

Information Retrieval Introduction

Course Outline

- Introduction
- Implementation Details of IR Systems
- Retrieval Strategies
- Retrieval Utilities
- Integrating Structured Data and Text
- Parallel IR
- Distributed IR
- The Text Retrieval Conference (TREC)

Introduction to IR

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

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

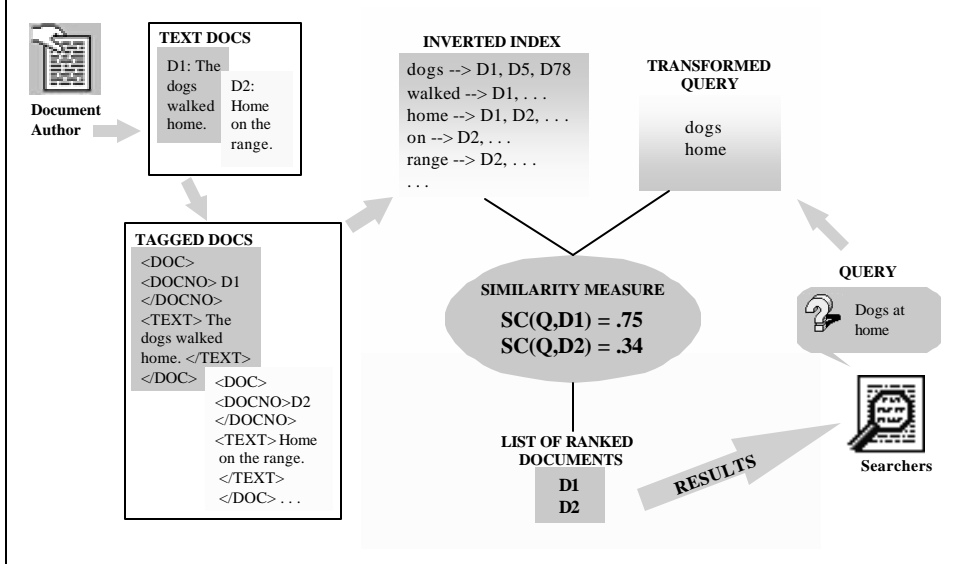
- Most algorithms in computer science have a “right” answer:
- Consider the two problems:
 - Sort the following ten integers
 - Find the highest integer
- Now consider:
 - *Find the document most relevant to “hippos in the zoo”*

Measuring Effectiveness

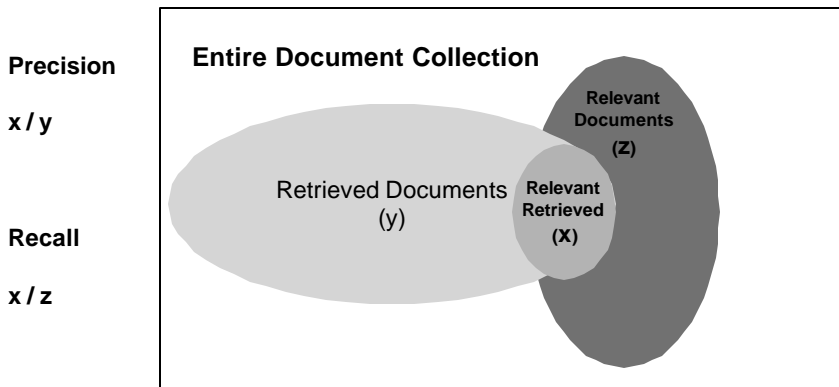
- An algorithm is deemed incorrect if it does not have a “right” answer.
- A heuristic tries to guess something close to the right answer. Heuristics are measured on “how close” they come to a right answer.
- IR techniques are essentially heuristics because we do not know the right answer.
- So we have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

Figure 1

Language of the User is different from the Language of the Author



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.

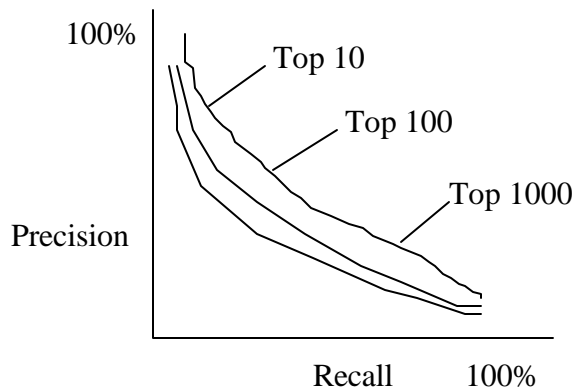
Evaluating IR

- *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



Strategy vs Utility

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

Strategies

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

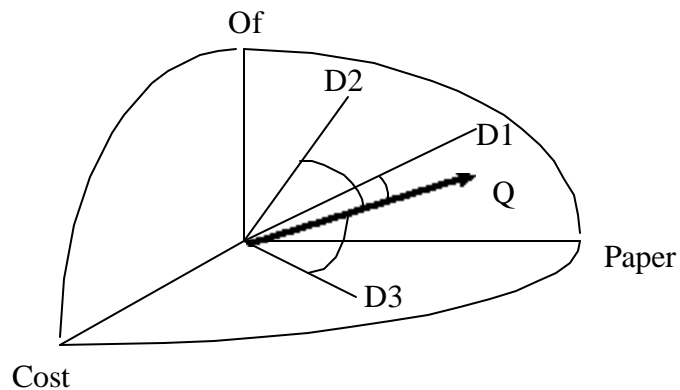
Boolean Queries

- Query: (cost OR price) AND paper
- D1: **Paper cost** increase of 5%. (*relevant*)
- D2: **Price** of jellybeans up 7%. (*not relevant*)

Automatic Strategy

- Query: **cost of paper**
- D1: **Paper cost** increase of 5%.
- D2: **Cost of copper** up 8%. **Cost of aluminum** down 2%.
- D3: **Miracles of modern medicine.**

Vector Space Model

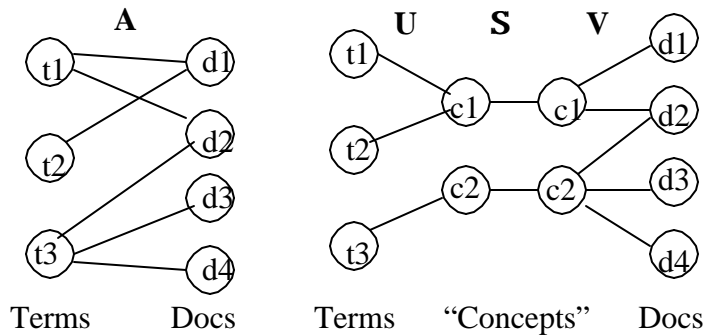


Vector Space Model

- D_{ij}, Q_j equals $tf_{ij} idf_j$
 - tf_{ij} = term frequency of term j in document i
 - idf_j = inverse database frequency of term j
- Usually scaled logarithmically
 - $D_{ij} = \log(tf_{ij} + 1) \log(d/(df_{ij} + 1))$
- Rank by cosine of angle between D_i and Q
 - $SC = \mathbf{D}_i \mathbf{Q} / (|\mathbf{D}_i| |\mathbf{Q}|)$

Latent Semantic Indexing

$A = USV^t$ by singular value decomposition (SVD),
where S is diagonal.



Adaptive Strategy

- Probabilistic
 - Learning based on straight probability estimates.
- Neural Networks
 - Learning based on a model of the brain.
- Genetic Algorithms
 - Learning based on a model of evolution.

Utilities

- Variant forms of terms
 - Stemming, N-grams
- Synonyms
 - Thesauri, semantic nets, relevance feedback, clustering, latent semantic indexing
- Term proximity
 - Passage-based retrieval, parsing

Utilities (Example)

- Query: biological weapons
- D1: Iraqi biologists in weapon program.
- D2: Iraq implicated in germ warfare probe.
- D3: Scientists use **biological** techniques as latest **weapons** against cancer.

Stemming

- *Stemming*: common prefixes and suffixes are removed
 - Biology, biologist, biologists
 - Uses language-dependent rules

N-grams: matching fixed-length strings of *N* characters

- Language independent
- Tolerates misspellings, errors
- Accuracy not as good as using words
- Typically, a two pass matching algorithm is used

Synonyms (Manual)

- A *thesaurus* lists related terms
 - weapon = arms, gun, warfare
- A *semantic net* describes relationships between terms
 - Biologist IS-A scientist
 - Weapon USED-IN war

Synonyms (Automatic)

- Premise: related words are often found in the same document.
 - *Relevance feedback*: terms from the top documents are used to construct a new query.
 - *Clustering*: documents with common terms are grouped.
 - *Latent semantic indexing*: uses a term-document matrix.

Term Proximity

- Premise: Documents are not just bags of words. Query terms are more significant if they occur close together
 - *Passage-based retrieval*: the document is divided into sections (paragraphs, overlapping fixed-length), ranked individually.
 - *Phrases*: pairs of words (or longer sequences) are treated as single terms.
 - *Parsing*: parts of speech (noun phrases, etc.) are identified and treated as terms.

TREC

- Text Retrieval Conference - Meets annually.
- A benchmark for evaluating IR systems.
 - Standard document set, several GB
 - Relevance assignments for 50 queries each year
 - Ad-hoc: evaluation using new queries
 - Routing: evaluation using new documents
 - Other tracks: foreign languages, multimedia, very large corpus, 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)