Search Engines (Information Retrieval)

Motivation and Outline

- Background
  - Definitions, etc.
- The Problem
  - 100,000+ pages
- The Solution
  - Indexing docs
  - Ranking results
- Extensions
  - Relevance feedback, clustering, query expansion, etc.

Search Engine Architecture

- Spider
  - Crawls the web to find pages. Follows hyperlinks. Never stops
- Indexer
  - Produces data structures for fast searching of all words in the pages
- Retriever
  - Query interface
  - Ranking (!!)

Indexing

- Arrangement of data (data structure) to permit fast searching.
- Index in the back of a text book.
  - Author names --- page numbers.
  - Key words --- page numbers.
- The web is a large "book".

Inverted Files

POS	
1	A file is a list of words by position
10	– First entry is the word in position 1 (first word)
20	– Entry 4562 is the word in position 4562 (4562nd word)
30	– Last entry is the last word
36	

FILE

An inverted file is a list of positions by word!

a (1, 4, 40)
entry (11, 20, 31)
file (2, 38)
list (5, 41)
position (9, 16, 26)
positions (44)
word (14, 19, 24, 29, 35, 45)
words (7)
4562 (21, 27)

INVERTED FILE

Inverted Files for Multiple Documents

LEXICON

WORD	NDOCS	PTR
jezebel	20	
jezer	3	
jezerit	1	
jeziah	1	
jeziel	1	
jezliah	1	
jezoar	1	
jezrahliah	1	
jezreel	39	

DOCID OCCUR POS 1 POS 2 ...

"jezebel" occurs 6 times in document 34, 3 times in document 44, 4 times in document 56 . . .

34	6	1	118	2087	3922	3981	5002
44	3	215	2291	3010			
56	4	5	22	134	992		
566	3	203	245	287			
67	1	132					
107	4	322	354	381	405		
232	6	15	195	248	1897	1951	2192
677	1	481					
713	3	42	312	802			

WORD INDEX

Ranking (Scoring) Hits

- Hits must be presented in some order
- What order?
  - Relevance, recency, popularity, reliability?
- Some ranking methods
  - Presence of keywords in title of document
  - Closeness of keywords to start of document
  - Frequency of keyword in document
  - Link popularity (how many pages point to this one)
- Can the page owner control?
- Can you find out what order is used?

Link Popularity

- How many pages link to this page?
  - Popular sites have many links pointing at them.
  - The answer is …. Yahoo!
  - (but what was the question again?)
- You can't just rely on number of links!
  - Need to factor in query words.
  - Need to weight links differently.

Authorities & Hubs

- Authorities: sites (pages) that are pointed to by many links! [in-degree]
- Hubs: sites (pages) that point to many links! [out-degree]
- Assign each page an authority weight & a hub weight. Ie, continuous parameters.
- Bigger weights are "better".

Authorities & Hubs II

- Authority(P)= sum of hub weights of pages that point to P.
- Hub(P)= sum of autority weights of pages that it points to.
- Isn't this definition circular?

Algorithm Outline

- Start w/ seed set of core docs.
- Initialize weights to be equal. Then,
  - Recompute authority weights based on hub weights.
  - Recompute hub weights based on authority weights.
- Repeat until ranking doesn't change much.

Information Retrieval (IR)

- Given a large repository of documents, how do I get at the ones that I want
  - Examples: Lexus/Nexus, Medical reports, AltaVista
- Different from databases
  - Unstructured (or semi-structured) data
  - Information is (typically) text
  - Requests are (typically) word-based

Information Retrieval Task

- Start with a set of documents
- User specifies *information need*
  - Keyword query, boolean expression, high-level description
- System returns a list of documents
  - Ordered according to relevance
- Known as the *ad-hoc retrieval problem*

Basic IR System

- Use word overlap to determine relevance
  - Word overlap alone is inaccurate
- Rank documents by similarity to query
- Computed using *Vector Space Model*

Vector Space Model

- Represent documents as a matrix
  - Words are rows
  - Documents are columns
  - Cell *i,j* contains the number of times word *i* appears in document *j*
  - Similarity between two documents is the cosine of the angle between the vectors representing those words

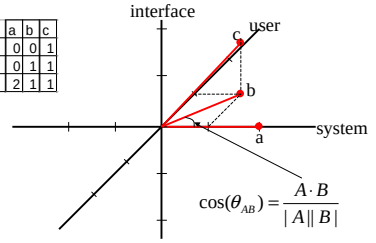
Vector Space Example

a: System and human system engineering testing of EPS
b: A survey of user opinion of computer system response time
c: The EPS user interface management system
d: Human machine interface for ABC computer applications
e: Relation of user perceived response time to error measurement
f: The generation of random, binary, ordered trees
g: The intersection graph of paths in trees
h: Graph minors IV: Widths of trees and well-quasi-ordering
i: Graph minors: A survey

	a	b	c	d	e	f	g	h	i
Interface	0	0	1	0	0	0	0	0	0
User	0	1	1	0	1	0	0	0	0
System	2	1	1	0	0	0	0	0	0
Human	1	0	0	1	0	0	0	0	0
Computer	0	1	0	1	0	0	0	0	0
Response	0	1	0	0	1	0	0	0	0
Time	0	1	0	0	1	0	0	0	0
EPS	1	0	1	0	0	0	0	0	0
Survey	0	1	0	0	0	0	0	0	1
Trees	0	0	0	0	0	1	1	1	0
Graph	0	0	0	0	0	0	1	1	1
Minors	0	0	0	0	0	0	0	1	1

Vector Space Example cont.

	a	b	c
Interface	0	0	1
User	0	1	1
System	2	1	1



Similarity in Vector Space

$$A \cdot B = A_1 B_1 + A_2 B_2 + \ldots + A_n B_n$$

Measures word overlap

$$\cos(\theta_{AB}) = \frac{A \cdot B}{|A||B|}$$

Other metrics exist

Normalizes for different length vectors

$$|A| = \sqrt{\sum_{i=1}^{n} A_i^2}$$

Answering a Query Using Vector Space

- Represent query as vector
- Compute distances to all documents
- Rank according to distance
- Example
  - "computer system"

	Query	a	b	c	d	e	f	g	h	i
Interface	0	0	0	1	0	0	0	0	0	0
User	0	0	1	1	0	1	0	0	0	0
System	1	2	1	1	0	0	0	0	0	0
Human	0	1	0	0	1	0	0	0	0	0
Computer	1	0	1	0	1	0	0	0	0	0
Response	0	0	1	0	0	1	0	0	0	0
Time	0	0	1	0	0	1	0	0	0	0
EPS	0	1	0	1	0	0	0	0	0	0
Survey	0	0	1	0	0	0	0	0	0	1
Trees	0	0	0	0	0	0	1	1	1	0
Graph	0	0	0	0	0	0	0	1	1	1
Minors	0	0	0	0	0	0	0	0	1	1

Common Improvements

- The vector space model
  - Doesn't handle morphology (eat, eats, eating)
  - Favors common terms
- Possible fixes
  - Stemming
    - Convert each word to a common root form
  - Stop lists
  - Term weighting

Handling Common Terms

- Stop list
  - List of words to ignore
    - "a", "and", "but", "to", etc.
- Term weighting
  - Words which appear everywhere aren't very good discriminators
  - Apply a scaling factor to frequencies

Term Weighting

- tf.idf weighting
  - Term frequency-inverse document frequency

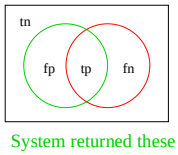
  tf = term frequency df = document frequency
- Family of schemes
  - Depends on details of tf and df scaling
    - Straight frequencies as above
    - log-based weightings
    - $1+\log(tf)$ for $tf>0$ $\log(\frac{N}{df})$

Extensions

- Meet demands of web-based systems
- Modified ranking functions for the web
- Relevance feedback
- Query expansion
- Document clustering
- Latent Semantic Indexing
- Other IR tasks

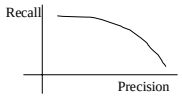
Measuring Performance

- Precision $\frac{tp}{tp+fp}$
  - Proportion of selected items that are correct
- Recall $\frac{tp}{tp+fn}$
  - Proportion of target items that were selected
- Precision-Recall curve
  - Shows tradeoff



System returned these

Actual relevant docs



IR on the Web

- Query AltaVista with "Java"
  - Almost 10^7 pages found
- Avoiding latency
  - User wants (initial) results **fast**
- Solution
  - Rank documents using word-overlap
  - Use special data structure - *inverted index*

Spamdexing

- Attempting to influence retrieval ranking by altering a web page
- Repeating words many times
- Adding words not present on the page
- Adding non-printing words to web pages through HTML <META> tags
- Some engines (Lycos) ignore META tags

Improved Ranking on the Web

- Not just arbitrary documents
- Can use HTML tags and other properties
  - Query term in <TITLE></TITLE>
  - Query term in <IMG>, <HREF>, etc. tag
  - Check date of document (prefer recent docs)
  - Search in a particular IP domain

Relevance Feedback

- System returns initial set of documents
- User identifies relevant documents
- System refines query to get documents more like those identified by user
  - Add words common to relevant docs
  - Reposition query vector closer to relevant docs
- repeat…

Query Expansion

- Given query, add words to improve recall
  - Workaround for synonym problem
- Example
  - boat → boat OR ship
- Can involve user feedback or not
- Can use thesaurus or other online source
  - WordNet

Document Clustering

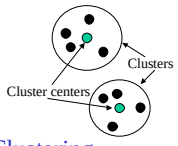
- Group similar documents
  - Similar means "close in vector space"
- If a document is relevant, return whole cluster
- Can be combined with relevance feedback
- GROUPER

  http://www.cs.washington.edu/research/clustering

Clustering Algorithms

- K-means

```
Initialize k cluster centers
Loop
   Assign all document to closest center
   Move cluster centers to better fit assignment
Until little movement
```

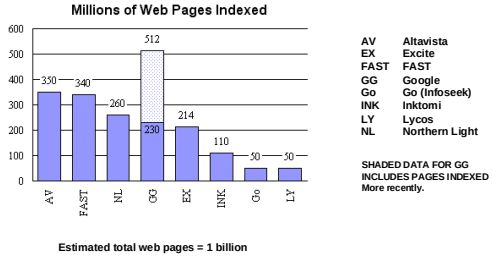


- Hierarchical Agglomerative Clustering

```
Initialize each document to a singleton cluster
Loop
   Merge two closest clusters
Until k clusters exist
```

Many ways to measure distance between clusters

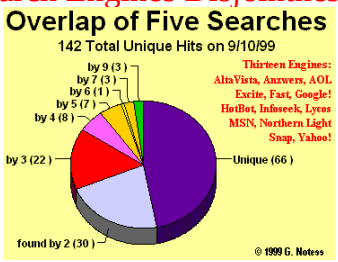
Search Engine Sizes (June 2, 2000)



Estimated total web pages = 1 billion

SOURCE: SEARCHENGINEWATCH.COM

Search Engines Disjointness



SOURCE: SEARCHENGINESHOWDOWN

Conclusions

- Web search engines use IR + web-specific features such as link popularity.
- Search engines are limited in their coverage (I.e., limited recall). Meta-search!
- Search engines often don't get the 'right' results (limited precision). Clustering!