CS630 Representing and Accessing Digital Information

Information Retrieval: Retrieval Models

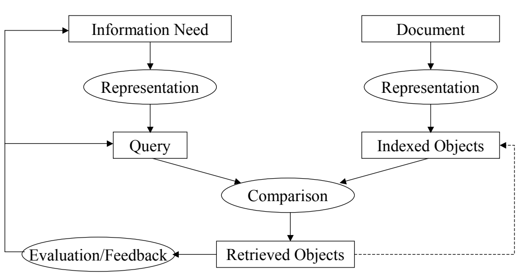
Thorsten Joachims
Cornell University

Based on slides from Jamie Callan and Claire Cardie

Information Retrieval

- Basics
- Data Structures and Access
- Indexing and Preprocessing
- **Retrieval Models**

Basic IR Processes



What is a Retrieval Model?

- **A model is an abstract representation of a process**
  - Used to study properties, draw conclusions, make predictions
  - Quality of the conclusions depends on how closely the model represents reality
- **A retrieval model describes the human and computational processes involved in ad-hoc retrieval**
  - Example: models human information seeking behavior
  - Example: models how documents are ranked computationally
  - Components: users, information needs, queries, documents, relevance assessments, …
  - Retrieval models have notion of relevance, explicitly or implicitly

Major Retrieval Models

- **Boolean**
- **Vector space**
- **Citation analysis models**
- Usage analysis models (later in semester)
- Probabilistic models (partially covered in text classification)

Types of Retrieval Models: Exact Match vs. Best Match Retrieval

- **Exact match**
  - Query specifies precise retrieval criteria
  - Every document either matches or fails to match query
  - Result is a set of documents
    - Usually in no particular order w.r.t. relevance
    - Often in reverse-chronological order
- **Best match**
  - Query describes retrieval criteria for desired documents
  - Every document matches the query to some degree
  - Result is a ranked list of documents, “best” first

Overview

- **Boolean** — **exact match**
- **Vector space** — **best match**
  - Basic vector space
  - Extended boolean model
  - Latent semantic indexing (LSI)
- **Citation analysis models** — **best match**
  - Hubs & authorities
  - PageRank
- **Usage analysis models** — **best match**
  - Direct Hit
  - Ranking SVM
- **Probabilistic models** — **best match**
  - Basic probabilistic model
  - Bayesian inference networks
  - Language models

Exact Match vs. Best Match Retrieval

- **Best-match models are usually more accurate/effective**
  - Good documents appear at the top of the rankings
  - Good documents often don't exactly match the query
    - Query may be too strict
    - Document didn't match user expectations
- **Exact match still prevalent in some markets**
  - Installed base
  - Efficient
  - Sufficient for some tasks
  - Web "advanced search"

Unranked Boolean Retrieval Model

- **Most common Exact Match model**
- **Model**
  - Retrieve documents iff they satisfy a Boolean expression
    - Query specifies precise relevance criteria
  - Documents returned in no particular order
- **Operators**
  - Logical operators: AND, OR, AND-NOT (BUT)
  - Distance operators: near, sentence, paragraph, …
  - String matching operators: wildcard
  - Field operators: date, author, title
- **Unranked Boolean model is not the same as Boolean queries**

Example

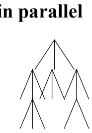
Boolean Query

((((professional OR elite) NEAR/1 competitive NEAR/1 eating) OR (competit* NEAR/1 eat*)) AND (FIELD date 7/4/2002) AND-NOT (weight NEAR/1 loss))

- **Studies show that people are not good at creating Boolean queries**
  - People overestimate the quality of the queries they create
  - Queries are too strict: few relevant documents found
  - Queries are too loose: too many documents found (but few relevant)

Implementation Details

- **Query subtrees can be evaluated in parallel**
  - Use multiple processes
  - Reduce I/O wait time

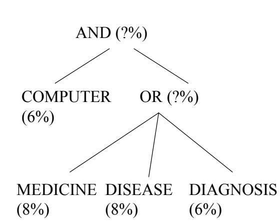


- **Query optimization is very important**
  - Order query by term frequency
  - "fail early" for intersection operators such as AND, proximity

computer (6%) **AND diagnosis** (2%) **AND medicine** (2%) **AND disease** (2%)

Boolean Query Optimization

- **Goal: lower average cost of evaluating query**



computer (6%) **AND (diagnosis** (6%) **OR medicine** (8%) **OR disease** (8%) **)**

Unranked Boolean: WESTLAW

- **Large commercial system**
- **Serves legal and professional markets**
  - Legal: court cases, statutes, regulations, …
  - Public records
  - News: newspapers, magazines, journals, …
  - Financial: stock quotes, SEC materials, financial analyses
- **Total collection size: 5-7 Terabytes**
- **700,000 users**
- **In operation since 1974**
- **Best-match and free text queries added in 1992**

Unranked Boolean: WESTLAW

- **Boolean operators**
- **Proximity operators**
  - Phrases: "Cornell University"
  - Word proximity: language /3 technology
  - Same sentence (/s) or paragraph (/p): Kobayashi /s "hot dog"
- **Restrictions: Date (After 1990 & Before 2002)**
- **Query expansion:**
  - Wildcard: K*ashi
  - Automatic expansion of plurals and possessives
- **Document structure (fields): Title**
- **Citations: Cites (Salton) & Date (After 1998)**

Unranked Boolean: WESTLAW

- **Queries are typically developed incrementally**
  - Implicit relevance feedback

  V1: machine AND learning

  V2: (machine AND learning) OR (neural AND networks) OR (decision AND tree)

  V3: (machine AND learning) OR (neural AND networks) OR (decision AND tree) AND (C4.5 OR Ripper OR EM)
- **Queries are complex**
  - Proximity operators used often
  - NOT is rare
- **Queries are long (9-10 words, on average)**

Unranked Boolean: Summary

- **Advantages**
  - Very efficient
  - Predictable, easy to explain
  - Structured queries
  - Works well when searcher knows exactly what is wanted
- **Disadvantages**
  - Difficult to create good Boolean queries
    - Difficulty increases with size of collection
  - Precision and recall usually have strong inverse correlation
  - Predictability of results causes people to overestimate recall
    - Documents that are "close" are not retrieved

Term Weights: A Brief Introduction

- **The words of a text are not equally indicative of its meaning**

  "Most scientists think that butterflies use the position of the sun in the sky as a kind of compass that allows them to determine which way is north. Scientists think that butterflies may use other cues, such as the earth's magnetic field, but we have a lot to learn about monarchs' sense of direction."
- **Important: butterflies, monarchs, scientists, direction, compass**
- **Unimportant: most, think, kind, sky, determine, cues, learn**
- **Term weights reflect the (estimated) importance of each term**

Term Weights: A Brief Introduction

- **There are many variations on how term weights are calculated**
  - The standard approach for many IR systems is *tf.idf* weights
  - Should include the term frequency
  - $tf_{i,j}$: number of times term *i* occurs in document *j*
  - But terms that appear in many documents in the collection are not very useful for distinguishing a relevant document from a non-relevant one
  - $idf_{i,j}$: inverse document frequency
    - Inverse of the frequency of a term *i* among the documents in the collection

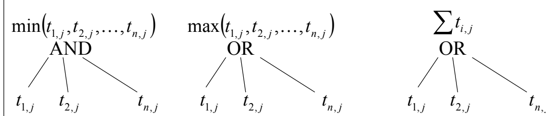
$$tf_{i,j} * idf_{i,j}$$

Ranked Boolean Retrieval Model

- **Ranked Boolean is another common Exact Match retrieval model**
- **Model**
  - Retrieve documents iff they satisfy a Boolean expression
    - Query specifies precise relevance criteria
  - Documents returned ranked by weight of query terms
- **Operators**
  - Logical operators: AND, OR, AND-NOT
    - Unconstrained NOT is expensive, so often not included
    - Distance operators: proximity
    - String matching operators: wildcard
    - Field operators: date, author, title

Ranked Boolean Retrieval

- **How document scores are calculated**
  - Term weight, $t_{i,j}$: function of frequency of query term *i* in document *j*
  - AND weight: minimum of argument weights
  - OR weight: maximum of argument weights
    sum of argument weights



Ranked Boolean Retrieval: Advantages

- **All of the advantages of the unranked Boolean model**
  - Very efficient, predictable, easy to explain, structured queries, works well when searchers know exactly what is wanted
  - Result set is ordered by how "redundantly" a document satisfies a query
    - Usually enables a person to find relevant documents more quickly
  - Variety of term weighting methods can be used
    - tf
    - tf.idf
    - …

Ranked Boolean Retrieval: Disadvantages

- **It's still an Exact Match model**
  - Good Boolean queries are hard to come by
  - Difficulty increases with size of collection
- **Precision and recall usually have strong inverse correlation**
- **Predictability of results causes people to overestimate recall**
  - The returned documents match expectations…
    - …so it is easy to forget that many relevant documents are missed
  - Documents that are "close" are not retrieved

Are Boolean Retrieval Models Still Relevant?

- **Many people prefer Boolean**
  - Professional searchers (e.g. librarians, paralegals)
  - Some Web surfers (e.g. "Advanced Search" feature)
  - About 80% of WESTLAW searches are Boolean
  - What do they like? Control, predictability, understandability
- **Boolean and free-text queries find different documents**
- **Solution: retrieval models that support free-text *and* Boolean queries**
  - Recall that almost any retrieval model can be Exact Match
  - Extended Boolean (vector space) retrieval model
  - Bayesian inference networks

Vector Space Retrieval Model

- **Best Match retrieval**
- **Approach: any text object is represented by a *term vector***
  - Examples: documents, queries, …
- **Similarity is determined by distance in a vector space**
- **The SMART system**
  - Developed at Cornell University, 1960-1999
  - Still used widely

Views of Ad-hoc Retrieval

- **Boolean**
  - Query: a set of FOL conditions that a document must satisfy
  - Retrieval: deductive inference
- **Vector space**
  - Query: a short document
  - Retrieval: finding similar text objects
    - Usually documents
    - Could be passages, sentences, …

Vector Space Retrieval Model: Representation

- **Document representation in the binary model**

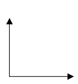
	$Term_1$	$Term_2$	$Term_3$	$Term_4$	…	$Term_n$
Doc_1	**1**	**0**	**0**	**1**	**…**	**1**
Doc_2	**0**	**1**	**1**	**0**	**…**	**0**
Doc_3	**1**	**0**	**1**	**0**	**…**	**0**
…						

- **A document is represented as a vector of binary values**
  - One dimension per term in the corpus vocabulary
- **An unstructured query can also be represented as a vector**

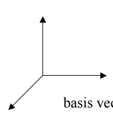
  Query 0 0 1 0 … 1
- **Linear algebra is used to determine which vectors are similar**

Vector Space Representation

- **Documents and queries are vectors in a Real *vector space***
- **Words correspond to orthonormal Basis**
  - Each word correspond to one basis vector (i.e. *direction* in the vector space)
  - Determines what can be described in the vector space
  - Basis vectors are *orthogonal* (=> *linearly independent*), i.e. a value along one dimension (i.e. word) implies nothing about a value along another.



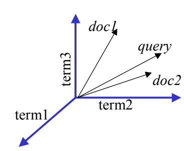
basis vectors for 2 dimensions



basis vectors for 3 dimensions

Vector Space Similarity

- **Similarity is inversely related to the angle between the vectors**



- **Doc2 is more similar to the query**
- **Rank the documents by their similarity to the query**

Vector Space Representation

- **What should be the basis vectors for information retrieval?**
  - "Basic concepts"
    - Difficult to determine
    - Orthogonal (by definition)
    - A relatively static vector space
  - Terms (words, word stems):
    - Easy to determine
    - Not *really* orthogonal (orthogonal enough?)
    - Each term corresponds to one dimension

Document and Query Vectors

- **The vector elements x_i (i.e. term weights) represent term presence, importance, or "representativeness"**

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

- **Some common choices**
  - $x_i=1$ if term is present, $x_i=0$ if term not present in document
  - $x_i=TF$
    - *tf* is a function of the frequency of the term i in the document
  - $x_i=TF * IDF$
    - *TF* is a function of the frequency of the term i in the document
    - *IDF* indicates the discriminatory power of term i

Term Weights Revisited

- **Term frequency (*TF*)**
  - The more often a word occurs in a document, the better that term is in describing what the document is about
  - Has some basis in the 2-Poisson probabilistic model of IR
  - Often normalized, e.g. by the length of the document
  - Sometimes biased to range [0.4..1.0] to represent the fact that even a single occurrence of a term is a significant event

$$TF = \frac{tf}{doc\_length} \quad TF = \frac{tf}{\max tf_d} \quad TF = \frac{tf}{tf + 0.5 + 1.5\frac{doc\_length}{avg\_doc\_length}}$$

Term Weights Revisited

- **Inverse document frequency (*IDF*)**
  - Terms that occur in many documents in the collection are less useful for discriminating among documents
  - Document frequency (*df*): number of documents containing the term
  - *IDF* often calculated as

$$IDF = \log\left(\frac{N}{df}\right) + 1$$

  - Sometimes scaled to [0..1]

$$IDF = \frac{\log\left(\frac{N+0.5}{df}\right)}{\log(N+1.0)}$$

  - *TF* and *IDF* are used in combination as product $x_i = TF * IDF$

Vector Space Similarity

- **Cosine of the angle between the two vectors**
  - Binary term vectors

$$\frac{|X \cap Y|}{\sqrt{|X|}\sqrt{|Y|}}$$

  - Weighted term vectors

$$\frac{\sum x_i y_i}{\sqrt{\sum x_i^2}\sqrt{\sum y_i^2}}$$

Vector Space Similarity: Example

	Term	wts	
Query	**0.0**	**0.2**	**0.0**

	Term	wts	
Doc1	**0.3**	**0.1**	**0.4**
Doc2	**0.8**	**0.5**	**0.6**

$$Sim(D_1, Q) = \frac{(0*0.3)+(0.2*0.1)+(0*0.4)}{\sqrt{0^2+0.2^2+0^2} * \sqrt{0.3^2+0.1^2+0.4^2}} = \frac{0.02}{0.10} = 0.20$$

$$Sim(D_2, Q) = \frac{(0*0.8)+(0.2*0.5)+(0*0.6)}{\sqrt{0^2+0.2^2+0^2} * \sqrt{0.8^2+0.5^2+0.6^2}} = \frac{0.10}{0.22} = 0.45$$

Inverted Index for Vector Space Model

- **Simple algorithm**
  - “word1 OR word2 OR …”
  - Keep track of partial scores in accumulator
  - Might rank 100.000 document just to get the top 10 documents
  - Large memory overhead for high frequency words
- **Refinements to improve efficiency**
  - Compute only the top k documents accurately
  - Process high-weight terms first (e.g. sort inverted lists by decreasing score)
  - Limit number of accumulators (e.g. introduce accumulator only for documents with high-weight term)

Top-Docs Ranking

- **Example:**
  - Find top 1 document only
  - Equal query weights of 1 for both query terms
- **Pruning criteria**
  - Bound on score of single document
  - Remaining maximum weight
- **Relax conditions**
  - Not necessarily optimal
  - Trade time/space for accuray

Term	DocIDs:weight
computer	6:0.7
database	3:0.3
human	2:0.8
learning	2:0.9, 1:0.5, 3:0.1
machine	1:0.7
operating	5:0.8
systems	6:0.3, 5:0.2, 3:0.2
theory	4:0.2

Vector Space Similarity: Summary

- **Standard vector space**
  - Each dimension corresponds to a term in the vocabulary
  - Vector elements are real-valued, reflecting term importance
  - Any vector (document, query, …) can be compared to any other
  - Cosine correlation is the similarity metric used most often