Consider the following documents **D1-D4** using 8 different terms:

D1 = {breakthrough drug schizophrenia}

D2 = {new schizophrenia drug}

D3 = {new approach treatment schizophrenia}

D4 = {new hope schizophrenia patient}

Task II.

approach -> 3
breakthrough -> 1
drug -> 1 -> 2
hope -> 4
new -> 2 -> 3 -> 4
patient -> 4
schizophrenia -> 1 -> 2 -> 3 -> 4
treatment -> 3

I) Draw the term-document incidence matrix for this document collection.

II) Draw the inverted index for this document collected using a linked list representation.

III) What are the returned results for the below queries (show bitwise operations):

	D1	D2	D3	D4
approach	0	0	1	0
breakthrough	1	0	0	0
drug	1	1	0	0
hope	0	0	0	1
new	0	1	1	1
patient	0	0	0	1
schizophrenia	1	1	1	1
treatment	0	0	1	0

q_1) schizophrenia AND drug

(intersection of these terms in linked list from Task II.) 1 -> 2

(from Task III. same but in binary form) 1111 AND 1100 = 1100

q_2) new AND NOT(drug OR approach)

0111 AND NOT(1100 OR 0010)

0111 AND NOT(1110)

…

0111 AND NOT(1100 OR 0010) = 0111 AND NOT(1110) = 0111 AND 0001 = 0001

Slides 11 and 13

Given the following postings list sizes, recommend a query processing order for:

Term	Postings size
eyes	120
kaleidoscope	80
marmalade	100
skies	300
tangerine	50
trees	500

I) eyes AND kaleidoscope AND tangerine (120, 80, 50)
1. kaleidoscope AND tangerine;
2. eyes AND 1.

II) marmalade OR tries OR skies (100, 500, 300)
1. marmelade OR skies;
2. trees OR 1.

Firstly, merge the ones with shorter posting size

III) (tangerine OR trees) AND (marmalade OR skies) AND (kaleidoscope OR eyes)
(50, 500) (100, 300) (80, 120)

50 + 500 = 550

100 + 300 = 400

80 + 120 = 200

1. (kaleidoscope OR eyes);
2. (marmelade OR skies);
3. (tangerine OR trees)

Note: if Task III. would contain "NOT": NOT is just the number of documents minus the length

1. Sort users by their IP:

User 1:
I. 1.1.1.1 [30:00:22:38] A
IV. 1.1.1.1 [30:00:41:55] D
VI. 1.1.1.1 [30:01:15:47] F

User 2:
II. 1.1.1.2 [30:00:29:47] B
III. 1.1.1.2 [30:00:41:47] C
V. 1.1.1.2 [30:01:00:02] E
VII. 1.1.1.2 [30:01:22:38] G

Identified Users

h1:
For User 1: 3 sessions (time spent on each URL (I., IV., VI.) exceeds 10 min threshold.
For User 2: 4 sessions (same situation as for User 1)

h2:
For User 1: First session: I. and II.; Second session: VI.
For User 2: One session: II., III., V., VII.

href: Cannot be defined on this example task, since reference URL needed (See Slide 40).

Given the following log file:

HH MM SS

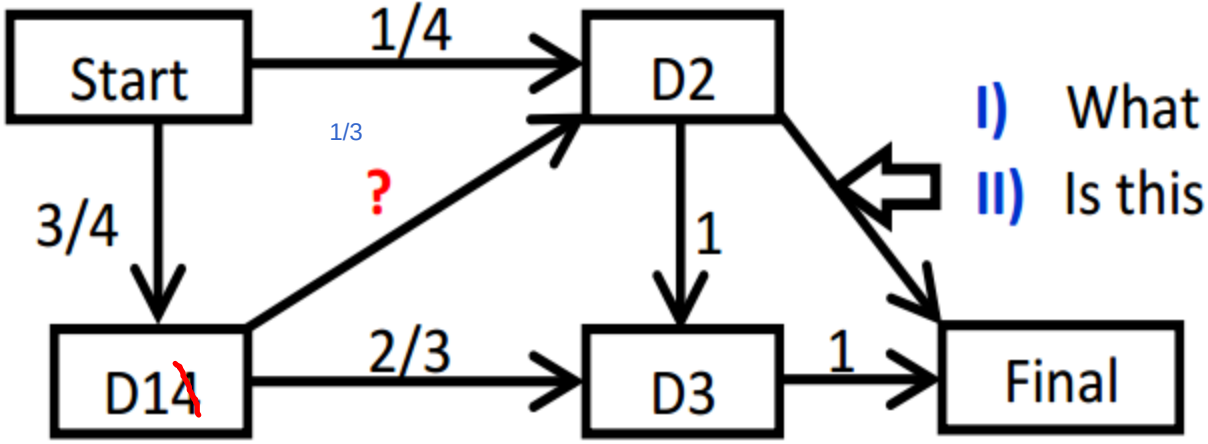
I. 1.1.1.1 [30:00:22:38] "GET /A.html HTTP/1.0" 200 156
II. 1.1.1.2 [30:00:29:47] "GET /B.html HTTP/1.0" 200 1788
III. 1.1.1.2 [30:00:41:47] "GET /C.htm HTTP/1.0" 200 1788
IV. 1.1.1.1 [30:00:41:55] "GET /D.html HTTP/1.0" 200 457
V. 1.1.1.2 [30:01:00:02] "GET /E.html HTTP/1.0" 200 1588
VI. 1.1.1.1 [30:01:15:47] "GET /F.html HTTP/1.0" 200 1788
VII. 1.1.1.2 [30:01:22:38] "GET /G.html HTTP/1.0" 200 1588

I) Identify the users using IP address and their sessions using H1 with timeout=10min or H2 with timeout=30min?

Note:
h1: Total session duration may not exceed a threshold;
h2: Total time spent on a page may not exceed a threshold;
href (referrer-based): Given two consecutive requests p and q, q is assigned to S, if the referrer for q was previously invoked in S

Slides 32, 33, 37, 39, 40

Given the following sessions: {D1 D2 D3}, {D1 D3}, {D1 D3}, {D2 D3}, **draw the Markov chain**.



I) What is P(D1→D2)?

II) Is this arrow making any sense?

III) What is the probability of starting/terminating a session in D2? 1 - 2/3 = 1/3

Note: Total 1.0 can go from one node.

IV) What is the probability of P(Start→D1→D3)? 3/4 * 2/3 = 1/2

V) What is the probability of P(D3|D1)? 2/3 + 1/3 * 2 = 1

Note: Sum of all paths from D1 to D3.

Slides 45, 46, 48

I) Consider the following documents: **D1** = $\{t_1\ t_2\ t_1\ t_2\ t_3\}$ and **D2** = $\{t_4\ t_2\ t_3\ t_3\}$. What is the Jaccard coefficient for D1 and D2?

$$Jac(D1,D2) = \frac{\text{intersection}}{\text{union}} = \frac{\{ t2, t3 \}}{\{ t1, t2, t3, t4 \}} = \frac{2}{4}$$

II) We know that **D3** contains 7 unique terms and **D4** contains 8 unique terms. Five terms are joint for **D3** and **D4**. What is the Jaccard coefficient for D3 and D4?

$$Jac(D3,D4) = \frac{7 + 8}{5} = \frac{15}{5}$$

Consider a set of terms **T** = {t_1, t_2, t_3, t_4} and the following collection of two documents:
D1 = {t_1 t_2 t_1 t_2 t_3} and **D2** = {t_4 t_2 t_2 t_3}. Consider query **Q** = {t_1 t_4}.

I) Represent D1, D2, and Q using TF (normalized Bag-Of-Words).

TF	t_1	t_2	t_3	t_4	Max
D1	2/2	2/2	1/2	0	2
D2	0/2	2/2	1/2	1/2	2
Q	1/1	0/1	0/1	1/1	1

normalized BOW (divided through the maximal number of occurrences of any term)

TF of a term (Matrix from Task I.) * IDF (Matrix from Task II.)

TF-IDF	t_1	t_2	t_3	t_4
D1	log2	0	0	log2 * 0 = 0
D2	0*log2 = 0	1*log1 = 0	1/2 * 0 = 0	1/2 * log2
Q	1*log2	0	0	1*log2

II) Compute IDFs for all four terms (note that only D1 and D2 are included in the collection).

III) Represent D1, D2, and Q using TF-IDF.

	t_1	t_2	t_3	t_4
IDF	log2	log1=0	log1=0	log(2/1) = log2

IDF of a term: idf = log (N / df), where :
- N - number of documents in the collection;
- df - number of documents containing this term.

IV) Compute the lengths of D1, D2, and Q vectors in the TF-IDF representation.

$|D1| = \sqrt{(\log2)^2 + 0^2 + 0^2 + 0^2} = \log2$ $\quad |D2| = \sqrt{0 + 0 + 0 + (1/2 * \log2)^2} = 1/2 * \log2$ $\quad |Q| = \sqrt{(\log2)^2 + 0 + 0 + (\log2)^2} = \sqrt{2} * \log2$

V) Compute the cosine similarities for D1 and D2 when compared with query Q? Show the ranking derived with our retrieval system.

$$sim(Q,D1) = \frac{Q * D1}{|Q| * |D1|} = \frac{\log2 * \log2 + 0 * 0 + 0 * 0 + \log2 * 0}{\sqrt{2} * \log2 * \log2} = \frac{1}{\sqrt{2}}$$

$$sim(Q,D2) = \frac{\log2 * 0 + 0 + 0 + 1/2 * \log2 * \log2}{2 * \log2 * 1/2 * \log2}$$

Ranking:

Slides 63, 72-75

$C =$

	D1	D2	D3	D4	D5
t_1	5	5	0	0	1
t_2	4	5	1	1	0
t_3	5	4	1	1	0
t_4	0	0	4	4	4
t_5	0	0	5	5	5
t_6	1	1	4	4	4

Consider the term-document matrix **C** for the bag-of-words representation of five documents **D1-D5** in the space of six terms t_1-t_6. Using the SVD factorization method, matrix **C** has been decomposed into matrices **K**, **S**, and $\mathbf{D}^T$ given below. The rank of **C** is 4 ($4 \leq \min\{6,5\}$), so 4 concepts (semantic dimensions) were discovered.

Slide 88

I) Where to find the importance (informativeness) of these concepts? 13.74 (S)
Based on these values, which seems the most obvious value for the reduced number of dimensions k? 2 (since their importance is too low)

$K =$ terms -> concepts

t1	-0.27	0.55	-0.78	0
t2	-0.29	0.47	0.44	-0.71
t3	-0.29	0.47	0.44	0.71
t4	-0.45	-0.29	-0.01	0
t5	-0.56	-0.36	-0.02	0
t6	-0.50	-0.18	-0.05	0

$S =$ concept space

13.74	0	0	0
0	10.88	0	0
0	0	1.36	0
0	0	0	1

Measures of the importance of the corresponding semantic dimension

$D^T =$ docs -> concepts

D1	D2	D3	D4	D5
-0.32	-0.32	-0.52	-0.52	-0.5
0.63	0.63	-0.25	-0.25	-0.29
-0.02	-0.02	0.41	0.41	-0.82
0.71	-0.71	0	0	0

How strongly is document related to the concept represented by semantic dimension

II) Where to find the mapping of all terms to the most important (informative) concept?
What about the least important concept?

III) Where to find the representation of documents D1 and D3 in the space of concepts?

Slides 85 - 89

$C =$

	D1	D2	D3	D4	D5
t_1	5	5	0	0	1
t_2	4	5	1	1	0
t_3	5	4	1	1	0
t_4	0	0	4	4	4
t_5	0	0	5	5	5
t_6	1	1	4	4	4

Assume that for the problem presented in the previous slide, we would leave only k=2 most important concepts. Matrices $\mathbf{K_k}$ and $\mathbf{S_k}$ are presented below.

I) Present $\mathbf{D_k}^T$ (i.e., the representation of documents in the 2-dimensional space of two most important concepts).

II) Having multiplied $\mathbf{K_k}$, $\mathbf{S_k}$, and $\mathbf{D_k}^T$, we obtain matrix $\mathbf{C_k}$. Compare it with $\mathbf{C}$ and show how to compute the Frobenius norm?

$K_k =$

-0.27	0.55
-0.29	0.47
-0.29	0.47
-0.45	-0.29
-0.56	-0.36
-0.50	-0.18

$S_k =$

13.74	0
0	10.88

$D_k^T =$

D1	D2	D3	D4	D5
-0.32	-0.32	-0.52	-0.52	-0.5
0.63	0.63	-0.25	-0.25	-0.29

Note: It is Dt from the previous slide without two last rows (since we cut off two last concepts)

$C_k =$

	D1	D2	D3	D4	D5
t_1	4.95	4.95	0.43	0.43	0.11
t_2	4.49	4.49	0.79	0.79	0.50
t_3	4.49	4.49	0.79	0.79	0.50
t_4	0	0	4	4	4
t_5	0	0	4.98	4.98	4.98
t_6	0.96	0.96	4.06	4.06	4

Task II.:

$$\|X\|_F = \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} X_{ij}^2} = \sqrt{(5-4.95)^2 + (5-4.95)^2 + (0-0.43)^2 + \ldots + (4-4.06)^2 + (4-4)^2}$$

An IR system returns 8 relevant documents and 10 non-relevant documents. There are a total of 20 relevant documents in the collection.

I) What is the precision of the system on this search? $P = \frac{8}{10+8} = \frac{8}{18}$

II) What is its recall? $R = \frac{8}{20}$

III) What is its F measure with α=1/2? $F = \frac{1}{(1/2) * (18 / 8) + (1/2) * (20/8)} = \frac{1}{19/8} = \frac{8}{19}$

$$\text{Precision} = \frac{\text{relevant retrieved items}}{\text{all retrieved items}}$$

$$\text{Recall} = \frac{\text{relevant retrieved items}}{\text{all relevant items}}$$

$$F = \frac{1}{\text{alpha} * (1 / \text{Precision}) + (1 - \text{alpha}) * (1 / \text{Recall})}$$

Slides 102, 103, 105

P@k = (1 / k) * number of relevant documents till k

R@k = (1 / number of total relevant documents) * number of relevant documents till k

MAP = (1 / number of total relevant documents) * Precision of each relevant document with k = position of a relevant document

R-precision = P@k, where k - number of total relevant documents

Consider an information need for which there are 4 relevant documents in the collection. Contrast two systems run on this collection. Their top 10 results are judged for relevance as follows:

System 1: R N R N N N | N N R R

System 2: N R N N R R | R N N N

I) What is the precision of each system at 6 (P@6)?

II) What is the recall of each system at 6 (R@6)?

III) What is MAP for each system? Which has a higher MAP?

IV) What is the R-precision of each system? Does it rank the systems the same as MAP?

System 1:
P@6 = (1/6) * (1 + 0 + 1 + 0 + 0 + 0) = 2/6.
System 2:
P@6 = (1/6) * (0 + 1 + 0 + 0 + 1 + 1) = 3/6.

System 1:
R@6 = (1/4) * (1 + 0 + 1 + 0 + 0 + 0) = 2/4
System 2:
R@6 = (1/4) * (0 + 1 + 0 + 0 + 1 + 1) = 3/4

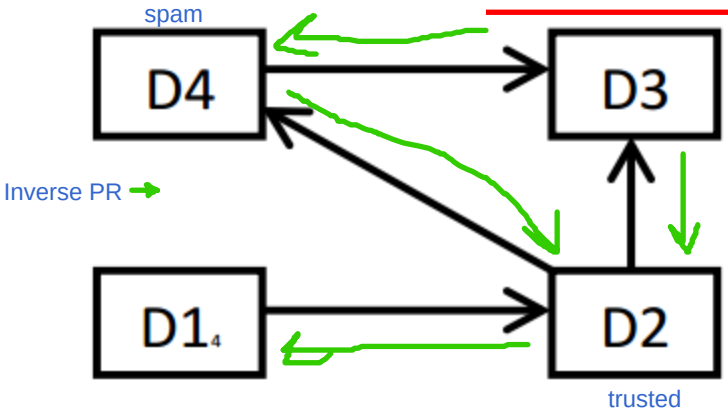
Task III.:
System 1:
MAP = (1/4) * (1/1 + 0 + 2/3 + 0 + 0 + 0) = 5/12
System 2:
MAP = (1/4) * (0 + 1/2 + 0 + 0 + 2/5 + 3/6) = (1/4) * (7/5) = 7/20

=> MAP of System 1 is higher

Task IV.:
System 1:
R-precision = P@4 = (1/4) * (1 + 0 + 1 + 0) = 2/4
System 2:
R-precision = P@4 = (1/4) * (0 + 1 + 0 + 0) = 1/4

Slides 106-108

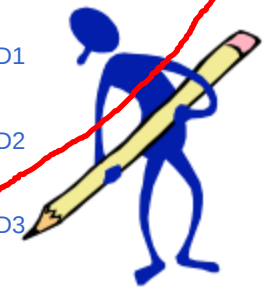
I) Consider the web graph presented below and fill in the stochastic matrix M.



$M =$

D1	D2	D3	D4	
0	? 0	0	0	D1
1	? 0	0	0	D2
0	? 1/2	0	1	D3
0	? 1/2	0	0	D4

from row D3



2 is number of outgoing links from D2

II) Write the equation for $PR(D3) = ?$ (with or without q)

PR(D3) = 0*PR(D1) + (1/2)*PR(D2) + 0*PR(D3) + 1*PR(D4)

III) Which page has the greatest/least PageRank? Justify why without computing PageRanks.

Since it's PR covers PRs of the 1/2 of D2 ang 1 of D4, which is the biggest out all the rows, it's the greatest. Only 1/2 from D2 for D4.

IV) Write the equation for $Inverse\ PR(D2) = ?$

Inverse PR(D2) = 0 * InvPR(D1) + 0 * InvPR(D2) + (1/2) * InvPR(D3) + 1 * InvPR(D4).

V) Which page has the greatest/least Inverse PageRank? Justify why.

VI) An oracle has evaluated D2 as trusted and D4 as spam? What is the starting vector d for TrustRank?

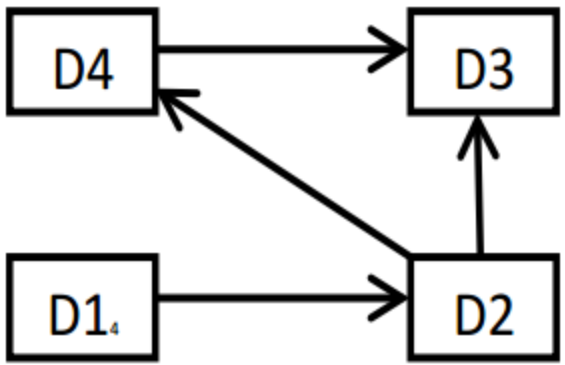
$d =$ | 0 | 1 | 0 | 0 |

Since D2 is trusted => add to vector d

(by constructing matrix for green arrows)

Slides 118-135

Consider the below web composed of 4 pages.



0	0	0	0
1	0	0	0
0	1	0	1
0	1	0	0

$= L^T$

LLt - for hubs

LtL - for authorities

The principal eigenvector of LL^T is [0, 1.618, 0, 1]

The principal eigenvector of L^TL is [0, 0, 1.618, 1]

D2 goes to D3, D4

I)

$L =$

0	1	0	0
?0	?0	?1	?1
0	0	0	0
0	0	1	0

D4

1	0	0	0
0	2	0	1
0	0	0	0
0	1	0	?2

$= L \cdot L^T$

D4 III)

sum of row D4 of L (0 + 0 + 1 + 0) and column D4 of Lt (0 + 0 + 1 + 0)

II) $a(D_4) = ?_1$

$h(D_4) = ?_1$

from eigenvectors

IV) Which page has the greatest authority score? Justify. D3

V) Which page has the greatest hub score? Justify. D2

Q*alpha + relevant documents * beta + non-relevant documents * omega

Consider a relevance feedback web search system, where the relevance feedback is based only on words in the title text returned for a page (for efficiency).

the amount of words in document/query

The user queries for: ***Q*** = banana slug

and the top three titles returned are:

D1 = banana slug columbianus

D2 = Santa Cruz mountains banana slug

D3 = Santa Cruz Mascot

corpus

	Q	D1	D2	D3	Q'
banana	1	1	1	0	1*1 + 1*1 + 1*1 + 0*1 = 3
columbianus	0	1	0	0	1
Cruz	0	0	1	1	2
Mascot	0	0	0	1	1
mountains	0	0	1	0	1
Santa	0	0	1	1	2
slug	1	1	1	0	3

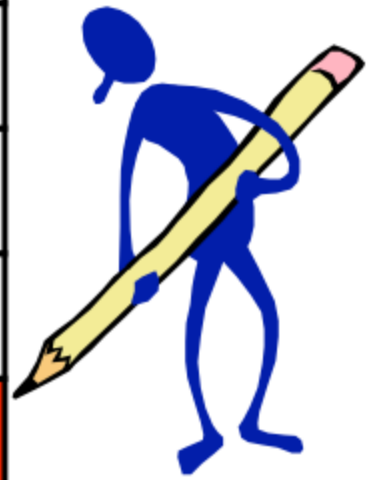
The user judges the first two documents relevant, and the third nonrelevant. Assume we use Bag-of-words representation and the Rocchio relevance feedback mechanism, with $\alpha = \beta = \gamma = 1$. Show the final revised query Q' that would be run (please list the vector elements in alphabetical order). Remember that in this mechanism, we consider the centroids of relevant and non-relevant documents, and the negative values are trimmed to 0.

alpha - query weight
beta - relevant documents weight
omega - non-relevant documents weight

Slides 163-164

Compute the Levenshtein distance for ”LEGIA” and ”LECHIA”

		L	E	C	H	I	A
	0	1	2	3	4	5	6
L	1	0	1	2	3	4	5
E	2	1	0	1	2	3	4
G	3	2	1	1	2	3	4
I	4	3	2	2	2	2	3
A	5	4	3	3	3	3	2



EXPLAINS
EVERYTHING

Given the following user-item rating matrix, predict rating of user U7 for item I4:

	I1	I2	I3	I4
U1	5	4	4	4
U2	5	3	7	3
U3	4	3	2	3
U4	6	4	5	4
U5	3	4	2	4
U6	4	3	5	3
U7	4	3	5	?

sim(U7,U·)
0.0
1.0
-0.5
0.5
-1.0
1.0

Task I.:

User-based CF with k=2 and simple average: (3s are selected, since the appear the most among k users of the greatest similarity)

U7(I4) = (3+3)/2 = ?

… or weighted average:

U7(I4) = (1·3 + 1·3)/(1+1) = ?

the greatest similarities

k - number of users with greatest similarity to consider

I) Employ user-based CF with k=2 and either simple average or weighted average?

U7(I4) = (3 + 3 + 4) / 3 = 3.3(3) (with k=3, we take user of next greatest similarity (0.5, in our case))

No, since, selecting next user, user has similarity of 0.0.

II) How the results change when using k=3? Does it make sense to account for k=4?

III) Employ user-based CF with k=2 and modify U7's average rating by the weighted modification of its nearest neigbors averages:

U7(I4) = 4 + (1·(3-5) + 1·(…))/(1+1) = ?

Slide 5 from IR-Lecture-Short-Exercises

IV) Which item should be analyzed to predict the rating when using item-based CF with k=1? What would be the predicted rating?

I2 with prediction of 3 (the greatest column similarity

Four advertisers A, B, C, and D with a daily budget of $2 bid for the following keywords ($1 each): A: w, x; B: x, z; C: x, y; D: y, z.

I) Use a simplified version of BALANCE to select the ads for the following query stream (in the case of a tie use the following order for breaking it A > B > C > D):

query stream	x	y	w	z	z	w	y	x
BALANCE	A	C	A	? B	? D	? --	? D	? B
OPTIMAL	C	C	A	B	D	A	D	B
GREEDY	A	C	A	B	B	--	C	--

II) What would be the optimal offline allocation of ads for the above query stream? How does it compare with the worst case evaluation for BALANCE?

OPTIMAL performs way better, since its CR = 1, whereas it is ~0.63 in worst case CR of BALANCE.

III) Compute a competetive ratio for the above given data.

CR BALANCE = 7/8
CR OPTIMAL = 8/8 = 1
} => CR = min(CR BALANCE, CR OPTIMAL) = 7/8

Three advertisers A, B, and C compete for the same keyword with the following bids: $1, $2 and $5, respectively. Their respective CTRs (click through rates) are 0.5, 0.1 and 0.2.

I) Use simple Google Adwords algorithm to rank the advertisers. Whose ad would be selected?

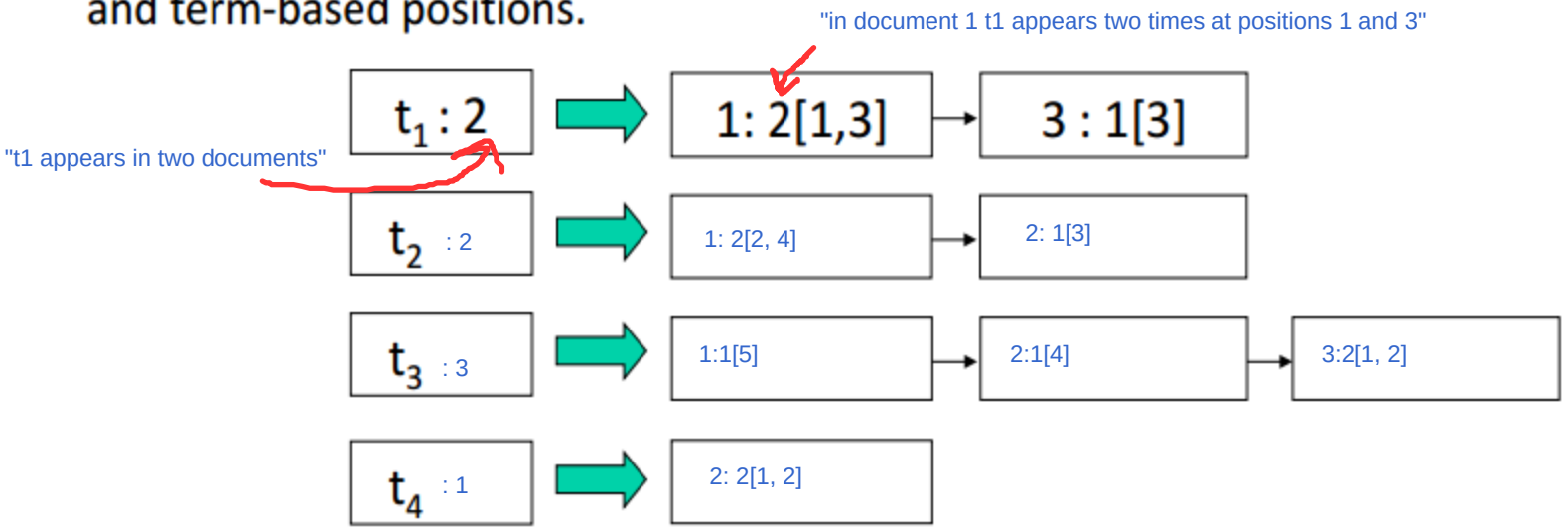
For A: 1 * 0.5 = 0.5;
B: 2 * 0.1 = 0.2;
C: 5 * 0.2 = 1.0.

} bid * CTR

II) How B needs to increase her bid so that to be ranked first?

B needs to increase its bid from 2$ to 10$ to achieve highest value for now (1.0)

I) Consider the following collection of three documents: **D1** = "$t_1\ t_2\ t_1\ t_2\ t_3$", **D2** = "$t_4\ t_4\ t_2\ t_3$", and **D3** = "$t_3\ t_3\ t_1$". Show an inverted index modeling document frequencies, term frequencies, and term-based positions.



II) Assume a biword index. Given an example of a document which will be returned for a query "Greater Poland University" but is actually a false positive which should not be returned.

I) Shown below is a portion of a **term-based positional index** in the format:

term: doc1: <position1,position2,…>; doc2: <position1,position2,…>; etc.:

fear: 2: <87,704,722>; 4: <13,43,113,433>; 7: <18,328,528>;

fools: 2: <1,17>; 4: <8,78,108>; 5:<12,101,222>; 7: <3,13,23>;

in: 2: <3,37,76,444,851>; 4: <100,130>; 7: <10,15>;

rush: 2: <2,66,194>; 3: <401>; 4: <9,69,149,569>; 7: <4,14,404>;

Which document(s) match the following phrase query: "**fools rush in**"? doc2, doc4, doc7

II) Consider the following fragment of a term-based positional index in the format:

term: doc1: <position1,position2,…>; doc2: <position1,…>; etc.

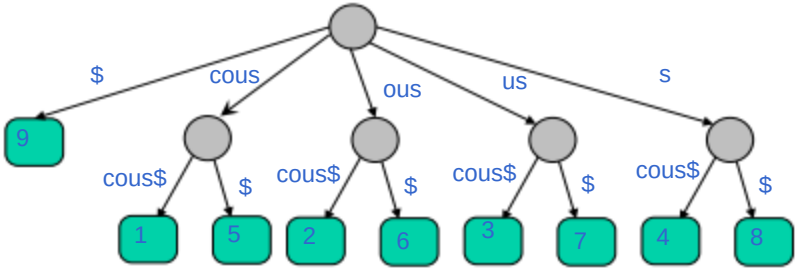
Gates: 1: <3>; 2: <6>; 3: <2,17>; 4: <1>;

IBM: 4: <3>; 7: <14>;

Microsoft: 1: <1>; 2: <1,21>; 3: <3>; 5: <16,22,51>;

The **/k** operator, **word1 /k word2** finds occurrences of word1 within k words of word2 (on either side), where **k** is a positive integer argument. Which document(s) satisfy the query "**Gates /2 Microsoft**"? 1: [3] 3: [2]
1: [1] 3: [3]

I) Show a complete suffix tree for the string: "couscous$". Indicate the root, intermediate nodes, leaves, and edges. Which is the longest repeated substring?



i	1	2	3	4	5	6	7	8	9
S[i]	c	o	u	s	c	o	u	s	$

Slide 245

II) Build a suffix array for "couscous$" using the *qsufsort* algorithm.

	i	1	2	3	4	5	6	7	8	9
h	x_i	c	o	u	s	c	o	u	s	$
	A[i]	9	1	5	2	6	4	8	3	7
	V[A[i]]	1	3	3	5	5	7	7	9	9
1	V[A[i]+h]		5	5	9	9	3	1	7	7
	A[i]	9	1	5	2	6	8	4	3	7
	V[A[i]]	1	3	3	5	5	6	7	9	9
2	V[A[i]+h]		9	9	7	6			3	1
	A[i]	9	1	5	6	2	8	4	7	3
	V[A[i]]	1	3	3	4	5	6	7	8	9
4	V[A[i]+h]		3	1						
	A[i]	9	5	1	6	2	8	4	7	3

Slide 254

To encode a number to Gamma: length N of number in unary + number in binary (without leading 1)

Length N of a number in unary: number = 2^N + the rest

To encode a number to Delta: Take length N;
Convert number N+1 to Gamma;
The final number in Delta: converted to Gamma N+1 number + initial number in binary (without leading 1)

Task IV.:
1. 2 leading zeros in 001
2. read 2 more bits i.e. 00111
3. decode N+1 = 00111 = 7
4. get N = 7 - 1 = 6 remaining bits for the complete code i.e. '000001'
5. encoded number = 2^6 + 1 = 65

I) Encode 15 in γ (0001111). 15 = 2^3 + 7 => Length N = 3 (0001 in unary)
15 in binary = 1111
15 = 0001111

II) Decode 000010101 written in the γ-code (21). Length N in unary = 00001 => N = 4
10101 from binary = 21

III) Encode 15 in δ (00100111). Length N = 3 => 0001;
N+1 = 3+1 = 4 => 4 in Gamma = 00100 (length N = 2, since 4 = 2^2; 4 in binary = 100)
15 = 00100111

IV) Decode 00111000001 written in the δ-code (65). N+1 = 00111 = 7 => N = 6
Answer: 2^6 + 1 = 65

V) Assume the most common term in the collection of documents occurred 120,000 times. Which is the predicted collection frequency for the third most frequent term according to the **Zipf's law**? cf3 = 120.000 / 3 = 40.000 (because cf r is the number of occurrences of the term r in the collection, i.e., 120.000, in case of cf1)

VI) For a collection of documents, assume k=60 and b=0.4. Which is the predicted number of different terms for the first 100,000 tokens according to the **Heaps' law**?

M = 60 * 100.000 ^ 0.4

Zipf's Law:

Predictions:
- If the most frequent term (r=1) occurs cf1 times, then:
  - the second most frequent term (r=2) occurs cf1 /2 times
  - the third most frequent term (r=3) occurs cf1 /3 times…

Slide 265

Heap's Law:
M = k*T^b, where:
- M is predicted number of terms;
- T - number of tokens

Slide 262

Slide 268