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

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??
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

1. Stripping non-textual data
2. Tokenizing
3. Removing stop words
4. Stemming
5. Indexing

Document

- **Original**
  
```
  <html>
  <body>
    <p>The solution for sshd/login errors: ...
    </p>
  </body>
  <html>
```

- **Text-only**
  
```
  The solution for sshd/login errors: ...
  ...
```

Stop Words

- Words that do not help in search and retrieval
  - Function words: a, an, and, the, of, for ...
  - Domain specific: “to be or not to be”

After stop words removal:

```
[solution] [sshd] [login] [errors]
```

Tokenizing

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

Stemming

- Reduce a word to its stem or root form.
- Examples:

```
connection, connections
connected, connecting
connective
```

```
[solution] [sshd] [login] [errors] \rightarrow [solve] [sshd] [login] [error]
...
```
Inverted Index

Query Processing

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
- 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

<table>
<thead>
<tr>
<th>Title: FTS with Lucene</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author: Chengyu Sun</td>
</tr>
</tbody>
</table>

| Content: lots of words ... |

Field

<table>
<thead>
<tr>
<th>Field</th>
<th>Analyzed (Tokenized)</th>
<th>Indexed</th>
<th>Stored</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field.Keyword(String,String)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Field.Keyword(String,Date)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Field.Unindexed(String,String)</td>
<td></td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>Field.Unstored(String,String)</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Field.Text(String,String)</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Field.Text(String,Reader)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Field.Text(String,Reader)

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


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