

# Information Retrieval

Books (Available on the web)

Information Retrieval, von Rijsbergen Univ of Glasgow

\* Information Retrieval: Algorithms & Data Structure

Chap?

W. Fratar & R. Baeza-Yates

Modern Information Retrieval: R. Baeza-Yates  
N. ... ?

⇒ acm.org/dl

Conf. ACM SIGIR  
Info. ret.

ACM SIGM  
Information & Knowledge  
management

ACM Trans on Information Systems

" " " Database Systems

Journal of the American Society for Information Science & Technology (JASIST)

Information Processing & Management (IPM)

New event Detection & Tracking

⇒ Efficient & Effective

↳ in terms of time & space

collect

↳ 400,000 dif. news → determine new events

2007 → 9 new resources ⇒ twitter, msn, ... (Collection)

Midterm ⇒ 25 March 2008 ⇒ Tuesday

Final ⇒ 21<sup>st</sup> May - 31<sup>st</sup> May

ad hoc queries → word .. word ..  
queries ↗

they do similarity comparison  
rank from most similar to the less similar

abundance problem

calculate similarity between queries.

Efficient  
effective system

→ relevant documents at the top

System Evaluation:

Recall & Precision

TREC : Text Retrieval Conference

trec eval package provides performance

bpref : binary preference

there is a part of data which has not been evaluated by human being & assumed as irrelevant

Clustering & Cluster Validation:

grouping

tf idf → term frequency inverse document frequency

a word appears in all of the documents → bit, rare  
if a term appears smaller number of times → can be used for differentiation

( fundamental file structures )

information → ( d10, d11, d30, d5 ) ← posting list

retrieval → d1, d10, d51

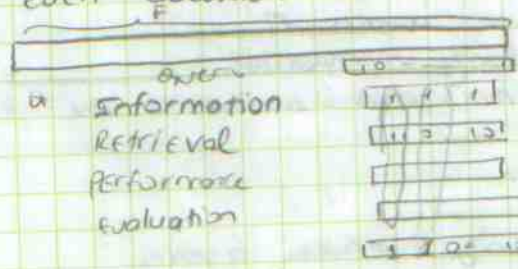
Query Processing:

ACM Computing, surveys, Alistair Moffat, Justin Zobel

Inverted Files for text search engines

Signature Files:

For each document we have a bit array, referred to as signature.



F : size of the array  
→ set random location 1  
not 999999999  
Random Number Generator with a seed

then we superimpose this ⇒ order this columnwise ⇒ 01111

Query signature Document " if (Qs & Ds) = Qs then document is retrieved.

if (Qs & Ds) = Qs  
then document is retrieved

assume you're dealing with huge data

↳ false drop resolution

↳ So how to make this more efficient

N-GRAM

2gram bigram  
3gram trigram

Information: in, nt, fo, or, cm, ma, at, ti, io, on

in \* on → word begins with in & ends with on

co \* m \* t

PAT TREES: (Patricial Trees)

similar to priority queue.  
record everything in telephone conv.  
if you construct pat tree you can get anything any string.

cm snider

(Suffix Trees)

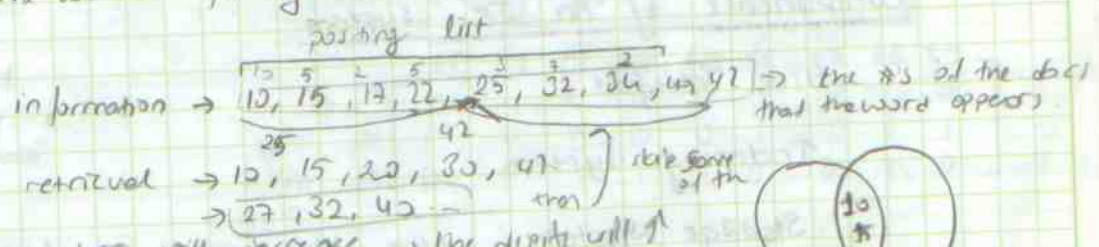
tri

good for intelligent purpose

for movies → pat tree structure you can easily find it (Not so easy to construct)

Compression:

Assume we have posting list



but the numbers will increase → the digits will ↑

Δ gap context document ⇒ by doing so we make the #s more efficient

10 + gap = 15  
15 + gap = 17

say for some reason we begin with retrieval code 17 smaller

with stop first begins with 27

with the stoppy (information) ⇒ we know we can stop first 5 of them since 25 < 27

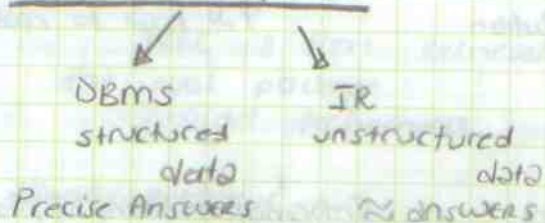
Self-Indexing  
Moffat & Label

ACM, TOIS, 1986  
93

4

# INFORMATION RETRIEVAL SYSTEMS OVERVIEW 3

## Information Systems 2



## Traditional IR Systems 2

MEDLARS provided by national Institutes of Health

MEDLINE online version of MEDLARS

LEXIS/NEXIS law

STAIRS provided by IBM

1993-1994

AltaVista

⋮

Google

SMART

← research IR system

Cornell Univ.  
Gerard Salton

## Components of An IR System

User Interface

Indexing System

Storage System (the file system that we want to use)

Query Processing System

Zobel & Mojtat

↓  
Inverted index for text search engines.

## An Indexing Example:

Doc 1 : Information Retrieval by Parallel Document Ranking

Doc 2 : An analysis of parallel text retrieval systems

Doc 3 : Information retrieval in the law office: an overview.

Stopwords : Frequent words (not good for distinguishing) documents from each other

stopword list

the

and

a

an

the

Cornell

signature forum (christ for)

say we use D<sub>2</sub>

Stopword list:

an  
by  
in  
at  
overview  
systems  
the

Indexing Text Information Retrieval

law  
office

\* uncontrolled

indexing environment

include any word that is in the document as long as it doesn't appear in the stopword list

\* Controlled Indexing

the indexing terms will be provided before hand

MeSH: Medical search Hierarchical?

after a while any word encountered may have been used before



Indexing Terms in Alphabetical Order:

- 1) analysis
- 2) document
- 3) information
- 4) law
- 5) office
- 6) parallel
- 7) ranking
- 8) retrieval
- 9) text

n = 9 (no of words)

m = 3 (no of Documents)

(full of 0's & of 1's are low)

(sparse matrix binary matrix)

$D = \begin{bmatrix} d_1 & d_2 & d_3 \end{bmatrix}$

	t1	t2	t3	t4	t5	t6	t7	t8	t9
d1	0	1	1	0	0	1	0	1	0
d2	1	0	0	0	0	1	0	1	1
d3	0	0	1	1	1	0	0	1	0

$n \times m$   $9 \times 3$

$$d_{ij} = (1 \leq i \leq m, 1 \leq j \leq n) = \begin{cases} 1 & \text{if } t_j \text{ appears in } d_i \\ 0 & \text{otherwise} \end{cases}$$

INSPEC database (online)

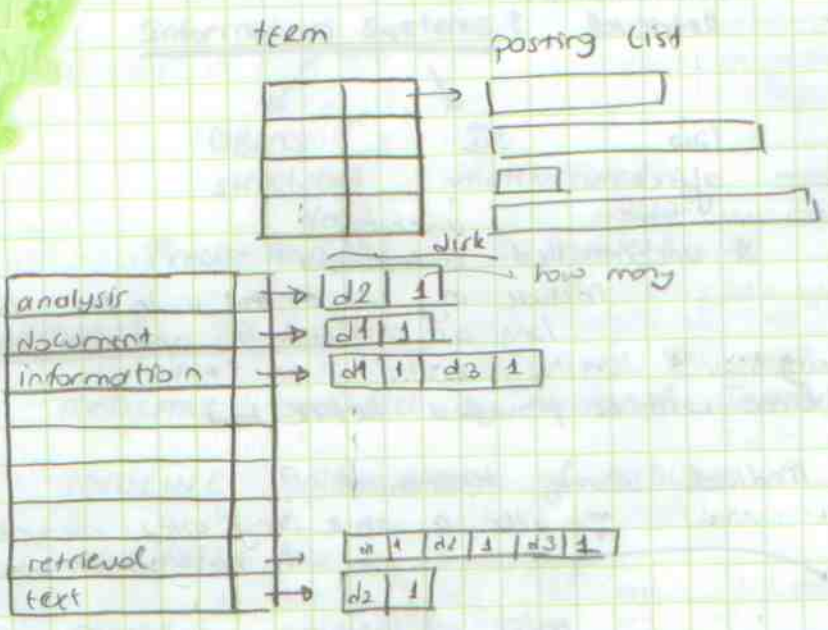
David Lewis

m : 12684  
n : 14573

D matrix of inspec: density of 1's  $\approx 2\%$

ACM,  
IEEE  
MPL

Inverted Index for the Example Collection:



Assume a Boolean Query Environment.

Q: information & retrieval

$$(d1 \ d2) \cap (d1 \ d2 \ d3) = (d1 \ d2)$$

19/10/2008

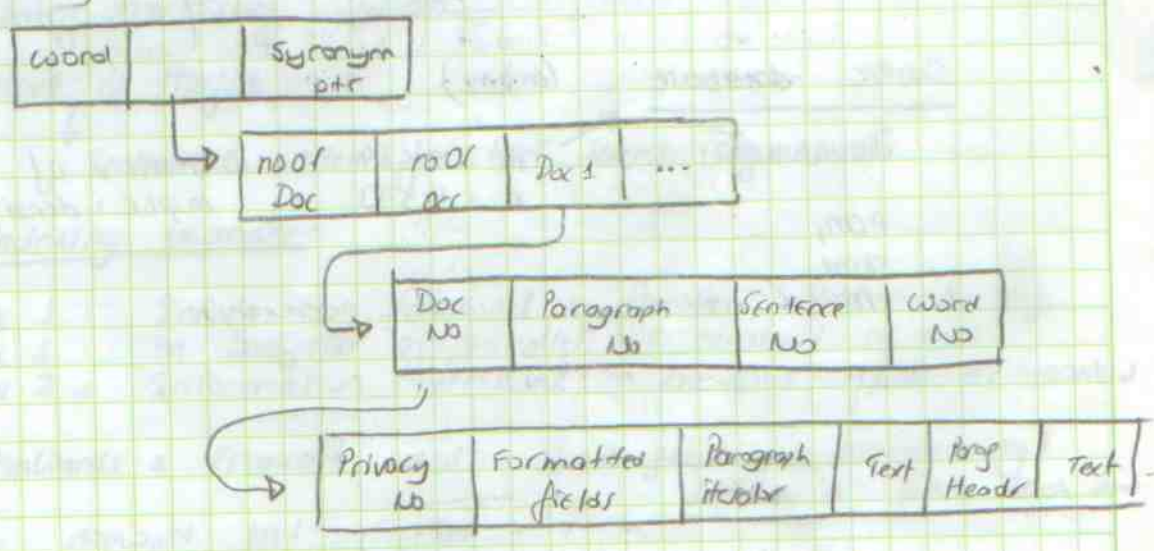
Example IR Systems:

STAIRS: Storage Information Retrieval Systems

Designed for mainframes, IBM

File System / Structure

Dictionary



### Formatted Field:

- Author Name
- Journal Name
- Page No

STAIRS has two programs:

1. utility programs : for database initialization & maintenance
2. Query & Retrieval Utility Systems (AQUARIUS)

### Modes of Operations:

Search Mode: for textual IR

Select Mode: for structural IR  
using formatted fields

### Queries:

HEART  
 HEART or DISEASE  
 HEART & and DISEASE & 3  
 WITH  
 SAME  
 in the same sentence  
 in the same paragraph

- Find matching documents using a boolean query
- Rank matching documents according to their significance

Value of a query term:  $f(a, b, c)$

a: the frequency of a word in the document

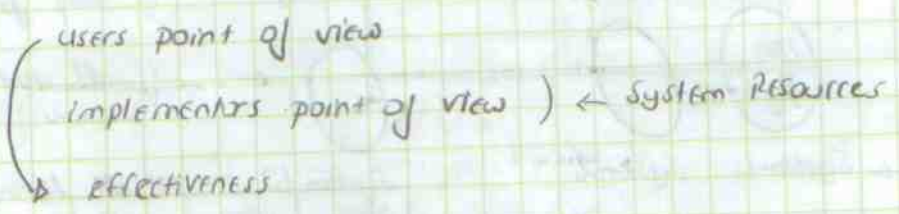
b: retrieved set

c: the no. of documents in the retrieved set in which the term occurs

$$\text{Query Term Value} = \frac{a \times b}{c}$$

Score of a document =  $\sum$  value of all query terms which appear in this document

### Evaluation of IR Systems:



efficiency: response time

system resource requirement x

- coverage

- links should be alive

- ease of use

# How to measure effectiveness?

TREC: Text Retrieval Conference [1992, ...]

waco.nist.gov/trec? (wiki)   
 ↳ national institute standards of technology

TREC 3  
Appendix A

trec-eval package

Precision (p):  $\frac{\# \text{ of retrieved \& relevant documents}}{\# \text{ of retrieved documents}}$



$\frac{2}{5} = 0.40$

p10  
p20

precision @ 10 or 20

first one // two pages

RECALL:  $\frac{\# \text{ of retrieved \& relevant documents}}{(\text{total } \# \text{ of relevant docs in the collection})}$

$\frac{2}{20}$  → total # of rel. doc

## TEST COLLECTION IN IR

A set of documents  
A set of queries & their relevant documents } \* 50

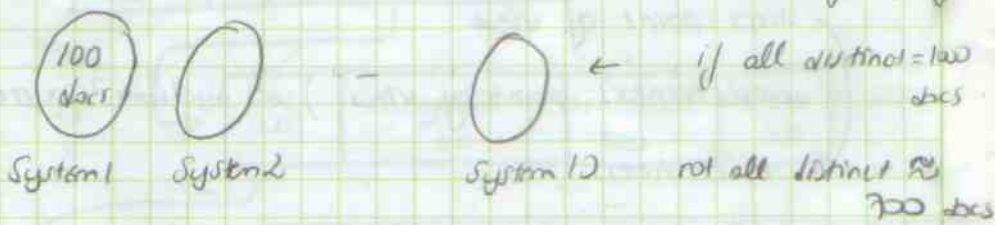
Standard test collections facilitate repetition of tests

in statistical sense 50 is (if you get 50 relevant docs)

Finding relevant documents for queries:

TREC uses the pooling concept

Retrieve top 100 documents & identify the relevant ones for query



all other docs they will be assumed as irrelevant

How reliable?

ACM SIGIR 1998  
Justin Zobel

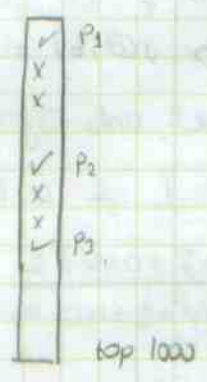
Documents which are not seen by the evaluators (annotators) are assumed as irrelevant. Actually some of them can be relevant.

A new measure (ACM SIGIR Conf. 2004, by Buckley & Voorhees)

bpref: binary preference

MAP: bpref is as reliable as MAP

↑ (mean Average Precision)



$$MAP = \frac{P_1 + P_2 + P_3 + 0}{4}$$

All together there are 4 relevant docs

Example for Recall & Precision

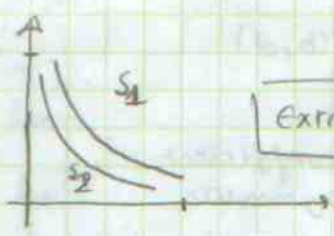
Rank	1	2	3	4	5	6	7	8	9	10
Relevance	0	1	0	1	1	1	1	0	0	0
Precision	0/1	1/2	1/3	2/4	3/5	4/6	5/7	5/8	5/9	5/10
Recall	0/10	1/10	1/10	2/10	3/10	4/10	5/10	5/10	5/10	5/10

Dugartlike  
Anno

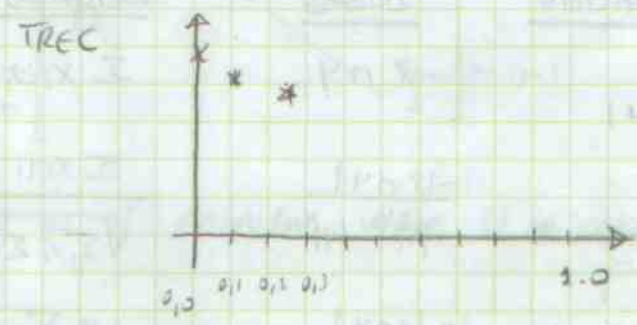
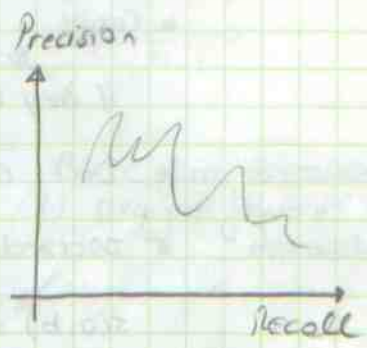
total no of relevant docs = 10

more relevant precision ↓

S1 is better than S2



Extrapolation to prevent those zigzags



makes the curve nicer...

11 point

t test → available

Is the difference statistically significant?

SNO	MAP <sub>1</sub>	MAP <sub>2</sub>
1		
2		
⋮		
50		

student t-test  
2-tail  
1-tail

not only looks at avg compares the individual values gives a "p" value

p = 0.05  
if p value is p < 0.05 significant

0.50 0.55

## Similarity Calculation:

Motivation: - For ranking documents according to their similarity to submitted query

browsing

- Cluster documents according to similarity to each other then use these clusters to find additional relevant documents like your favorite document

\* There are several similarity coefficients

Most of them is symmetric  $\Rightarrow s(a,b) = s(b,a)$

Van Rijsbergen, Information Retrieval, Univ. of Glasgow

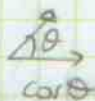
### Similarity Coefficient

Binary

Weighted

\* Dot product

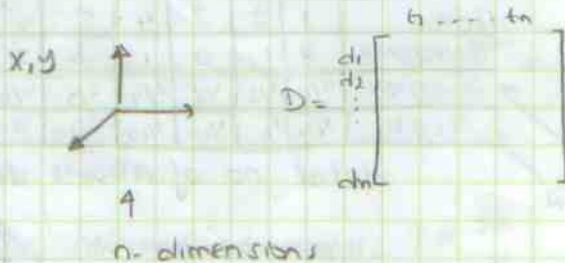
\* Cosine



if they overlap  $\Rightarrow \cos 0 = 1$

\* Dice Coeff

\* Jaccard



$$s(a,b) = s(b,a)$$

cover coefficient  
symmetric

### Similarity Coefficient

Binary

Weighted

Inner Product  
(Dot Product)

$X \cdot Y$

$\sum x_i y_i$

Cosine

$\frac{|X \cdot Y|}{|X|^{1/2} |Y|^{1/2}}$

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

Dice

$\frac{2 \cdot |X \cap Y|}{|X| + |Y|}$

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

Jaccard

$\frac{|X \cap Y|}{(|X| + |Y|) - |X \cap Y|}$

$\frac{\sum x_i y_i}{\sum x_i^2 + \sum y_i^2 - \sum x_i y_i}$

$$\begin{array}{l}
 X = (1 \ 0 \ 1 \ 1 \ 1) \quad |X| = 4 \\
 Y = (1 \ 1 \ 0 \ 1 \ 0) \quad |Y| = 3
 \end{array}
 \left. \begin{array}{l}
 \text{Inner Product} = 2 \\
 \text{Dice} = \frac{2 \cdot 2}{4+3} = \frac{4}{7} \\
 \text{Jaccard} = \frac{2}{4+3-2} = \frac{2}{5} \\
 \text{Cosine} = \frac{2}{\sqrt{4 \cdot 3}}
 \end{array} \right\}$$

$\Rightarrow$  tf idf (we assign higher values to words which appear more frequently in documents  $\Rightarrow$  like stopwords)  
 $\downarrow$   
 inverse document frequency  
 Salton Buckley Term Weighting Approaches Information processing and management. P&I

$$\begin{array}{l}
 X = (2 \ 0 \ 1 \ 3 \ 2) \\
 Y = (1 \ 0 \ 2 \ 1 \ 5) \\
 \text{Dice} = \frac{2(2+0+2+3+10)}{(4+0+1+9+4) + (1+0+4+1+25)} = 0.69 \\
 \text{Cosine} = \frac{17}{(18 \cdot 31)^{1/2}} = \frac{17}{23.6} \approx 0.72
 \end{array}$$

Clusty

How To CALCULATE SIMILARITY AMONG DOCUMENTS

$$D = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{bmatrix} \begin{matrix} t_1 & t_2 & \dots & t_n \end{matrix}$$

$S_{ij} = S_{ji}$

Brute force approach  
1) straight forward approach

$$S = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \begin{matrix} 1 & 2 & 3 & \dots & m \\ S_{12} & S_{13} & \dots & S_{1m} \\ S_{23} & \dots & S_{2m} \\ S_{34} & \dots & S_{3m} \\ \dots & \dots & \dots & \dots & \dots \\ S_{m-1} & \dots & \dots & \dots & S_{m-1} \\ 1 \end{matrix} \leftarrow m-1$$

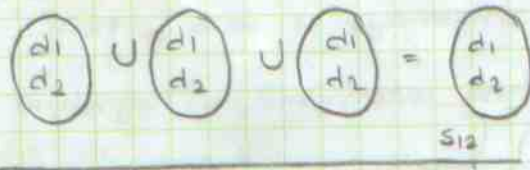
Total no of similar value to be calculated =  $1+2+\dots+(m-1) = \frac{m(m-1)}{2}$

2) Using the knowledge of term distributions in documents.

	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	
$d_1$	1	1	0	0	1	0	$t_1 \rightarrow d_1, d_2$
$d_2$	1	1	0	1	1	0	$t_4 \rightarrow d_2, d_5$
$d_3$	0	0	0	0	0	1	$t_2 \rightarrow d_1, d_2$
$d_4$	0	0	1	0	0	1	$t_5 \rightarrow d_1, d_2$
$d_5$	0	0	1	1	0	1	$t_3 \rightarrow d_4, d_5$
							$t_6 \rightarrow d_3, d_4, d_5$

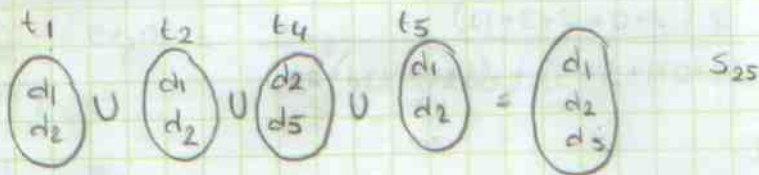
Consider  $d_1$ :

$t_1, t_2, t_5$

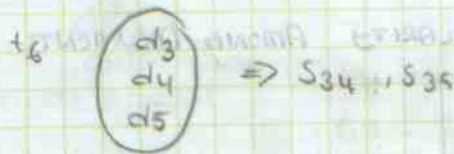


1	$S_{12}$	$S_{13}$	$S_{14}$	$S_{15}$
	1	$S_{23}$	$S_{24}$	$S_{25}$
		1	$S_{34}$	$S_{35}$
			1	$S_{45}$
				1

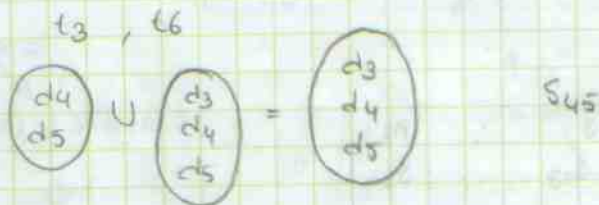
$d_2$ :



$d_3$ :



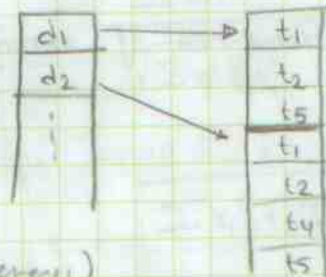
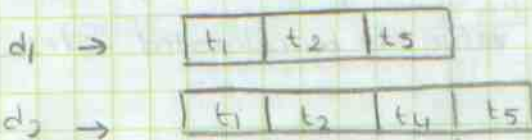
$d_4$ :



There are 10 similarity values to be calculated, but we need to calculate only 5 of them.

$5/10 \Rightarrow 50\%$  savings

3) Using the Inverted File for Term



use pointers to find the similarity  $\rightarrow$  (See common elements)

In order to get rid of inverted pointer

$t_1$	$t_1$	or they same	✓	increment counter
$t_2$	$t_2$	"	✓	"
$t_5$	$t_4$	"	—	increment pointer
$t_5$	$t_5$	"	✓	" counter

	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$
$d_1$	1	1	0	0	1	0
$d_2$	1	1	0	1	1	0
$d_3$	0	0	0	0	0	1
$d_4$	0	0	1	0	0	1
$d_5$	0	0	1	1	0	1

document  $\rightarrow$  freq

$t_1 \rightarrow \langle 1, 1 \rangle \langle 2, 1 \rangle$        $t_4 \rightarrow \langle 2, 1 \rangle \langle 5, 1 \rangle$   
 $t_2 \rightarrow \langle 1, 1 \rangle \langle 2, 1 \rangle$        $t_5 \rightarrow \langle 4, 1 \rangle \langle 2, 1 \rangle$   
 $t_3 \rightarrow \langle 4, 1 \rangle \langle 5, 1 \rangle$        $t_6 \rightarrow \langle 3, 1 \rangle \langle 4, 1 \rangle \langle 5, 1 \rangle$

Dice Coef:  $\frac{2|X \cap Y|}{|X| + |Y|}$

document  $\rightarrow$  length array

$d_1$	$d_2$	$d_3$	$d_4$	$d_5$
3	4	1	2	3

$$S = \begin{bmatrix} 1 & S_{12} & S_{13} & S_{14} & S_{15} \\ & 1 & S_{23} & S_{24} & S_{25} \\ & & & & & \\ & & & & & \\ & & & & & \end{bmatrix}$$

Consider  $d_1$ :

Similarity Array:

$S_{11}$	$S_{12}$	$S_{13}$	$S_{14}$	$S_{15}$
X	0	0	0	0

mail boxes

$$\frac{2 \times 3}{3+4} = \frac{6}{7}$$

$t_1, t_2, t_5$

Consider  $d_2$ :

$S_{21}$	$S_{22}$	$S_{23}$	$S_{24}$	$S_{25}$
X	X	0	0	0

$$\frac{2 \times 1}{3+4} = \frac{2}{7}$$

$t_1, t_2, t_4, t_5$

$\begin{matrix} 1,1 & 1,1 & 2,1 & 1,1 \\ 2,1 & 2,1 & 5,1 & 2,1 \end{matrix}$ 
 ignore  $(1,1)$ 's      increment others

Consider the Computation Requirements:

- $m$
  - $X_d$ : depth of indexing (no of avg terms / doc)
  - $t_g$ : term generality (avg posting list length, avg no of docs / term)
- $O(m \times X_d \times t_g)$

$t_4 \rightarrow \langle 2, 1 \rangle \langle 5, 1 \rangle \leftrightarrow \langle 5, 1 \rangle \langle 2, 1 \rangle$

try to reduce the comparison looking at this order. In postable keep X



\* Doing the calculations faster: organize the posting lists in reverse order: first higher numbered documents

$$x_d \text{tg } \frac{m-1}{m} + x_d \text{tg } \frac{m-2}{m} + \dots$$

$$x_d \text{tg } \frac{1}{m} \left( \frac{m(m-1)}{2} \right) = x_d \text{tg } \frac{m}{2}$$

CLUSTERING:

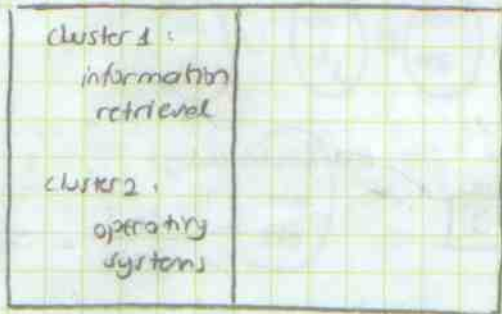
Jain, Murty, Flynn Data Clustering  
ACM Computing Surveys, Sept 1995

Clusby  
Sophia

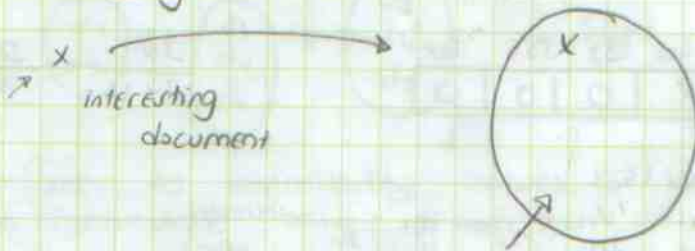
Read (link on the course web page)

How to use clusters for IR:

1- For clustering the search results



2- For browsing:



look at the other members of this document's cluster.

3- Cluster-based retrieval

Sutton & Sinden

CT Yu

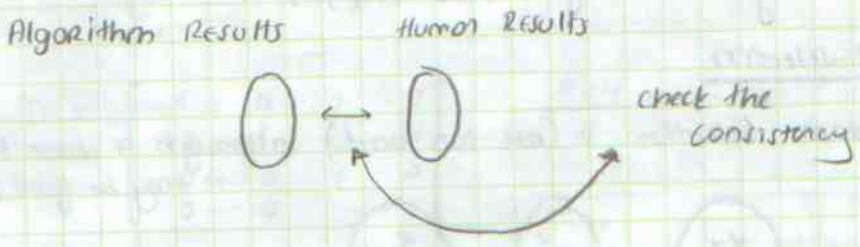
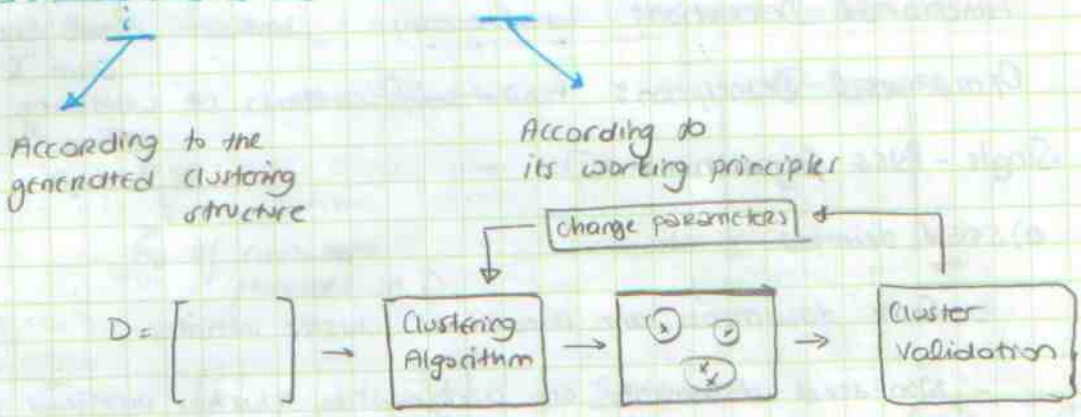
- First choose the best matching clusters
- Then match your query with the documents of these selected clusters.

Clusters → cluster centroids  
Documents → document vectors

# Classification of Clustering Algorithms

**Categorization:** Groups are defined before categorization ← supervised classification

**Clustering:** Groups are undefined when we begin ← unsupervised



- partitioning :  $C_i \cap C_j = \emptyset, i \neq j$
- overlapping :  $C_i \cap C_j \neq \emptyset, i \neq j$   
can be
- hierarchical :
- single pass :
- multi-pass :
- graph-theoretical
- based on user queries  
omickenski : ACM TODS, 1990

Peter J. Denning ⇒ (working set model 'i sikaran amca)

IT Profession diye bir column 'i var  
Mastering the mass  
Abstraction ← kosmalar ufak ufak (ingiliz ama yagmris)  
Flow ⇒ sevdimiz bir isi yaparken naril de akip gecir yonun  
Mihaly Csikszentmihalyi lay Bg!

Communication  
of  
ACM April  
2007

\* Cluster hypothesis (Van Rijsbergen)

Functional Description: what ← easy

Operational Description: "how ← difficult

Paul Erdős  
Gail & Dubin  
Clustering Algorithm  
1988

1 \* Single-Pass Algorithms

a) seed oriented

- Some documents are selected as cluster initiators
- Non-seed documents are assigned to clusters initiated by seeds

! how many number of clusters ( $n_c$ )?

\* SEED-selection

- Random Selection (not too bad) although it doesn't seem so it may be good for some applications



- Choose first  $n_c$
- Generate  $n_c$  synthetic seed documents
- Use pasting list (Peter Willet)

take the first make a cluster  
 get the second if similar  
 put it in the first one's cluster  
 or it creates its own cluster  
 take third is it similar to any  
 of them -- and so on

Anderberg : 1973  
 Jain & Dubes : 1988  
 Jain, Flynn, Murthi : 1999

Single-Pass Algorithms : (continued)

a) Seed oriented

How to choose seed.

$$n_c = \frac{m \times n}{t}$$

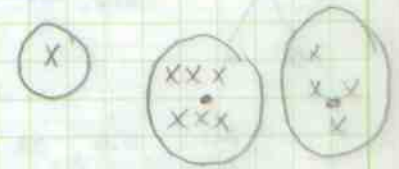
↑  
no of non-zero elements in D.

$$D = \begin{bmatrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \end{bmatrix}$$

dm

$$D = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\frac{3 \times 4}{12} = 1$$



$$D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\frac{4 \times 4}{4} = 4$$

b) Heuristic Approach :

Von Rijsbergen

used in new event detection & tracking

temporal order

- 1- Process documents (objects) serially (one by one)
- 2- The first document becomes a cluster by itself.
- 3- Consider the next document

if it is not similar to existing clusters then it starts its own cluster

else

joins to the most similar cluster (s)

Questions:

- which similarity measure
- similarity threshold
- order of processing

↑  
as we change the cluster also change their centroids



How to represent clusters (cluster centroids)?

\* Multi-pass Algorithms

A typical approach

use a seed oriented approach  
" the heuristic approach.

1- Obtain the initial clusters using an efficient algorithm.

2- repeat

- Generate cluster centroids
- Reassign objects to the clusters according to their similarity to the cluster centroids

until all documents stay in their previous cluster OR

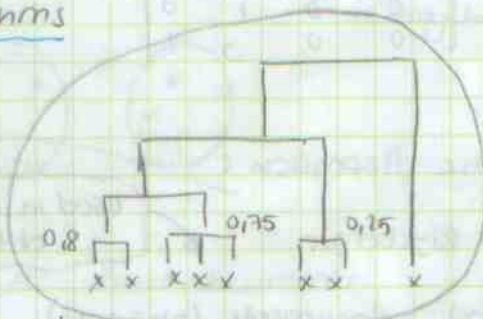
100% stability  
90% "

no of iterations =  $\frac{\text{limit}}{10 \text{ iterations?}}$

Graph Theoretical Algorithms

Agglomerative

\* bottom-up



this cluster is referred as  $\alpha \Rightarrow$  Dendrogram

Individual similarities are used as a starting point, and a gluing process collects similar items, or group, into larger groups.

- Single-link
- Complete-link
- Average-link

SPSS  
SAS  
MATLAB?

Ellen Voorhees  
IASS  
Sallon  
Cornell



Single-link: the similarity between a pair of clusters is taken to be the similarity between the most similar pair of documents, one of which appears in each cluster; thus each cluster member will be more similar to at least one member in that same cluster than to any member of another cluster

04/03/2008

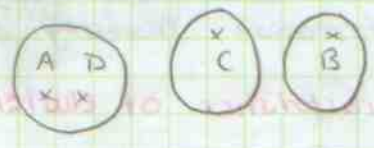
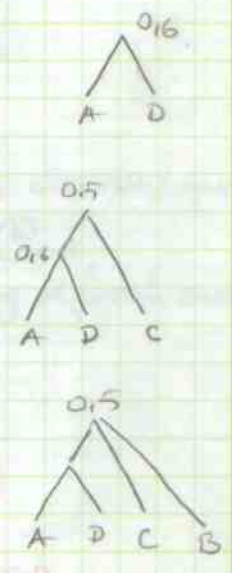
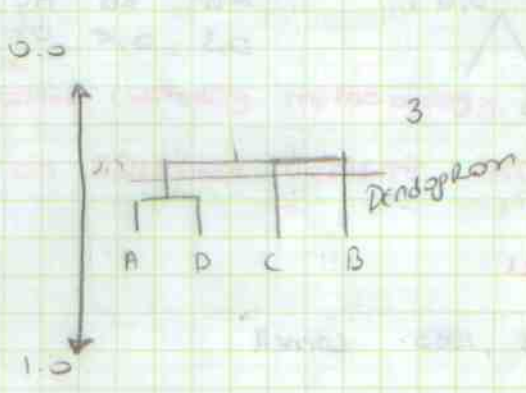
$$S = \begin{matrix} & A & B & C & D \\ \begin{matrix} A \\ B \\ C \\ D \end{matrix} & \begin{bmatrix} 1.0 & 0.3 & 0.5 & 0.6 \\ - & 1.0 & 0.4 & 0.5 \\ - & - & 1.0 & 0.3 \\ - & - & - & 1.0 \end{bmatrix} \end{matrix}$$

CLUSTER.EXE

C++  
OOL (Object Windows Library)

STEP	PAIR	SIMILARITY VALUE
1	AD	0.6
2	AC	0.5
3	BD	0.5
4	BC	0.4
5	AB	0.3
6	CD	0.3

STEP	Sim Pair
1	AD, 0.6
2	AC, 0.5
3	BD, 0.5



$$A \begin{bmatrix} A & B & C & D \\ 1.0 & 0.5 & 0.5 & 0.6 \\ - & 1.0 & 0.5 & 0.5 \\ - & - & 1.0 & 0.5 \\ - & - & - & 1.0 \end{bmatrix}$$

Similarity matrix implied by the dendrogram.

$S_{ij}$  vs  $S_i$  Product moment correlation

$[-1, +1]$

total agreement / total disagreement





If the clustering structure reflects the nature of the original data then there will be a high agreement of the values of  $S$  &  $S_i$  matrices

0.8  $\rightarrow$  good no for agreement

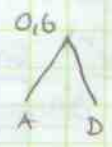
Join + ?

Advances in Computers 1985

### Complete Algorithm:

the sim between C-D not known thus we cannot join this one

① AD 0.6

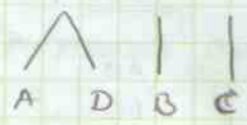


② AC, 0.5

AC, CD

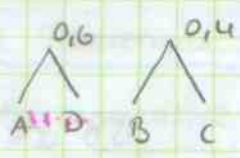
③ BD 0.5

AB: ?



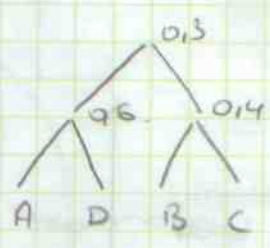
↑ singletons the cluster with one member

④ BC 0.4



⑤ AB: 0.3

AB	BD	AC	CD
0.3	0.5	0.5	0.3



### Average Link:

Ellen Voorhees, 1965 Cornell

Survey: Peter Willet on hierarchical clustering Information Processing & Management 1988

### Desirable Characteristics of Clustering Algorithms:

**EFFECTIVE:** generates a meaningful clustering structure + provides an effective IR environment

**EFFICIENT:** Time & Space

