2 Evaluation

$P = \frac{TP}{TP+FP}$

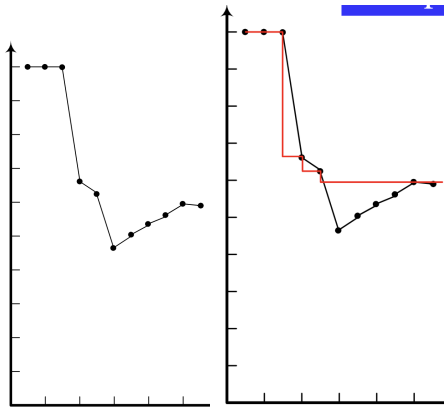
$R = \frac{TP}{TP+FN}$

$F = \frac{FP}{FP+TN}$

- Relevant documents: 20 total.

n	relevant	Recall	Precision	n	relevant	Recall	Precision
1	x	0.05	1	11			
2	x	0.1	1	12			
3	x	0.15	1	13	x	0.3	0.46
4		~~0.15~~	~~0.75~~	14	x	0.35	0.5
5				15	x	0.4	0.53
6	x	0.2	0.67	16	x	0.45	0.56
7				17	x	0.5	0.59
8	x	0.25	0.63	18			
9				19	x	0.55	0.58
10				20			

Recall	Precision
0.05	1
0.1	1
0.15	1
0.2	0.67
0.25	0.63
0.3	0.46
0.35	0.5
0.4	0.53
0.45	0.56
0.5	0.59
0.55	0.58



$P_{interp}(r) = max_{r' \geq r} p(r')$ # best current or future performance

Precision-recall curve - take average of precision at 11 levels of recall from 0 to 1

3 Boolean

Term-document incidence matrix

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

Generate Postings

Doc 1. **I did enact julius caesar I was killed i' the capitol brutus killed me.**

Doc 2. **so let it be with caesar the noble brutus hath told you caesar was ambitious**

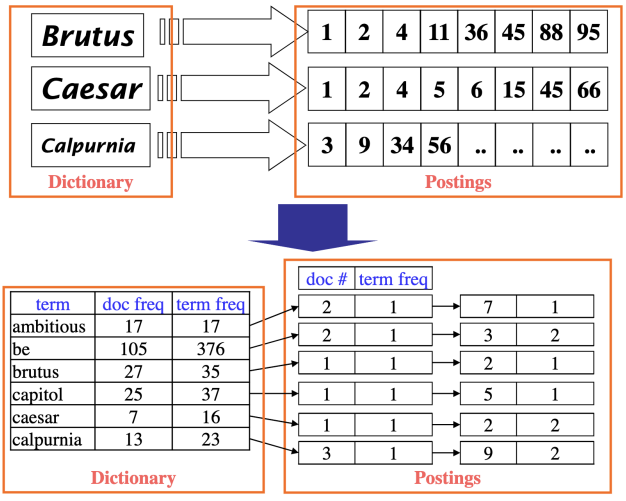
term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
……	……

Sort Postings

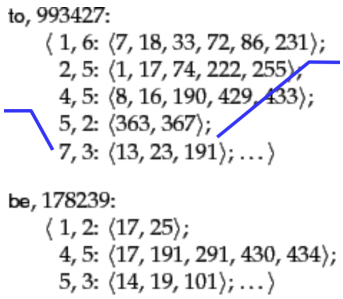
term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i’	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
……	……

term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i’	1
it	2
julius	1
killed	1
killed	1
let	2
……	……

Inverted index:



Positional index (t,df,tf->d#,tf->...):



4. Vector Space

$jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}$

$jaccard(A, A) = 1$

$jaccard(A, B) = 0 \; if \; A \cap B = 0$

Set of words model - no order, no duplicates

Bags of words model - no order, with duplicates

$tf_{t,d}$ - frequency of term t in document d

df_t - number of documents that contain term t (if higher, term is less informative)

N - total number of documents

$idf_t = \log_{10} \frac{N}{df_t}$ (log dampens the effect of idf)

N=1M:

term	DF	IDF
calpurnia	1	6
animal	100	4
sunday	1,000	3
fly	10,000	2
under	100,000	1
the	1,000,000	0

$w_{t,d} = tf_{t,d} * idf_t = tf_{t,d} * \log_{10} \frac{N}{df_t}$ # tf-idf weight # best known

$tf_{t,d} = 1 + log_{10} tf_{t,d} \quad if \; tf_{t,d} > 0 \quad else \; 0$ # won't be used on exams

term frequency		document frequency		normalization	
n (natural)	$tf_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1+\log(tf_{t,d})$	t (idf)	$\log \frac{N}{df_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5+\frac{0.5\times tf_{t,d}}{\max_t(tf_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N-df_t}{df_t}\}$	u (pivoted unique)	$1/u$ (Section 17.4.4)
b (boolean)	$\begin{cases}1 \text{ if } tf_{t,d}>0\\0 \text{ otherwise}\end{cases}$			b (byte size)	$1/CharLength^{\alpha}$, $\alpha<1$
L (log ave)	$\frac{1+\log(tf_{t,d})}{1+\log(ave_{t\in d}(tf_{t,d}))}$				

$D1 = [w_{1,1} = tf_{\text{"a"},D1} \times idf_{\text{"a"}}, w_{2,1}, \ldots, w_{N,1}]$

Document vectors in vector space:

	$\|D_i\|$	a	arrived	damaged	delivery	fire	gold	in	of	silver	shipment	truck
df		3	3	2	1	1	2	3	3	1	2	3
idf		0.125	0.125	0.301	0.602	0.602	0.301	0.125	0.125	0.602	0.301	0.125
D_1	0.825	0.125		0.301		0.602	0.301	0.125	0.125		0.301	
D_2	1.375	0.125	0.125		0.602			0.125	0.125	1.204		0.125
D_3	0.509	0.125	0.125				0.301	0.125	0.125		0.301	0.125
D_4	0.349		0.125	0.301								0.125

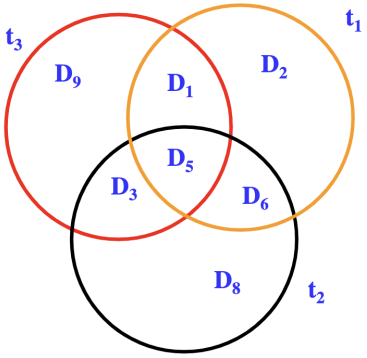
Row: document vector. Column: term dimension

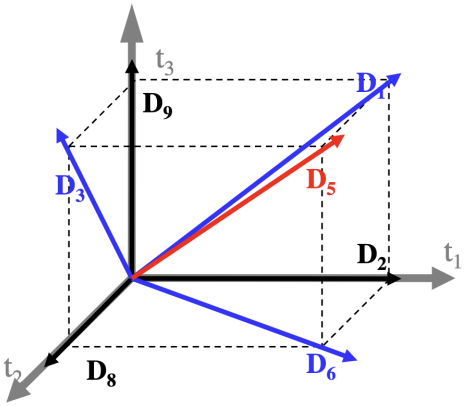
(optional normalization:)

	$\|D_i\|$	a	arrived	damaged	delivery	fire	gold	in	of	silver	shipment	truck
df		3	3	2	1	1	2	3	3	1	2	3
idf		0.125	0.125	0.301	0.602	0.602	0.301	0.125	0.125	0.602	0.301	0.125
D_1	1	0.151		0.365		0.730	0.365	0.151	0.151		0.365	
D_2	1	0.091	0.091		0.438			0.091	0.091	0.876		0.091
D_3	1	0.245	0.245				0.592	0.245	0.245		0.592	0.245
D_4	1		0.358	0.862								0.358

For queries: create vector out of idf

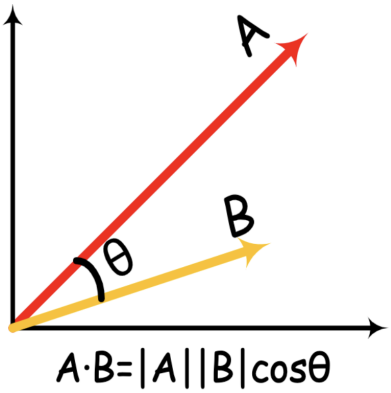
Boolean VS Vector models:





The following two notions are equivalent.

- Rank documents in increasing order of the angle between query and document
- Rank documents in decreasing order of the cosine of the angle.



$$\cos(\vec{q},\vec{d}) = \frac{\vec{q}\cdot\vec{d}}{|\vec{q}||\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

Cosine similarity:

$$sim(D_1, D_2) = \frac{\sum_{i=1}^{t} w_{1i} * w_{2i}}{\sqrt{\sum_{i=1}^{t}(w_{1i})^2} * \sqrt{\sum_{i=1}^{t}(w_{2i})^2}}$$

4.5 Efficiency

Efficient ranking: pruning
Find a set A of contenders, with $K < |A| << N$
A does not necessarily contain the top K, but has many docs from among the top K
Return the top K docs in A

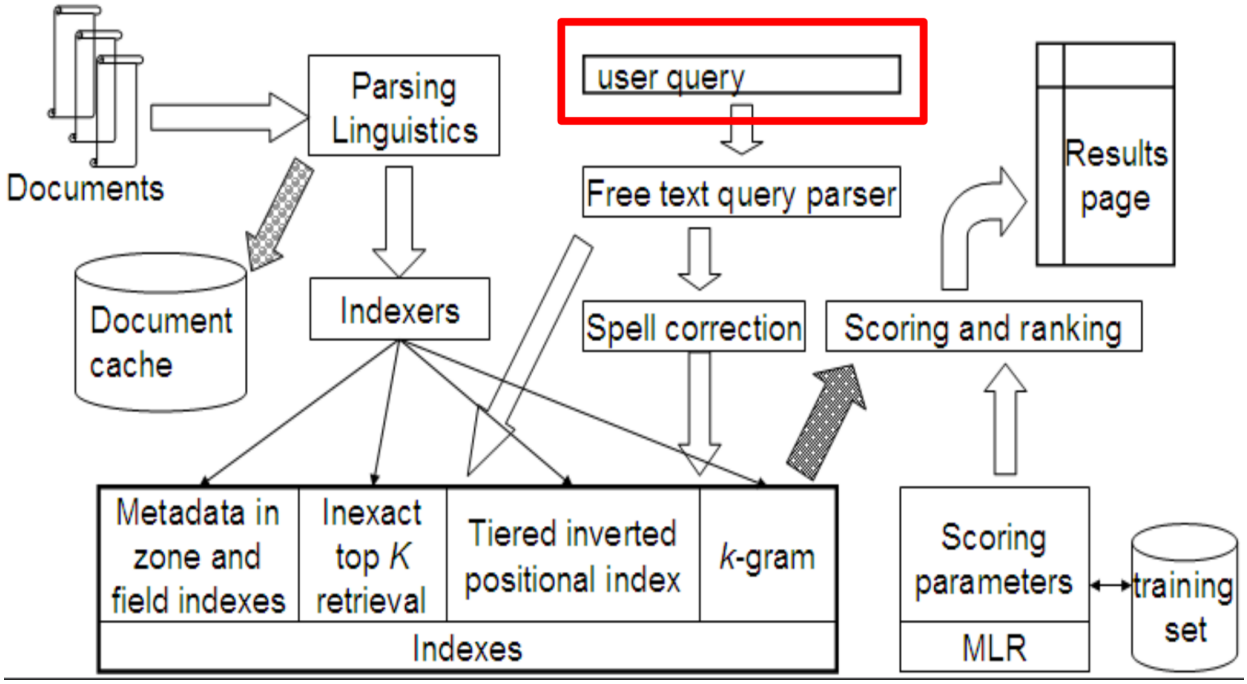
- Only consider high-idf query terms (i.e, without the, in)
- Only consider docs containing many query terms (i.e >1)
- Only consider top docs for each query term

Champion list for t - pre-compute for each dictionary term t, the r docs of highest weight in t's postings
Plus, order documents by authority of source - and then compute cosine for first k

5 Text

Remove stop words (to, in)
Tokenize (split compound nouns, split on punctuation)
Normalize (color, colour)
    Solutions: index under both, or expand query to search under both
Lemmatization (dupes->dupe, is->be)
Stemming (building->build, automates->automat)

6 IR Systems



Evaluation criteria:

- recall and precision
- response time
- user effort
- form of presentation
- content coverage

7 Feedback

Add the vectors for the relevant documents to the query vector.
Subtract the vectors for the irrelevant docs from the query vector.

$\vec{\mu}(C) = \frac{\sum_{\vec{d} \in C} \vec{d}}{|C|}$ # Centroid. C - set of documents

$\vec{q_m} = a\vec{q_0} + \beta \frac{\sum_{\vec{d_i} \in D_r} \vec{d_i}}{|D_r|} - \gamma \frac{\sum_{\vec{d_j} \in D_{nr}} \vec{d_j}}{|D_{nr}|}$ # modified query vector

D_r # set of known relevant doc vectors

D_{nr} # set of known irrelevant doc vectors

Different from C_r and C_{nr}

q_0 # original query vector

α, β, γ # weights, hand-chosen

D4=[0 0.602 0.301 0.602 0.301 0.301 0.301 0.249]
|D4|=1.072 (sqrt(x^2+y^2)) # Normalize: divide each by |D4|
D4' = [0 0.5616 0.2808 0.5616 0.2808 0.2808 0.2808 0.2323]
Q1=[0 0 0 0 0 0 0 0.125]
Q = α * Q1' + β * D4' = Q1' + 0.5 * D4'
    = [0 0 0 0 0 0 0 1] + 0.5 * [0 0.5616 0.2808 0.5616 0.2808 0.2808 0.2808 0.2323]
    = [0 0.2808 0.1404 0.2808 0.1404 0.1404 0.1404 1.1161]
|Q| = 1.2175
Normalization: Q' = [0 0.2306 0.1153 0.2306 0.1153 0.1153 0.1153 0.9168]
sim(Q', D4) = 0.6015

TODO

Print notes
Review notes
EXAM Overview