

Lecture 8 – Database Search

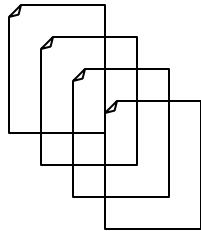
INLS 760
Web Databases
Spring 2013
Rob Capra

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)

Search Basics

Collection



Indexer

Index

<u>word</u>	<u>document ids</u>
Run	101, 102
jump	102, 104
skip	102, 103, 105
walk	101, 104

Search: `run and jump`

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

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

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

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

2. Let web search engine do some of the work

- Google Custom Web Search
 - http://code.google.com/apis/customsearch/v1/getting_started.html
- Bing Search API
 - <http://www.bing.com/toolbox/bingdeveloper/>

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/5.1/en/fulltext-search.html>

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)  
);
```

lect8/texts/create-texts.sql

* can load with MySQL
source command



Enables FULL-TEXT searching on these fields

Metadata Records

lect8/texts/texts-metadata.txt

The Love Song of J. Alfred Prufrock|T.S. Eliot|Prufrock and Other Observations|1917|
http://www.ils.unc.edu/courses/2009_spring/inls760_001/lect8/texts/tseliot/lovesongjap.txt

Portrait of a Lady|T.S. Eliot|Prufrock and Other Observations|1917|
http://www.ils.unc.edu/courses/2009_spring/inls760_001/lect8/texts/tseliot/portofalady.txt

Preludes|T.S. Eliot|Prufrock and Other Observations|1917|
http://www.ils.unc.edu/courses/2009_spring/inls760_001/lect8/texts/tseliot/preludes.txt

Rhapsody on a Windy Night|T.S. Eliot|Prufrock and Other Observations|1917|
http://www.ils.unc.edu/courses/2009_spring/inls760_001/lect8/texts/tseliot/rhapwindynight.txt

Morning at the Window|T.S. Eliot|Prufrock and Other Observations|1917|
http://www.ils.unc.edu/courses/2009_spring/inls760_001/lect8/texts/tseliot/mornwindow.txt

The Boston Evening Transcript|T.S. Eliot|Prufrock and Other Observations|1917|
http://www.ils.unc.edu/courses/2009_spring/inls760_001/lect8/texts/tseliot/bostoneventrans.txt

Aunt Helen|T.S. Eliot|Prufrock and Other Observations|1917|
http://www.ils.unc.edu/courses/2009_spring/inls760_001/lect8/texts/tseliot/aunthelen.txt

Cousin Nancy|T.S. Eliot|Prufrock and Other Observations|1917|
http://www.ils.unc.edu/courses/2009_spring/inls760_001/lect8/texts/tseliot/cousinnancy.txt

Mr. Apollinax|T.S. Eliot|Prufrock and Other Observations|1917|
http://www.ils.unc.edu/courses/2009_spring/inls760_001/lect8/texts/tseliot/mrapollinax.txt

Hysteria|T.S. Eliot|Prufrock and Other Observations|1917|
http://www.ils.unc.edu/courses/2009_spring/inls760_001/lect8/texts/tseliot/hysteria.txt

Conversation Galante|T.S. Eliot|Prufrock and Other Observations|1917|
http://www.ils.unc.edu/courses/2009_spring/inls760_001/lect8/texts/tseliot/convgalante.txt

Load the metadata

lect8/texts/load-metadata.txt

```
load data
local infile "texts-metadata.txt"
replace into table texts
fields terminated by '|'
(title, authors, publication, year, url)
;
```

```
select itemnum, title, authors from texts;
```

```
mysql> source load-metadata.txt
```

Cleaning and loading the body text

lect8/texts/load-body.php

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

// 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>" . $url . "<br>";
    $ethel = "UPDATE texts SET body='" . $text .
        "' WHERE itemnum = '$itemnum'";
    $result2 = mysql_query($ethel);
    echo $ethel;
}
mysql_close($db);
?>
```

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

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");
$fred = "SELECT * FROM texts where $searchby LIKE '%$searchstring%'";
$result = mysql_query($fred);

// 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);
?>
```

lect8/texts/browse2.php

Fulltext Searching using MATCH

lect8/browse3.php

```
<?php
// should include error checking
$searchstring = strip_tags($_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);
?>
```

Try out some fulltext searches

- `browse3.php?searchstring=fingers`
- `browse3.php?searchstring=horses`
- `browse3.php?searchstring=over`

MySQL Full-Text 'gotchas'

- Some things to be aware of:
 - Stopwords are not indexed
 - The default list of stopwords has more than you think
 - <http://dev.mysql.com/doc/refman/5.1/en/fulltext-stopwords.html>
 - Words that occur in 50% or more of the rows are not indexed
 - Words shorter than 4 characters are not indexed

Full-Text 'gotcha' example

lect8/create-swsearch.sql

```
create table swsearch (  
    itemnum int unsigned not null auto_increment primary key,  
    title varchar(100),  
    body text,  
    FULLTEXT (title, body)  
);
```

lect8/insert-swsearch.sql

```
INSERT INTO swsearch (title,body) VALUES  
    ('zzz jedimaster', 'greetings skywalker computer'),  
    ('amidala', 'stormtrooper ewoks computer'),  
    ('yoda', 'anakin vader computer'),  
    ('aaaaaaaaa', 'hansolo tatooine'),  
    ('bbbbbbbbb', 'aaaaaaaaa hoth');
```

gotcha example

itemnum	title	body
1	zzz jedimaster	greetings skywalker computer
2	amidala	stormtrooper ewoks computer
3	yoda	anakin vader computer
4	aaaaaaaa	hansolo tatooine
5	bbbbbbbb	aaaaaaaaa hoth

```
mysql> select * from swsearch where match (title, body) against  
      ('skywalker');
```

```
mysql> select * from swsearch where match (title, body) against  
      ('greetings');
```

```
mysql> select * from swsearch where match (title, body) against  
      ('computer');
```

```
mysql> select * from swsearch where match (title, body) against ('zzz');
```

gotcha example

itemnum	title	body
1	zzz jedimaster	greetings skywalker computer
2	amidala	stormtrooper ewoks computer
3	yoda	anakin vader computer
4	aaaaaaaaa	hansolo tatooine
5	bbbbbbbbb	aaaaaaaaa hoth

```
mysql> select * from swsearch where match (title, body) against  
('skywalker');
```

itemnum	title	body
1	zzz jedimaster	greetings skywalker computer

```
1 row in set (0.00 sec)
```

```
mysql>
```

```
mysql> select * from swsearch where match (title, body) against  
('greetings');
```

```
Empty set (0.01 sec)
```

```
mysql> select * from swsearch where match (title, body) against  
('computer');
```

```
Empty set (0.00 sec)
```

```
mysql> select * from swsearch where match (title, body) against ('zzz');
```

```
Empty set (0.00 sec)
```

MySQL Fulltext Search Notes

- See <http://dev.mysql.com/doc/refman/5.1/en/fulltext-search.html>
- When to create fulltext indices
 - in the create table statement
 - later, using alter table or create index
- Types of MySQL fulltext search
 - Boolean (+foo –bar)
 - Natural Language (what we've been covering)
 - Query Expansion (performs two search steps)

Multiple Fulltext Indices

lect8/texts/create-twoidx.sql

```
create table twoidx (  
    itemnum int unsigned not null auto_increment primary key,  
    title varchar(100),  
    abstract varchar(200),  
    body text,  
    FULLTEXT (title, abstract),  
    FULLTEXT (body)  
);
```

lect8/texts/insert-twoidx.sql

```
INSERT INTO twoidx (title,abstract,body) VALUES  
    ('aaaa','skywalker','greetings r2d2 computer'),  
    ('bbbb','amidala','stormtrooper amidala computer'),  
    ('cccc','yoda','hansolo amidala computer'),  
    ('dddd','ben','tatooine hoth computer'),  
    ('eeee','lando','vader skywalker ewok');
```

```
select * from twoidx where match (title, abstract)  
against ('amidala');
```

4. Index documents into DB tables

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

documents

<u>did</u>	<u>title</u>
101	Hansel and Gretel
102	Tom Thumb
103	Rapunzel
...	...

words

<u>wid</u>	<u>word</u>
1	run
2	jump
3	walk
...	...

occurrences

<u>wid</u>	<u>did</u>
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>

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

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
```

5. Index documents using a PHP search engine

- Examples include:
 - phpdig
 - Sphider

6. Index documents using a search engine package

- Ht/dig
- Lucene/SOLR
- Commercial packages

Parsing XML

```
<videos>
  <video videoid="744">
    <title>The Future of Engery Gases, segment 03 of 13</title>
    <year>1993</year>
    <genre>Documentary</genre>
  </video>
  <video videoid="5539">
    <title>Oil Today - Power Tomorrow</title>
    <year>1950</year>
    <genre>Ephemeral</genre>
  </video>
  <video videoid="5434">
    <title>Golden Gate City</title>
    <year>0</year>
    <genre>Ephemeral</genre>
  </video>
</videos>
```

lect8/ov.xml

lect8/parse1.php

```
<?php
    $myvideos = simplexml_load_file('ov.xml');
    echo $myvideos->video[0][videoid];
    echo ":\t";
    echo $myvideos->video[0]->title;
    echo "\n";
    echo "\n";
?>
```

lect8/parse2.php

```
<h1>Open Video Listing</h1>
<table border="1">
<th>
    <td>VideoID</td>
    <td>Title</td>
    <td>Year</td>
</th>

<?php
$ovurl = "http://www.open-video.org/details.php?videoid=";
$myvideos = simplexml_load_file('ov.xml');
foreach ($myvideos as $myvideo) {
    echo "<tr>\n";
    echo "    <td>\n";
    echo '        <a href="' . $ovurl . $myvideo[videoid] . '">' .
        $myvideo[videoid] . '</a>';
    echo "    </td>\n";
    echo "    <td>" . $myvideo->title . "</td>\n";
    echo "    <td>" . $myvideo->year . "</td>\n";
    echo "    <td>" . $myvideo->genre . "</td>\n";
}
?>
```

Concepts from: <http://blog.teamtreehouse.com/how-to-parse-xml-with-php5>