

# Information Retrieval

## Introduction

# Course Outline

- Introduction
- Retrieval Strategies
- Retrieval Utilities
- Indexing and Efficiency Issues
- Integrating Structured Data and Text
- Distributed IR
- Cross Language IR (CLIR)
- The Text Retrieval Conference (TREC)

# Introduction to IR

- Database vs. Information Retrieval
- Why is IR so hard?
- How do we evaluate an IR system?
- Main Components
- High-level introduction to IR Techniques:
  - Overview of Retrieval Strategies
  - Overview of Utilities
- Overview of TREC
- References

# Database vs. Information Retrieval

	<b>Structured Data (Transactional)</b>	<b>Structured Data (Data Warehouse)</b>	<b>Text Data</b>
<b>Accuracy</b>	100%	100%	~30-40%
<b>Query Language</b>	SQL	SQL, OLAP	Natural language
<b>Volumes</b>	5 TB	~200TB	~200TB (Web) 15-20%
<b>Foundation</b>	Algorithm	Algorithm	Heuristics
<b>Validation</b>	Objective	Objective	Subjective

# Definitions

- A *database* is a collection of documents.
- A *document* is a sequence of terms, expressing ideas about some topic in a natural language.
- A *term* is a semantic unit, a word, phrase, or potentially root of a word.
- A *query* is a request for documents pertaining to some topic.

# Definitions (Cont.)

- An *Information Retrieval (IR) System* attempts to find relevant documents to respond to a user's request.
- The real problem boils down to matching the language of the query to the language of the document.

# Hard Parts of IR

- Simply matching on words is a very brittle approach.
- One word can have a zillion different semantic meanings
  - Consider: Take
  - “take a place at the table”
  - “take money to the bank”
  - “take a picture”
  - “take a lot of time”
  - “take drugs”

# More Problems with IR

- You can't even tell what part of speech a word has:
  - “I saw her duck”
  - A query that searches for “pictures of a duck” will find documents that contain
  - “I saw her duck away from the ball falling from the sky”

# More Problems with IR

- Proper Nouns often use regular old nouns
- Consider a document with “a man named Abraham owned a Lincoln”
- A word matching query for “Abraham Lincoln” may well find the above document.

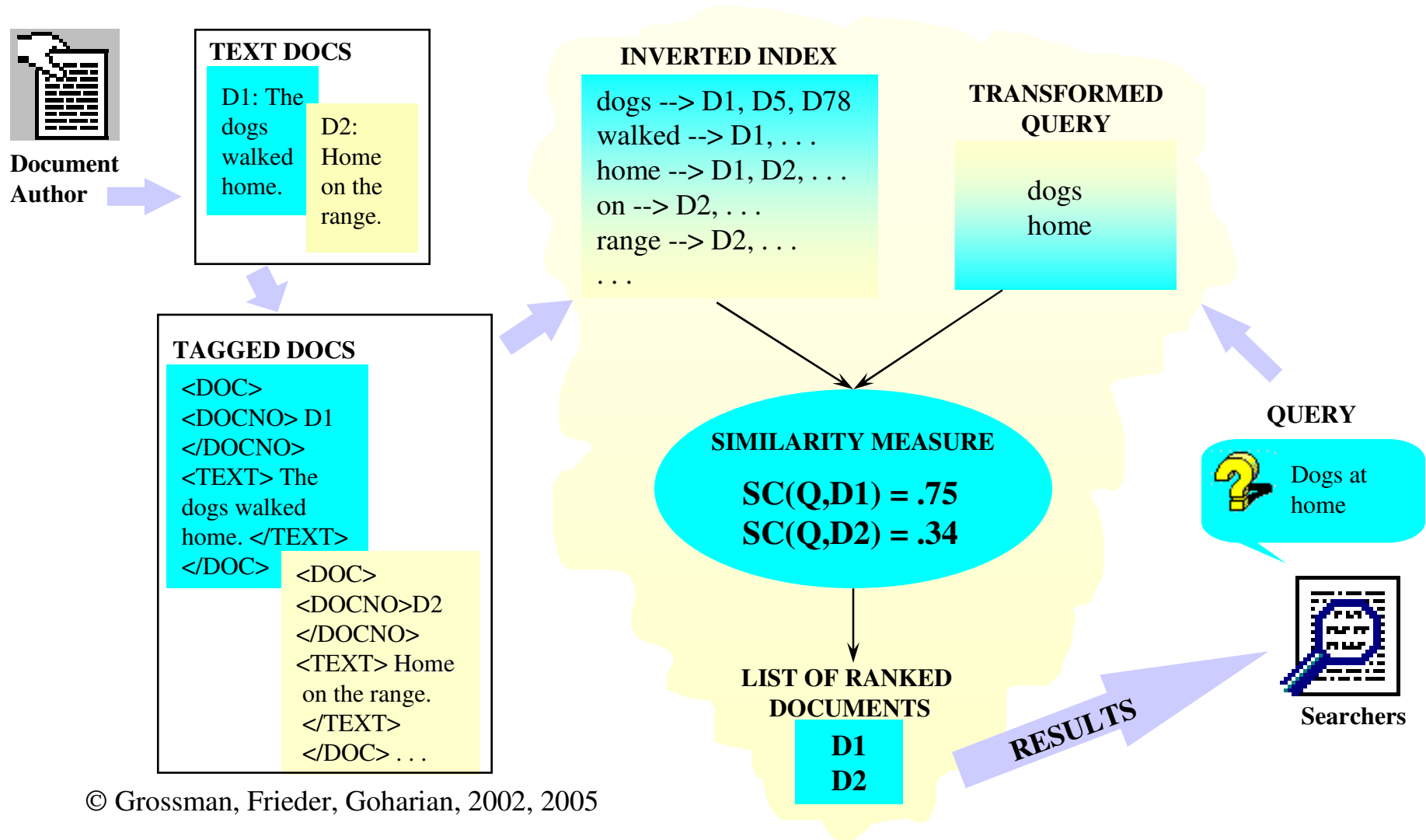
# What is Different about IR from the rest of Computer Science

- Most algorithms in computer science have a “right” answer:
- Consider the two problems:
  - Sort the following ten integers
  - Find the highest integer
- Now consider:
  - *Find the document most relevant to “hippos in the zoo”*

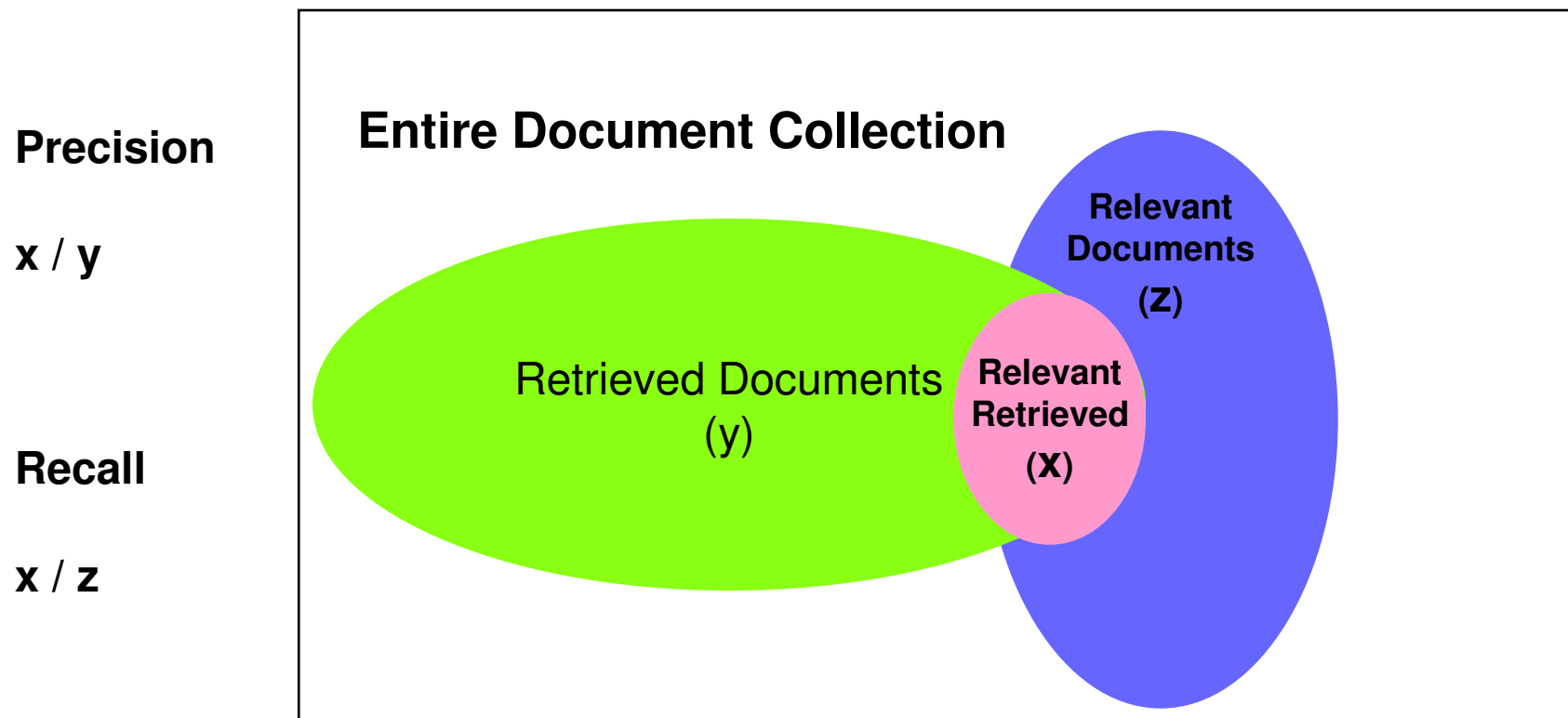
# Measuring Effectiveness

- An algorithm is deemed incorrect if it does not have a “right” answer.
- A heuristic tries to guess something close to the right answer. Heuristics are measured on “how close” they come to a right answer.
- IR techniques are essentially heuristics because we do not know the right answer.
- So we have to measure how *close* to the right answer we can come.

# Measuring Relevance



# Precision / Recall



# Precision / Recall

## Example

- Consider a query that retrieves 10 documents.
- Lets say the result set is.
  - D1
  - D2
  - D3
  - D4
  - D5
  - D6
  - D7
  - D8
  - D9
  - D10
- If all ten were relevant, we would have 100 percent precision.  
If there were only ten relevant in the whole collection, we would have 100 percent recall.

# Example (continued)

- Now lets say that only documents two and five are relevant.
- Consider these results:
  - D1
  - D2**
  - D3
  - D4
  - D5**
  - D6
  - D7
  - D8
  - D9
  - D10
- Since we have retrieved ten documents and gotten two of them right, precision is 20 percent. Recall is 2 / total relevant in entire collection.

# Levels of Recall

- If we keep retrieving documents, we will ultimately retrieve all documents and achieve 100 percent recall.
- That means that we can keep retrieving documents until we reach  $x\%$  of recall.

# Levels of Recall (example)

- Retrieve top 2000 documents. Lets say there are five total documents relevant.

Document	DocId	Recall	Precision
– 100	A	.20	.01
– 200	B	.40	.01
– 500	C	.60	.006
– 1000	D	.80	.004
– 1500	E	1.0	.003

# Recall / Precision Graph

- Compute precision at .1, .2, .3, ..., 1.0 levels of recall.
- Optimal graph would have straight line -- precision always at 1, recall always at 1.
- Typically, as recall increases, precision drops.

# Evaluating IR

- *Recall* is the fraction of relevant documents retrieved from the set of total relevant documents collection-wide.
- *Precision* is the fraction of relevant documents retrieved from the total number retrieved.
- An IR system ranks documents by SC, allowing the user to trade off between precision and recall.

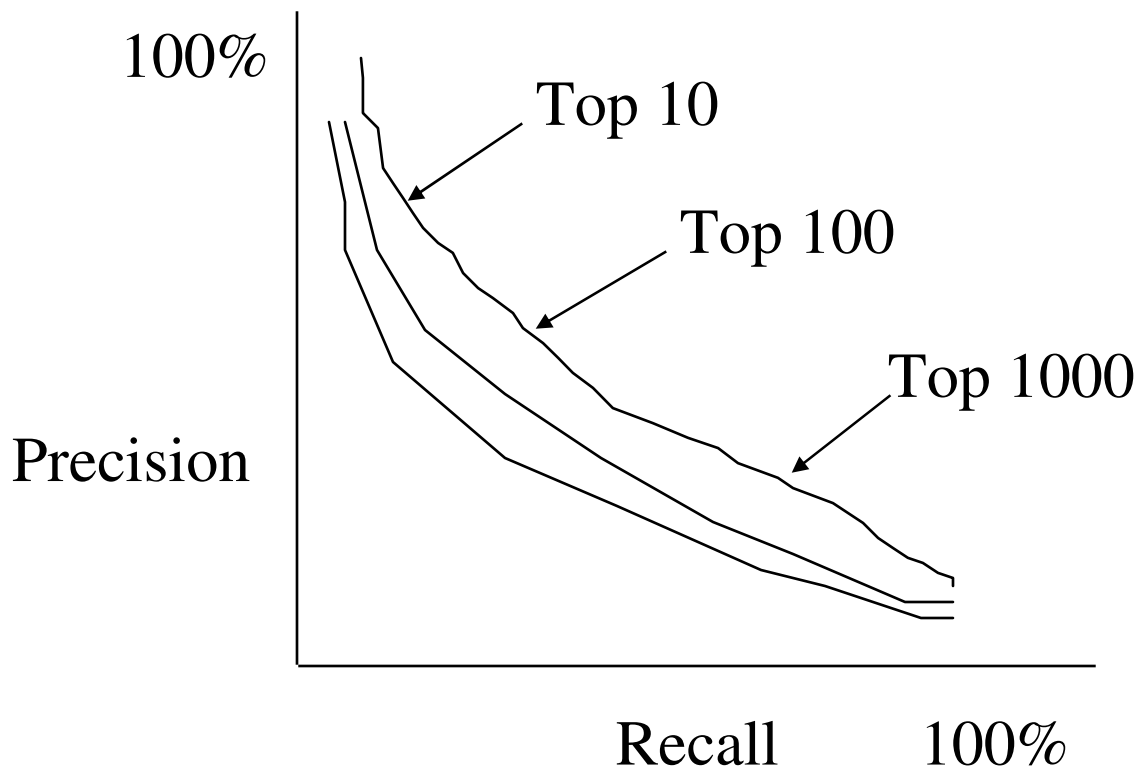
# Example

---

Query: prison overcrowding	SC
D1: Prisoners protest overcrowding at Attica	97%
D2: Commission to study overcrowding in state prisons.	94%
D3: Sales tax to fund prison construction	86%

---

# Precision/Recall Tradeoff



# IR Engine Main Components

- IR engine has two main components
  - Indexing: to index documents
  - Query Processing: to accept and process queries.
- Most IR systems use a structure called an *inverted index* to index documents.

# Requirements

- Scalability
  - Must handle large document collections
- Index Efficiency
  - Must build indexes in a reasonable amount of time
- Query Efficiency
  - Queries must run fast
- Query Effectiveness
  - Result set must be relevant

# Strategy vs. Utility

- An IR *strategy* is a technique by which a relevance assessment (*relevance ranking*) is obtained between a query and a document.
- An IR *utility* is a technique that may be used to improve the assessment (*effectiveness*) given by a strategy. A utility may plug into any strategy.

# Strategies

- Manual
  - Boolean
- Automatic
  - Probabilistic
    - OKAPI, Robertson/Spack-Jones
    - Kwok
  - Language Models
  - Vector Space Model
  - Inference Networks
  - Latent Semantic Indexing (LSI)
- Adaptive Models
  - Genetic Algorithms
  - Neural Networks

# Utilities

- Parsing
- Stemming
- N-grams
- Thesauri
- Relevance Feedback
- Clustering
- Passage-based retrieval
- Semantic Networks

# Efficiency

- Indexing
- Compression
- Index Pruning (Top Doc)
- Efficient Query Processing
- Duplicate Document Detection

# TREC

- Text Retrieval Conference- sponsored by NIST
- A benchmark for evaluating IR systems.
  - Standard document sets (2GB SGML, 10 GB Web HTML, 436 GB, planning to increase to 1 TB.
  - Relevance assignments for 50 queries each year
  - Ad-hoc: evaluation using new queries
  - Routing: evaluation using new documents
  - Other tracks: CLIR, Multimedia, Question Answering, Biomedical Search, etc.
  - Check out: <http://trec.nist.gov/>

# Important IR References (Latest Research Papers on IR)

- ACM SIGIR Conference Proceedings
- ACM Transactions on Information Systems
- ACM Transactions on Database Systems
- ACM SIGMOD Conference
- Conference on Very Large Databases (VLDB)
- Journal of the American Society of Information Science (JASIS)
- Conference on Information and Knowledge Management (CIKM)

# Other IR Books

- Managing Gigabytes, Moffat and Zobel
  - Outstanding book, covers implementation details of IR and Image Retrieval. Very strong on efficiency, not much on effectiveness.
- Information Retrieval, Gerard Salton
  - Classic text -- latest version is 1989.
- Information Retrieval, Baeza-Yates
  - has all the string searching and stemming algorithms as well as a good overview of IR
- Readings in Information Retrieval
  - Contains most of the classic papers on effectiveness, nothing on efficiency.
- Information Retrieval, Jerry Kowalski
  - High level overview of architecture of IR Systems (frequently used at the undergraduate level)