INFORMATION RETRIEVAL SYSTEMS
IV B.Tech (CSE)-I Sem

By
Mr. K. Yellaswamy
Assistant Professor
Department of Computer Science & Engineering
CMR College of Engineering & Technology
Information Retrieval Systems

• Information
  What is “information”?

• Retrieval
  What do we mean by “retrieval”?
  What are different types information needs?

• Systems
  How do computer systems fit into the human information seeking process?
Information Hierarchy

Data

Information

Knowledge

Wisdom

More refined and abstract
Information Hierarchy

- **Data**
  The raw material of information

- **Information**
  Data organized and presented in a particular manner

- **Knowledge**
  “Justified true belief”
  Information that can be acted upon

- **Wisdom**
  Distilled and integrated knowledge
  Demonstrative of high-level “understanding”
A (Facetious) Example

- **Data**
  98.6° F, 99.5° F, 100.3° F, 101° F, …

- **Information**
  Hourly body temperature: 98.6° F, 99.5° F, 100.3° F, 101° F, …

- **Knowledge**
  If you have a temperature above 100° F, you most likely have a fever

- **Wisdom**
  If you don’t feel well, go see a doctor
What types of information?

- Text
- Structured documents (e.g., XML)
- Images
- Audio (sound effects, songs, etc.)
- Video
- Programs
- Services
Outline of Unit-1

- Definition of IR Systems
- Objectives of IR Systems
- Functional Overview
- Relationship to DBMS
Definition of IR Systems

• An IR System is a system capable of storage, retrieval, and maintenance of information.
  Information: text, image, audio, video, and other multi-media objects
  Focus on textual information here

Item:
The smallest complete textual unit processed and manipulated by an IR system
Depend on how a specific source treats information
  ■ Book? Chapter? Paragraph?
‘Item’ and ‘Document’ are used interchangeably
An IR system facilitates a user in finding the information the user needs.

Success measure (Objectives of an IR System)
Minimize the overhead for finding information
Overhead: The time a user spends in all of the steps leading to reading an item containing needed information, excluding the time for actually reading the relevant data
- Query generation
- Search composition
- Search execution
- Scanning results of query to select items to read
Supporting the Search Process

Source Selection -> Query Formulation -> Search

IR System

Query Reformulation and Relevance Feedback

Query

Search -> Ranked List

Nominate

Selection

Examination

Choose

Document

Delivery

Reselection

Predict

Nominate

Choose
Supporting the Search Process

Source Selection → IR System → Query Formulation → Search → Ranked List → Selection → Examination → Document Delivery

- Acquisition → Indexing → Collection → Index → Query → Document
Objectives of IR Systems
Overview

• The general objective of an IR system is to minimize the overhead of a user locating needed information
• The two major measures commonly associated with information systems are precision and recall
• Support of user search generation
• How to present the search results in a format that facilitate the user in determining relevant items
Basic Measures for Text Retrieval

- **Precision**: the percentage of retrieved documents that are in fact relevant to the query (i.e., “correct” responses)
  
  \[
  \text{precision} = \frac{|\{\text{Relevant}\} \cap \{\text{Retrieved}\}|}{|\{\text{Retrieved}\}|}
  \]

- **Recall**: the percentage of documents that are relevant to the query and were, in fact, retrieved

  \[
  \text{precision} = \frac{|\{\text{Relevant}\} \cap \{\text{Retrieved}\}|}{|\{\text{Relevant}\}|}
  \]
Precision and Recall

\[
\text{Precision} = \frac{\text{Number Retrieved Relevant}}{\text{Number Total Retrieved}}
\]

\[
\text{Recall} = \frac{\text{Number Retrieved Relevant}}{\text{Number Possible Relevant}}
\]
Measuring Precision and Recall

Assume there are a total of 14 relevant documents

<table>
<thead>
<tr>
<th>Hits 1-10</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1</td>
<td>1/2</td>
<td>1/3</td>
</tr>
<tr>
<td>1/4</td>
<td>2/5</td>
<td>3/6</td>
</tr>
<tr>
<td>4/8</td>
<td>4/9</td>
<td>4/10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hits 11-20</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/11</td>
<td>5/12</td>
<td>5/13</td>
</tr>
<tr>
<td>5/14</td>
<td>6/16</td>
<td>6/17</td>
</tr>
<tr>
<td>6/18</td>
<td>6/19</td>
<td>6/20</td>
</tr>
</tbody>
</table>

= relevant document
Graphing Precision and Recall

- Plot each (recall, precision) point on a graph.
- Visually represent the precision/recall tradeoff.
Precision and Recall (Cont.)

• Precision
  Measures retrieval overhead for a particular query
  In the WWW-world, precision is more important than recall

• Recall
  How well a system is able to retrieve the relevant items for users

• Ideal Precision and Recall
Two More Objectives of IR Systems

- Support of user search generation
  - How to specify the information a user needs
    - Language ambiguities – “field”
    - Vocabulary corpus of a user and item authors
  - Must assist users automatically and through interaction in developing a search specification that represents the need of users and the writing style of diverse authors

- How to present the search results in a format that facilitate the user in determining relevant items
  - Ranking in order of potential relevance
  - Item clustering and link analysis…
Functional Overview
Functional Overview

• Four major functional process
  Item Normalization
  Selective Dissemination of Information
  Archival Document Database Search
  Index Database Search + Automatic File Build Process (Support index files)
Item Normalization

• Normalize incoming items to a standard format
  Language encoding
  Different file formats...

• Logical restructuring – zoning

• Create a searchable data structure (Indexing)
  Identification of processing tokens
  Characterization of the tokens – single words, or phrase
  Stemming of the tokens
Functional Overview – Item Normalization
Overview

1. Standardize Input
2. Logical Subsetting (Zoning)
3. Identify Processing Tokens
4. Stop Algorithm
5. Characterize Tokens
6. Apply Stemming
7. Create Searchable Data Structure
Standardize Input

- Standardizing the input takes the different external format of input data and performs the translation to the formats acceptable to the system.
- Translate foreign language into Unicode
  Allow a single browser to display the languages and potentially a single search system to search them
- Translate multi-media input into a standard format
  Video: MPEG-2, MPEG-1, AVI, Real Video…
  Audio: WAV, Real Audio
  Image: GIF, JPEG, BMP…
Logical Subsetting (Zoning)

- Parse the item into logical sub-divisions that have meaning to user
  Title, Author, Abstract, Main Text, Conclusion, References, Country, Keyword...

- Visible to the user and used to increase the precision of a search and optimize the display
  The zoning information is passed to the processing token identification operation to store the information, allowing searches to be restricted to a specific zone display the minimum data required from each item to allow determination of the possible relevance of that item (display zones such as Title, Abstract…
Identify Processing Tokens

• Identify the information that are used in the search process – *Processing Tokens (Better than Words)*

• The first step is to determine a word
  Dividing input symbols into three classes
  • Valid word symbols: alphabetic characters, numbers
  • Inter-word symbols: blanks, periods, semicolons (non-searchable)
  • Special processing symbols: hyphen (-)

A word is defined as a contiguous set of word symbols bounded by inter-word symbols
Stop Algorithm

• Save system resources by eliminating from the set of searchable processing tokens those have little value to the search Whose frequency and/or semantic use make them of no use as a searchable token
  • Any word found in almost every item
  • Any word only found once or twice in the database

Frequency * Rank = Constant

Stop algorithm v.s. Stop list
Characterize Tokens

• Identify any specific word characteristics

  Word sense disambiguation
  Part of speech tagging
  Uppercase – proper names, acronyms, and organization
  Numbers and dates
Stemming Algorithm

- Normalize the token to a standard semantic representation
  - Computer, Compute, Computers, Computing
    - Comput
- Reduce the number of unique words the system has to contain
  - ex: “computable”, “computation”, “computability”
  - small database saves 32 percent of storages
  - larger database: 1.6 MB $\rightarrow$ 20%
    - 50 MB $\rightarrow$ 13.5%
- Improve the efficiency of the IR System and to improve recall $\Rightarrow$ Decline precision
  - Expand a search term to similar token representations in run time?
Create Searchable Data Structure

- Processing tokens ➔ Stemming Algorithm ➔ update to the searchable data structure
- Internal representation (not visible to user) Signature file, Inverted list, PAT Tree…
- Contains
  Semantic concepts represent the items in database
  Limit what a user can find as a result of the search
Functional Overview – Selective Dissemination of Information
Selective Dissemination of Information (SDI)

- Provides the capability to dynamically compare newly received items in the information system against standing statements of interest of users and deliver the item to those users whose statement of interest matches the contents of the items

- Consist of
  - Search process
  - User statements of interest (Profile)
  - User mail file
A profile contains a typically broad search statement along with a list of user mail files that will receive the document if the search statement in the profile is satisfied. As each item is received, it is processed against every user’s profile. When the search statement is satisfied, the item is placed in the mail file(s) associated with the process. User search profiles are different than ad hoc queries in that they contain significantly more search terms and cover a wider range of interests.
Functional Overview –
Document Database Search

- Provides the capability for a query to search against all items received by the system
- Composed of the search process, user entered queries and document database.
- Document database contains all items that have been received, processed and stored by the system.
  Usually items in the Document DB do not change.
  May be partitioned by time and allow for archiving by the time partitions.
- Queries differ from profiles in that they are typically short and focused on a specific area of interest.
Functional Overview – Index Database Search

☐ When an item is determined to be of interest, a user may want to save it (file it) for future reference. Accomplished via the index process.

☐ In the index process, the user can logically store an item in a file along with additional index terms and descriptive text the user wants to associate with the item.

  An index can reference the original item, or contain substantive information on the original item.

  Similar to card catalog in a library.

☑ The Index Database Search Process provides the capability to create indexes and search them.
Functional Overview –
Index Database Search (Cont.)

• The user may search the index and retrieve the index and/or the document it references.

• The system also provides the capability to search the index and then search the items referenced by the index records that satisfied the index portion of the query.

  Combined file search

• In an ideal system the index record could reference portions of items versus the total item.
Functional Overview – Index Database Search (Cont.)

- Two classes of index files: public and private index files
  
  Every user can have one or more private index files leading to a very large number of files, and each private index file references only a small subset of the total number of items in the Document database
  
  Public index files are maintained by professional library services personnel and typically index every item in the Document database
  
- The capability to create private and public index files is frequently implemented via a structured Database Management System (RDBMS)
To assist the users in generating indexes, the system provides a process called Automatic File Build (Information Extraction)

Process selected incoming documents and automatically determine potential indexing for the item
- Authors, date of publication, source, and references

The rules that govern which documents are processed for extraction of index information and the index term extraction process are stored in Automatic File Build Profiles

When an item is processed it results in creation of Candidate Index Records ➔ for review and edit by a user prior to actual update of an index file
What about databases?

• What are examples of databases?
  Banks storing account information
  Retailers storing inventories
  Universities storing student grades

• What exactly is a (relational) database?
  Think of them as a collection of tables
  They model some aspect of “the world”
A (Simple) Database Example

Student Table

<table>
<thead>
<tr>
<th>Student ID</th>
<th>Last Name</th>
<th>First Name</th>
<th>Department ID</th>
<th>email</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Kandula</td>
<td>Ashok</td>
<td>EE</td>
<td><a href="mailto:ashok@gmail.com">ashok@gmail.com</a></td>
</tr>
<tr>
<td>2</td>
<td>R</td>
<td>Vishnu</td>
<td>HIST</td>
<td><a href="mailto:vishnu@gmail.com">vishnu@gmail.com</a></td>
</tr>
<tr>
<td>3</td>
<td>Kandula</td>
<td>Srinandan</td>
<td>HIST</td>
<td><a href="mailto:Srinandan@gmail.com">Srinandan@gmail.com</a></td>
</tr>
<tr>
<td>4</td>
<td>Kandula</td>
<td>Yellaswamy</td>
<td>CLIS</td>
<td><a href="mailto:toyellaswamy@gmail.com">toyellaswamy@gmail.com</a></td>
</tr>
</tbody>
</table>

Department Table

<table>
<thead>
<tr>
<th>Department ID</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>EE</td>
<td>Electrical Engineering</td>
</tr>
<tr>
<td>HIST</td>
<td>History</td>
</tr>
<tr>
<td>CLIS</td>
<td>Information Studies</td>
</tr>
</tbody>
</table>

Course Table

<table>
<thead>
<tr>
<th>Course ID</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>lbsc690</td>
<td>Information Technology</td>
</tr>
<tr>
<td>ee750</td>
<td>Communication</td>
</tr>
<tr>
<td>hist405</td>
<td>American History</td>
</tr>
</tbody>
</table>

Enrollment Table

<table>
<thead>
<tr>
<th>Student ID</th>
<th>Course ID</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>lbsc690</td>
<td>90</td>
</tr>
<tr>
<td>1</td>
<td>ee750</td>
<td>95</td>
</tr>
<tr>
<td>2</td>
<td>lbsc690</td>
<td>95</td>
</tr>
<tr>
<td>2</td>
<td>hist405</td>
<td>80</td>
</tr>
<tr>
<td>3</td>
<td>hist405</td>
<td>90</td>
</tr>
<tr>
<td>4</td>
<td>lbsc690</td>
<td>98</td>
</tr>
</tbody>
</table>
• What would you want to know from a database?

What classes is John Arrow enrolled in?
Who has the highest grade in LBSC 690?
Who’s in the history department?
Of all the non-CLIS students taking LBSC 690 with a last name shorter than six characters and were born on a Monday, who has the longest email address?
## Databases vs. IR

<table>
<thead>
<tr>
<th>What we’re retrieving</th>
<th>Databases</th>
<th>IR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structured data. Clear semantics based on a formal model.</td>
<td>Mostly unstructured. Free text with some metadata.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Queries we’re posing</th>
<th>Databases</th>
<th>IR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formally (mathematically) defined queries. Unambiguous.</td>
<td>Vague, imprecise information needs (often expressed in natural language).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Results we get</th>
<th>Databases</th>
<th>IR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact. Always correct in a formal sense.</td>
<td>Sometimes relevant, often not.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interaction with system</th>
<th>Databases</th>
<th>IR</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-shot queries.</td>
<td>Interaction is important.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other issues</th>
<th>Databases</th>
<th>IR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concurrency, recovery, atomicity are all critical.</td>
<td>Issues downplayed.</td>
<td></td>
</tr>
</tbody>
</table>