Information Retrieval

CISC437/637, Lecture #23

Ben Carterette

Text Search

- Consider a database consisting of long textual information fields
  - News articles, patents, web pages, books, …
- What is the best way to search within this data?
  - grep for keywords in it?
  - Store it in a DBMS and use SQL queries with LIKE '%keyword1%' AND LIKE '%keyword2%'…?
- If there's enough data, these approaches are very slow and very inefficient
  - Hash and B+-tree indexes don't help

Information Retrieval

- **Information retrieval** (IR) studies systems for indexing and querying large full-text corpora
  - Google is the most widely-known modern example
- Work in IR and DB has mostly been separate
  - IR has roots in library science and information science going back to the 1950s, today allied with AI
  - DB is more firmly rooted in algorithms and systems
- In recent years they have begun to intersect via XML, text mining, data mining

IR systems vs DBMS

- Both involve queries that are matched to records (possibly using an index) to retrieve results
- After that, many differences:

IR	DBMS
Relevance semantics	Relational semantics
Keyword search	Full SQL query language
Unstructured data	Structured data
Read-only (mostly)	Read/write
Rank best-matching results	Return full result set

Common Topics, Different Focus

- IR and DB have many things in common, but focus differs between the two:

DBMS	IR
Users	Users
Query language	Query language
Query/record matching	Query/record matching
———	Record ranking
Building indexes	Building indexes
Indexing strategies	Indexing strategies
Query optimization	Query optimization
Concurrency control	———

Relevance and Ranking

- In a DBMS, records either match a SQL query or they don't—there is no middle ground
- In IR systems, some documents can be better matches than others
  - Matching documents may not be relevant
  - Documents that don't match may be relevant
- **Relevance** describes the usefulness of a document to a particular user

