INFO I427: Search Informatics  
(aka: Google under the hood)  
Fall 2013

Course details
- Techniques and tools used to automatically crawl, parse, index, store, rank, and search Web information
  - Using Perl 5 and UNIX
  - In other words, you'll build your own (simple) version of Google!
- Meetings
  - Lecture: Tuesdays 9:30-10:45 in Info West 107
  - Lab: Thursdays 9:30-10:45 in Info West 109
- Prerequisites
  - Info I211 and I308, OR CS C211 and C212
  - Or an equivalent amount of experience with programming

Resources
- Oncourse site
  - Announcements, reading assignments, contact info, etc.
- Textbook
- Highly recommended: A book on Perl
  - E.g. Schwartz, "Learning Perl." 3rd edition (2001) is fine and very cheap - $0.01 used on Amazon!
  - Other books are fine; some are available online for free

Grading and assignments
- Grading
  - 40% Programming projects
  - 20% Final project
  - 25% Final exam
  - 15% Labs and quizzes (in class, unannounced)
- Assignments
  - ~4 programming projects
  - May be done individually or in groups of 2
  - May submit up to 48 hours late with a 10% penalty

Final project
- Option 1: Implement a working search engine
  - A (much) simpler version of Google
- Option 2: Term paper on a search-related topic of your choice

Course staff
- Prof. David Crandall
  - djcran@indiana.edu
  - Informatics West 227
  - Office hours: Wednesdays 2-3
- AI: Jangwon Lee
  - leejang@indiana.edu
  - Office and office hours TBA
Warning!

• This course requires a significant amount of programming

• This is not the course for you if …
  – You hate programming
  – You’ve never done any programming before

• This could be the course for you if …
  – You don’t know Perl or Unix, but you’d like to learn
  – You’ve done some programming and would like more practice
  – You haven’t done any programming in a while, and would like to refresh your knowledge
  – You’re willing to invest some extra time in learning and practicing … but you need to proactively seek help when necessary!

Topics

• Introduction
• Perl and Unix
• Web crawling
• Indexing
• Retrieval and ranking
• Search engine optimization
• Ethical and legal implications of search
• Current & ongoing work

Academic integrity

• Read and understand the AI policy on syllabus
• We will look for and prosecute AI violations
• Be especially careful with homework assignments
  – You may discuss homework problems at a high level (e.g. general strategies for solving problems), but you may not share code, and you must cite the other student in your submission
  – If you use ideas or code from another source (like a webpage or textbook) you must acknowledge the source in your submission

Why take this course?

1. How to organize and search for information is a fundamental problem of Informatics.

Welcome to the information age!

• e-mail
• telephone calls
• television
• web pages
• books
• academic papers
• newspapers
• music
• movies
• medical records
• instant messages
• RFID
• satellite images
• financial records
• surveillance video
• tweets & blogs

Estimates from [IDC08, Bohn08]

1 exabyte = 1,000 petabytes = 1,000,000 terabytes = 1,000,000,000 gigabytes = 1 × 10^{18} bytes

Web search: the challenges

• There are >1,000,000,000,000 web pages in the world, and >1,000,000,000 new pages every day [Google08]
  – You want to find it
• There’s no central directory of web pages
• There’s no way of detecting when a new site comes online
• You want search results very quickly (<1 second)
• ~2 billion other Internet users want their results quickly
• There are malicious sites
• There are many, many websites that could be returned for any search query
1990 - First web browser

1993 - Mosaic, 26 web servers

1994 - Rapid growth, 1500+ servers


- Human-edited directories
  - Jerry’s Guide to the World Wide Web

- Early crawlers and indexers
  - WWW Worm (110,000 pages, 1500 queries/day in 1994)
  - Lycos
  - WebCrawler

- Altavista - 1995
  - By 1997, indexed 100 million documents, 20 million queries per day
  - But search results were becoming worse...

The problem with keyword search

- The web has grown exponentially
  - 1994: >100,000 web pages
  - 2008: >1,000,000,000,000 pages

- Computer speed, storage has also grown exponentially

- But human processing capacity hasn’t changed
  - Fixed human processing speed
  - Fixed set of possible keywords
  - Fixed attention span
Enter Google

- Larry Page and Sergey Brin, 1997
  - Commercialized a new way of ranking pages, based on the web’s structure

Why take this course?

1. How to organize and search for information is a fundamental problem of Informatics.
2. Web search raises interesting CS problems, pushing hardware and software to the limits.

Google servers, 1998

Google servers, 2008

- Estimated to include:
  - 36 data centers
  - 800,000 servers
  - > 10 petabytes of data
  - > 100 petaflops of processing power

- A single user query takes ~0.25sec, uses 700-1,000 machines
Why take this course?

1. How to organize and search for information is a fundamental problem of Informatics.
2. Web search raises interesting CS problems, pushing hardware and software to the limits.
3. There’s a tremendous amount of money (and many, many jobs) in the search industry.
4. Web search has become a fundamental part of our society and economy.
Why take this course?

1. How to organize and search for information is a fundamental problem of Informatics.
2. Web search raises interesting CS problems, pushing hardware and software to the limits.
3. There’s a tremendous amount of money (and many, many jobs) in the search industry.
4. Web search has become a fundamental part of our society and economy.
5. Perl and Linux/Unix experience will be valuable, no matter your career path.

Welcome to the information age!

Welcome to the information age!

Estimates from [IDC08, Bohn08]

1 exabyte = 1,000 petabytes = 1,000,000 terabytes = 1,000,000,000 gigabytes = 1 × 10^18 bytes

How does the Web work?

How does the Web work?

Client (say, your machine)

http://www.britneyspears.com/

Server (say, www.britneyspears.com)

What’s going on under the hood?

What’s going on under the hood?

Client: "http://www.britneyspears.com"

Server (www.britneyspears.com)
So how does a search engine like Google work?

1. Crawl the Web (lots of pages!)
2. Build index
3. When a query arrives, retrieve hits
4. Rank hits & show to user

In truth, “Me Against The Music” is hardly about declaring war against grooves. “Actually, it’s about the intensity that people approach music with,” Britney shares. “It’s about getting totally lost in the music and pushing yourself to the edge in every way you can imagine. I love thoroughly immersing myself in music, and I wanted to capture that intensity in a song.”
Is text enough?

How does Google rank hits?

Quick Survey

- On the card, please write:
  - Your name
  - Your major
  - Something you hope to learn in this class
  - A concern you have about the class
  - How much programming experience do you have? Which language(s)?

- Then please rate your familiarity with each of the following, using a scale from 1 (never heard of) to 5 (very familiar with):
  1. Unix
  2. Perl
  3. PageRank
  4. HTTP
  5. Emacs
  6. Breadth-first search