Introduction to Information Retrieval

CMSC 491/691-I
Ian Soboroff
Overview

- What is Information Retrieval?
- Searching
- Indexing
- Course Overview
What is Information Retrieval?

- Finding needles in haystacks
  - Haystacks are pretty big (the Web, the LOC...)
  - Needles can be pretty vague ("find me anything about..."
  - Lots of kinds of hay (text, images, video, audio...)
- Compare a user’s query to a large collection of documents, and give back a ranked list of documents which best match the query
Text REtrieval Conference (TREC) Home Page
Call to TREC 2001 The TREC Conference series is co-sponsored by the NIST, Information Technology ...
Description: Conference series co-sponsored by NIST and DARPA.
Category: Computers > Software > Information Retrieval
trec.nist.gov/ - 3k - Cached - Similar pages

Text REtrieval Conference (TREC) TREC-6 Proceedings
NIST Special Publication 500-240: The Sixth Text REtrieval Conference (TREC 6). ... NOTE: Portions ...
trec.nist.gov/pubs/trec6/t6_proceedings.html - 22k - Cached - Similar pages
[ More results from trec.nist.gov ]

Overview of the Seventh Text REtrieval Conference (TREC-7) - ...
Overview of the Seventh Text REtrieval Conference (TREC-7) (1998) (Correct) (2 citations)
Eli M. Hauptmann, Brookhaven National Laboratory, SUNY Stony Brook
Searching for Free

- UNIX gives you great tools to search for stuff
  - **grep**: find lines in files matching an expression
  - **wc**: count words/lines/characters in a file
  - **sort**: sort lines
  - **uniq**: cut out (or count) duplicate lines
  - **tr** and **sed** for modifying text
  - **awk** and **perl** are the Swiss Army Knives of UNIX
  - **pipes** tie it all together
Searching your E-mail

- Pine (and others) keep mail in *mbox* files
  - one message after the other, starts with "From 
  - RFC 822 gives the gory details on email headers
- Find mail from Seth
  ```
grep -n 'From:.*Seth' mail
```
prints lines containing 'From… Seth' with line numbers
Searching your E-mail II

- Search by name, print name and subject
  cat mail | awk ' BEGIN { found = 0 } /From:.*Seth/ { found = 1; print } /Subject/ && found == 1 { found = 0; print }'

- awk programs are 'pattern { action } ...'
Searching your E-mail III

- Make an address book from your e-mail mailbox
  ```
  cat mail | grep '^From:' | cut -d: -f2- | sort | uniq -c | sort -n
  ```

- This also counts the number of e-mails from each sender
SHEBOYGAN: a Simple HypertExt Bookmark Organizer Using Grep and Names

- Idea: let you grep your Netscape bookmarks
  - Save bookmarked pages, with URL as filename
  - Searching with grep …
    
    cd ~/bookmarks
grep Linux *
  
  … prints out URLs of bookmarks!
Automating SHEBOYGAN

- Problem: have to save bookmarks manually!

- Solution
  - Extract bookmarks from
    ~/.netscape/bookmarks.html (using 'sed')
  - Use 'lynx -dump' to download pages automatically
  - Use 'cron' to do it every evening
Problems with SHEBOYGAN

- Have to look through every file
  - Query might only contain terms which occur in one or two documents
  - Very inefficient if we have many documents
- The Right Thing is to have a structure which reduces our search time
The Library Approach

• Solution: organize documents into a hierarchical structure
  • Put similar documents into directories
  • Only grep in directories which are related to query
• What should the categories be?
• Where do we put each document?
Indexing

- Solution: build an **index** of terms
  - Array of terms
    - The "dictionary"
  - Each term points to a list of documents that the term occurs in
    - The "postings"
Searching an Index

- Find query terms in index
- Only search documents which are in the query term lists

Query: "human being"
Ranking Documents in an Index

1. Chocolate mousse pie
2. Chocolate chip cookies
3. Spinach Pie
4. Baklava

"I want to bake something with chocolate"
The IR Process

Information need

Documents

Search

Text DB

Index

Query
Course Overview

- First half: fundamentals
  - indexing, search models, implementation
- Second half: beyond the basics
  - advanced models, filtering, Web search, user interfaces
The Project

- Write your own search engine
  - Phase I: indexing
  - Phase II: searching
  - Phase III: up to you!
- faster, more effective, good interface, hypertext search, dynamic DB, …
Project Benchmarks

1. Time/space to index a small collection
2. A larger collection
3. A larger coll. with test queries
   • Measure efficiency AND effectiveness!
4. Results posted on the web site