

CS520 Web Programming

Full Text Search with Lucene

Chengyu Sun
California State University, Los Angeles

Search Text

- ◆ Web search
- ◆ Desktop search
- ◆ Applications
 - Search posts in a bulletin board
 - Search product descriptions at an online retailer
 - ...

Database Query

- ◆ Find the posts regarding "SSHD login errors".

```
select * from posts  
where content like '%SSHD login errors%';
```

Here are the steps to take to fix the SSHD login errors:
...

Please help! I got SSHD login errors!

Problems with Database Queries

Please help! I got an error when I tried to login through SSHD!

There a problem recently discovered regarding SSHD and login. The error message is usually ...

The solution for sshd/login errors: ...

- ◆ And how about performance??

Full Text Search (FTS)

- ◆ More formally known as Information Retrieval (IR)
- ◆ Deals with the *representation, storage, organization, and access* of LARGE quantity of textual data.

Characteristics of FTS

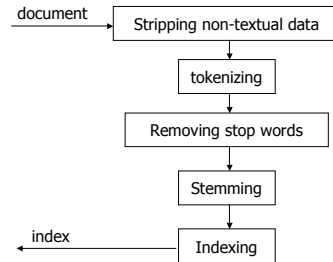
- ◆ Vs. database
 - "Fuzzy" query processing
 - Relevancy ranking

Accuracy of FTS

$$\text{Precision} = \frac{\text{\# of relevant documents retrieved}}{\text{\# of documents retrieved}}$$

$$\text{Recall} = \frac{\text{\# of relevant documents retrieved}}{\text{\# of relevant documents}}$$

Journey of a Document



Document

◆ Original

```

<html>
<body>
<p>The solution for
ssh/login errors:
...</p>
</body>
</html>
  
```

◆ Text-only

```

The solution for
ssh/login errors:
...
  
```

Tokenizing

```

[the] [solution] [for] [ssh] [login] [errors]
...
  
```

Stop Words

◆ Words that do not help in search and retrieval

- n Function words: a, an, and, the, of, for ...
- n Domain specific: "to be or not to be"

◆ After stop words removal:

```

[the] [solution] [for] [ssh] [login] [errors]
...
  
```

Stemming

◆ Reduce a word to its stem or root form.

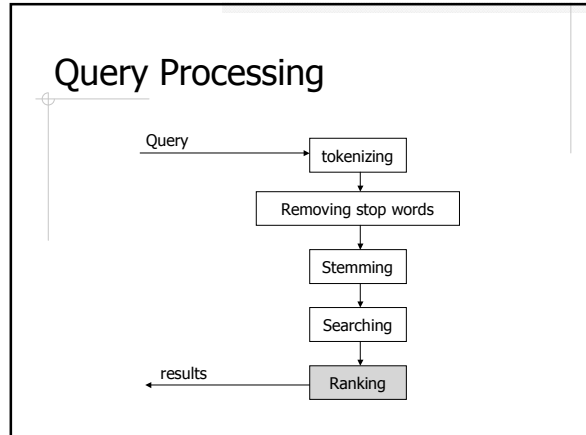
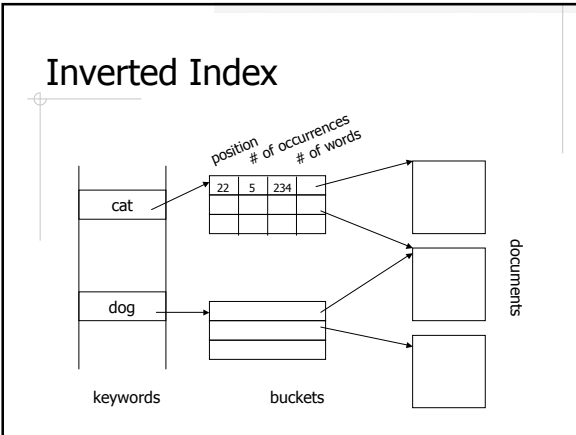
◆ Examples:

```

connection, connections
connected, connecting  →  connect
connective
  
```

```

[solution] [ssh] [login] [errors] → [solve] [ssh] [login] [error]
...
  
```



- ### Ranking
- ◆ How well the document matches the query
 - E.g. weighted vector distance
 - ◆ How "important" the document is
 - E.g. based on ratings, citations, and links

- ### FTS Implementations
- ◆ Databases
 - MySQL: MyISAM tables only
 - PostgreSQL: tsearch2 module; OpenFTS
 - Oracle, DB2, MS SQL Server
 - ◆ Standard-alone IR libraries
 - Lucene, Egothor, Xapian, MG4J, ...
 - ◆ *Database vs. Standard-alone Library??*

- ### Lucene Overview
- ◆ <http://lucene.apache.org/>
 - ◆ Originally developed by Doug Cutting
 - ◆ THE full text search solution for Java applications
 - ◆ Handles text only – needs external converters to convert other document types to text

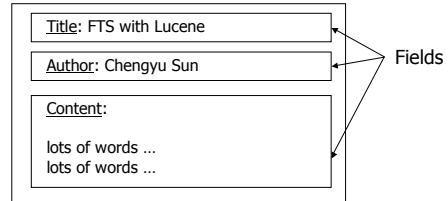
- ### Example 1: Index Text Files
- ◆ Directory
 - ◆ Document and Field
 - ◆ Analyzer
 - ◆ IndexWriter

Directory

- ◆ A place where the index files will be stored
- ◆ `FSDirectory` – file system directory
- ◆ `RAMDirectory` – virtual directory in memory

Document

- ◆ A document consists of a number of user-defined fields



Field

Field	Analyzed (Tokenized)	Indexed	Stored
<code>Field.Keyword(String,String)</code>		Y	Y
<code>Field.Keyword(String,Date)</code>		Y	Y
<code>Field.Unindexed(String,String)</code>			Y
<code>Field.Unstored(String,String)</code>	Y	Y	
<code>Field.Text(String,String)</code>	Y	Y	Y
<code>Field.Text(String,Reader)</code>	Y	Y	

The API for Field was changed in Lucene 2.0.

Analyzer

- ◆ Pre-processing the document or query text – tokenization, stop words removal, stemming
- ...
- ◆ Lucene built-in analyzers
 - `WhitespaceAnalyzer`, `SimpleAnalyzer`, `StopAnalyzer`
 - `StandardAnalyzer`
 - Grammar-based
 - Recognize special tokens such as email addresses
 - Handle CJK text

Analyze Chinese Text

- ◆ Unigram
 - `Lucene StandardAnalyzer`
 - `MySQL`, `PostgreSQL`
- ◆ Bigram
 - `Lucene CJKAnalyzer`
- ◆ Grammar-based
 - Usually in commercial products

Chinese Text Example

Text: 今天天气不错。

Unigram:
[今][天][天][气][不][错]

Bigram:
[今天][天天][天气][气不][不错]

Grammar-based:
[今天][天气][不错]

IndexWriter

- ◆ addDocument(Document)
- ◆ close()
- ◆ optimize()

Example 2: Search

- ◆ Query and QueryParser
- ◆ IndexSearcher
- ◆ Hits
- ◆ Document (again)

Query and QueryParser

Query ::= (Clause)*

Clause ::= ["+", "-"] [<TERM> ":"] (<TERM> | "(" Query ")")

Sample Queries

full text search

+full +text search

+full +text -search

+title:"text search"

+(title:full title:text) -author:"bob dole"

IndexSearcher

- ◆ search(Query)
- ◆ close()

Hits

- ◆ A ranked list of documents used to hold search results

- ◆ Methods

- Document doc(int n)

- int id(int n)

- int length()

- float score(int n)

Document (again)

- ◆ Methods to retrieve data stored in the document
 - String get(String name)
 - Field getField(String name)

Handle Rich Text Documents

- ◆ HTML
 - NekoHTML, JTidy, TagSoup
- ◆ PDF
 - PDFBox
- ◆ MS Word
 - TextMining
- ◆ More at Lucence FAQ -
<http://wiki.apache.org/jakarta-lucene/LuceneFAQ>

Example: FTS in Evelyn

- ◆ Indexer and Searcher interface
- ◆ FileHandler interface
- ◆ File handler implementations
 - DefaultFileHandler
 - TextFileHandler
 - HtmlFileHandler
 - PdfFileHandler
- ◆ Spring beans configuration

Further Readings

- ◆ *Lucene in Action* by Otis Gospodnetic and Erik Hatcher
- ◆ Lucene documentation –
<http://lucene.apache.org/java/docs/index.html>