COMS E6111
Advanced Database Systems
Spring 2020

Computer Science Department
Columbia University
Your Instructor: Luis Gravano

- Ph.D. in Computer Science, Stanford U.
- Professor, Computer Science Department (at Columbia U. since Fall 1997)
- At Google: Senior Research Scientist (2001), Visiting Faculty Researcher (2018-19)

- Research interests: Databases, Web Search, Information Extraction, Social Media
Class Resources

- Class website: http://www.cs.columbia.edu/~gravano/cs6111/
- Discussion board: Piazza, which you can access from CourseWorks, at https://courseworks.columbia.edu/
- Announcements from class staff: on CourseWorks
Your Instructor: Luis Gravano

- gravano@cs.columbia.edu
- 706 Schapiro CEPSR
- +1-212-939-7064

Office hours:

- Thursdays, 9:30-11:30 a.m., or by appointment by email

This week only: Friday (January 24), 9:30-11:30 a.m.
instead of Thursday (January 23)

- Instructor’s office hours always held in 706 Schapiro CEPSR
Your IAs

- Shubham Agrawal
- Tomás Larrain
- Ruoyi Nie

- IA office hours are held in the CS TA/CA Room
- IA office hours will be listed on class webpage
Class Information: Prerequisites

- COMS W4111—Introduction to Databases (equivalent courses taken elsewhere are acceptable as well)
- Fluency in Java or Python

You need permission from the instructor if you don’t have the prerequisites

Note that COMS W4112 is **not** a prerequisite
Class Information: Lectures

- Tuesdays, 2:10-4:00 p.m.
- 1127 Mudd
Grading Information

- **Midterm** (Tue Mar 10, in class): 25%
  Covers all lectures and required readings; closed book

- **Final** (Tue May 12, 1:10-2:40 p.m.): 25%
  Covers all lectures after midterm and required readings; closed book; **not cumulative**

- **Projects** (3, all equally weighted): 50%

Median grade will be a B+ or slightly higher

Alternative or make-up exams will not be given
3 Projects, in Teams of 2 Students

- You will do the projects on the Google Cloud platform
- You will have more-than-enough free credit through individual codes that you will receive once enrollments stabilize
- You get to choose your teammate, and you can change teammates with each project
- You will receive further team-related information with first project
- You can use Java or Python (your choice)

More details announced soon; please be patient and wait until projects announced
Project Policies

- To be fair to all students in the class, I will grant no extensions or exceptions for project submission.
- Instead, you have three grace late days total for projects that you can use as you wish throughout the semester; weekends and university holidays are not counted.
- After using all grace days, you will get a 25% grade deduction for each additional late day.

Check full details on [website](#)
Collaboration Policy

- Please check “Collaboration and Academic Honesty” page from the main webpage for the class.
- Exams are to be done individually.
- Projects are done in teams; you cannot collaborate or share any code with other teams.
- We will not tolerate cheating, which would be wrong and unfair to the rest of the class. Check the CS Department policies and procedures regarding academic honesty at http://www.cs.columbia.edu/education/honesty; they fully apply to this course.
- Contact the instructor right away if you have any questions.
Ongoing Feedback

- Don’t wait until the end-of-semester course evaluations to complain or give feedback on how to improve the course. (It’s too late then!)
- Come see the instructor early on during office hours or send email with your concerns and suggestions; or use the IAs to forward them to the instructor
Topics Covered (Subject to Change)

- Information Retrieval
- Web Search
- Information Extraction
- Data Mining
- OLAP
- Non-traditional DBMSs
- ...
Information Retrieval: Text Databases

- Objects are text documents
- User queries are usually less “precise” than in the relational world

Examples:
- The archive of a newspaper
- Your email
- A web search engine
- ...
Answering Queries?

Key issue:
Ranking documents in order of expected relevance for a given query
Indexes?

- Traditional RDBMS indexes (B+-trees, hash indexes) not appropriate

- Instead, inverted files:
  - “gravano”: doc$_1$, doc$_3$, ...
  - “databases”: doc$_1$, doc$_4$, ...
  - ...

16
Web Search Engines

- How do they work?
- How can we exploit the link structure of the web?
- What are their latest tricks and trends?
Data Mining

Goal: To find interesting trends or patterns in large datasets

Examples:
- Identify target customers for junk mail
- Decide what to place next to beer on supermarket shelves
- Recommend products to online shoppers
- … and much, much more
Required Readings

- No required textbook

- Papers and materials available freely online, including some chapters from:
  - Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze: *Introduction to Information Retrieval*, 2008
  and others

- Required readings for each lecture listed on course schedule webpage
Information Retrieval

Goal: Search over a “database” of text documents

- web pages
- library catalog entries
- research papers
- news articles
- email, files, …
- ...

Query? In simplest incarnation, a list of words:
[ data mining ]
Information Retrieval

Problem: Mapping from a user’s “information need” into the items in the text database that satisfy the need

… or/and, most recently, into information extracted or synthesized from the database contents
IR Challenges?

First, what should answer to a keyword query be?

Definition not as crisp as in RDBMSs:
Employees(ssn, name, age, sal, ...)
SELECT *
FROM Employees E
E.age<40 AND E.sal>50K;

Not up to discussion! “Find those employees under 40 years of age who make above 50K per year”
IR Challenges?

First, what should answer to a keyword query be?

In contrast, what should answer to query [data mining] be?
Information Retrieval Models (many!)

- Boolean (“set theoretic”)
- Vector Space (“algebraic”)
- Probabilistic
- Language Modeling
- Deep Learning
- …
Boolean Retrieval Model

Example queries:

[information AND retrieval]

[data AND mining]

[query AND optimization AND (RDBMS OR “relational databases”)]
Boolean Retrieval Model

Connectives:

Proximity operators:
Boolean Retrieval Model

Fuzzy match operators:

Stemming:
Boolean Retrieval Model

Thesaurus expansion:

Stop-word elimination:
Side (but relevant) discussion: Zipf’s Law
How much of all this should be done automatically, without user control?
Evaluation of Query Result Quality?

- In RDBMSs?
- In Information Retrieval?
Human Relevance Judgments

Is document d relevant or not for query q?
… needs to be determined by a human

Given the results a system returns for a query, need to analyze their quality. The more relevant documents there, the better, of course.
Precision@k

Answer to a query:

d1
d2
d3
d4
d5
d6
d7
d8
d9
d10
Effect of user intent in evaluation framework?

Different classes of queries might have different requirements