($fired);

// loop through the results
while ($row = mysql_fetch_array($result,MYSQL_ASSOC)) {
    $itemnum = $row['itemnum'];
    $url = $row['url'];
    $html = file_get_contents($url);
    $notags = strip_tags($html);
    $text = preg_replace('/[^\A-Za-z0-9 ]/', ' ', $notags);

    echo "<hr>";
    echo $url;
    echo "<br>";

    $sethel = "UPDATE texts SET body=' ' . $text .
              ' ' WHERE itemnum = '$itemnum'";
    $result2 = mysql_query($sethel);

    echo $sethel;
}

mysql_close($db);
?>
```

lect8/load-body.php

Note: This example loads body text from documents accessed via URLs, but a similar approach could be used for documents on a filesystem.

Copyright 2007, Robert G. Capra III

12

## Fulltext Searching using MATCH

lect8/browse3.php

```
<?php
// should include error checking
$searchby = strip_tags($_GET['searchby']);
$searchstring = strip_tags($_GET['searchstring']);

// connect to db, set up, and execute query
require_once("/export/home/r/rcapra/dbconnect.php");
$fred = "SELECT * FROM texts WHERE " .
        "MATCH (authors, title, publication, body) " .
        "AGAINST ('" . $searchstring . "')";
$result = mysql_query($fred);

echo '<table border=1 cellpadding=2>'; echo '<th><tr>';
echo '<td>Item #</td>'; echo '<td>Author</td>';
echo '<td>Title</td>'; echo '<td>Publication</td>';
echo '<td>Year</td>'; echo '</tr></th>';

// loop through the results, output them into a table
while ($row = mysql_fetch_array($result,MYSQL_ASSOC)) {
    echo '<tr><td>' . $row['itemnum'] . "</td>";
    echo '<td>' . $row['authors'] . "</td>";
    echo '<td><a href="' . $row['url'] . '>' . $row['title'] . '</a></td>';
    echo '<td>' . $row['publication'] . "</td>";
    echo '<td>' . $row['year'] . "</td></tr>";
}
echo '</table>';
mysql_close($db);
?>
```

Copyright 2007, Robert G. Capra III

13

## 4. Index documents into DB tables

- Idea: “roll your own” search engine using DB tables

documents

| did | title             |
|-----|-------------------|
| 101 | Hansel and Gretel |
| 102 | Tom Thumb         |
| 103 | Rapunzel          |
| ... | ...               |

words

| wid | word |
|-----|------|
| 1   | run  |
| 2   | jump |
| 3   | walk |
| ... | ...  |

occurrences

| wid | did |
|-----|-----|
| 1   | 101 |
| 1   | 102 |
| 2   | 101 |
| 2   | 103 |
| 3   | 103 |
| ... | ... |

This approach is based on:

<http://www.onlamp.com/pub/a/php/2002/10/24/simplesearchengine.html?page=1>

Copyright 2007, Robert G. Capra III

14

## 4. Index documents into DB tables

- Indexer algorithm:

For each document

For each word

If word is already in word table, use existing word\_id

Else, add word to table and get its word\_id

Insert (word\_id, document\_id) into occurrences table

Copyright 2007, Robert G. Capra III

15

## 4. Index documents into DB tables

- Search algorithm:

```
SELECT title
FROM documents, words, occurrences
WHERE words.word = 'searchterm' AND
      words.wid = occurrences.wid AND
      documents.did = occurrences.did
```

Copyright 2007, Robert G. Capra III

16

## 5. Index documents using a PHP search engine

- Examples include:
  - phpdig
  - Sphider

## 6. Index documents using a search engine package

- Ht/dig
- Lucene
- Commercial packages