Information Retrieval

→ Data :- (structure-wise)
- Structured
- Semi-structured
- Unstructured

→ Data Nature :- wise
- image
- text ✓
- audio
- video

Raw data without interpretation only text & number.

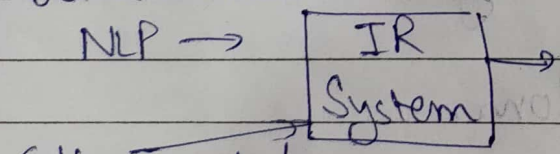
data
↓
information (data with some interpretation).
↓
knowledge
↓
wisdom.

→ Database only support exact match.

PAGE NO.: _____
DATE: / /

DB	IR/Web search.
1) Fixed format of Data	1) Free flowing data:- text, audio, video, image.
2) Structured Data type	2) * Unstructured *
3) Knowledge of Schema.	3) No knowledge is required
4) Parameterized mode of Querying.	4) Natural Language
5) Exact search.	5) Approximate search
6) Set based search.	6) Rank-based retrieval.

Retrieval user:-



Google has accessed to several billions page.

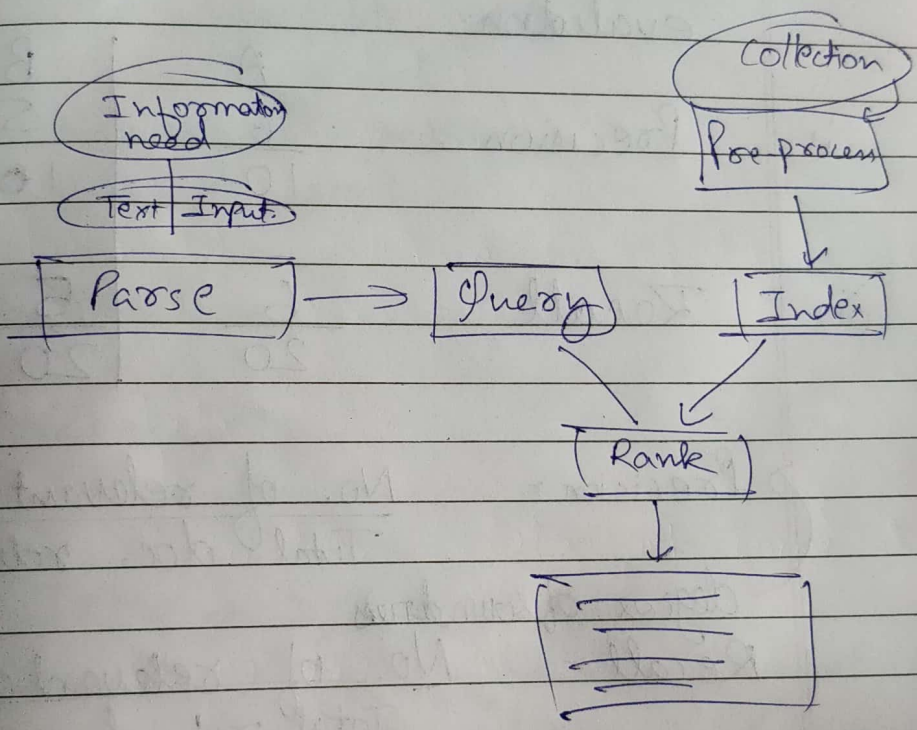
20-30% publicly available Surface web

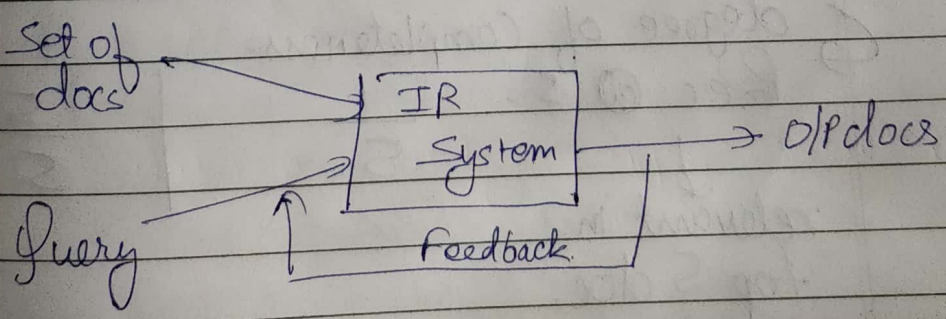
Small subset of collection.

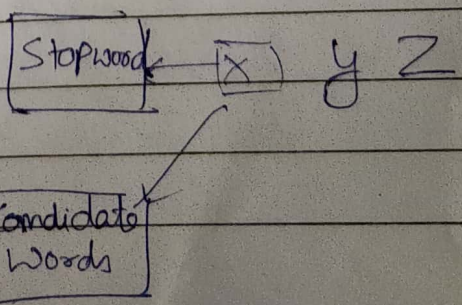
PAGE NO.:
DATE: / /

Retrieval ← matching
ranking

Information Retrieval Process







meeting (user need)

PAGE NO.:
DATE: / /

IR — Retrieval
IR — Evaluation

2 basic technique for evaluation:

		A	B
i)	Precision	6/10	5/10
ii)	Recall	6/20	5/20

Precision = No. of relevant doc. / Total doc. returned.
degree of soundness

Recall = No. of relevant doc. / Total rel. document
degree of completeness.

Prec @ 5 = 1/5 | 5/5

↓ relevant in top 5 doc.

Cristopher D Manning Prabhakar Raghvan.

PAGE NO.:
DATE: / /

IR applications:-

(i) Web search.
(ii) Desktop search
(iii) email search
(iv) Enterprise search.
(v) Bibliographic.
(vi) Legal search
(vii) Medical search
(viii) Social search.

Filtering:-

Clustering
Categorization.

IRM = $< D, Q, F, R(q_i, q_j) >$

D: Document, Q: Query, F: System, R: Ranking Function.

Measurable Properties.

⇒ Response time of System.

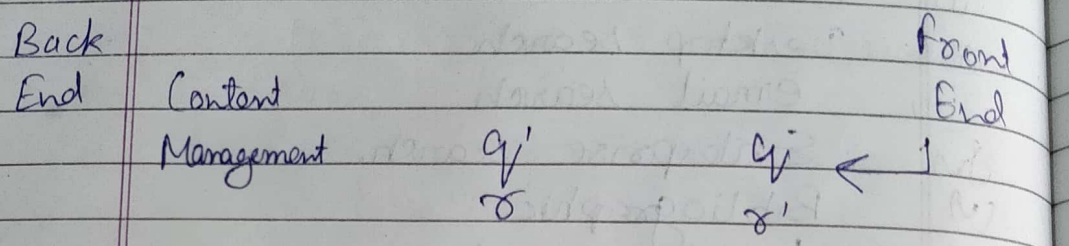
how fast it process?

No. of doc./ho.

how fast it search?

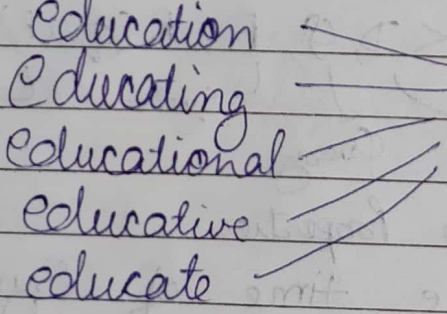
PAGE NO.:
DATE: / /

High level view of IR process.



Case folding:- Bring all the characters in same case.

Stemming:- Provides the root of any word by chopping off prefix or suffixes.



Lemmatization

ii) Getting the root from (lemma) of a word by applying rules of the lang. concerned.
Eg:- countrymen → country man.

We either use lemmatization or stemming not both.

Document Parsing.

Stopword Removal.

Phrases :- 'New', 'York' → 'Newyork'

Indexing.

docs

↓

tokenization

↓

Case folding

↓

Stemming / lemmatization.

↓

Stopword removal

↓

list of terms ← single-words, phrases.

PAGE NO.: _____
DATE: / /

Inverted index construction.

index means list of words with document id.

Stemming

Lang. dep.

Lang. independent

↓

Replacement ×

Cement ×

↓ result in false goal.

Language independent technique are mostly statistical.

Download Porter's

Saw ← Past of see

saw.

lemmatization :- NLP tool.

$10^6 \times 6 \times 1000$ $\quad$ 100×1.5
6×10^9 $\quad$ 0.4

PAGE NO.:
DATE: / /

We need variable-size posting list.
On disk, a continuous run of postings is normal and best.

Boolean Retrieval.

Considering only presence (1) and absence (0) of terms are connected to each other by AND, OR, NOT

merge op.	$O(m+n)$
And result	$O(\min(m,n))$
OR result	$O(m+n)$.

From documents we get term. that's why invert indexing.

$O(n^2)$

n = 1 Gb dictionary in Mb.

Postings are usually larger $O(n)$.
- Normally kept on Disk.

Heap's law:-

$$V = K N^{\beta} \qquad \beta \in [0.4, 0.6]$$

const.

V = Voc. size / dictionary length.

N = total no. of terms in collection.

PAGE NO.: _____
DATE: / /

Dictionary size < posting size:

When taking and operation take two small document and ~~min~~. of first

$$\phi = t_1 \$ t_2 \text{ --- } \$ t_n$$

Process in order of increasing freq.

Start with smallest one first.

Word	doc-freq.
t_1	n_1
t_2	n_2
-	-

Dict.

PAGE NO.:
DATE: / /

⇒ Introduction to Information Retrieval:-

dictionary postings

<term, doc. freq> <doc ids>

single-word Multi words phrases.

→ search:- "Stanford University"

Not Relevant ⇒ I found Stanford Henry in the univ. campus. ← Doc 9

Stanford:- 1 3 5 9

University:- 3 9 11 25

Stanford University:-

Proximity Search

iit Varanasi

iit at Varanasi.

Store location of the word in the document in our posting list.

It will help in proximity search.

fools rush in.

	2	4	7
fools	1	8	3, 13
rush	2	9	4, 14
in	3	10	5, 15

{2, 4, 7}

angels fear to thread:

	2	4
angels	x	12
fear	x	13
to	x	14
thread	x	15

{4}

{4}

positional index size

A positional index expands postings storage substaintially.

2:<1> 4:<8> 7:<5,13>

PAGE NO.:
DATE: / /

- A positional index is 2-4 as large as a non-positional index.
- Positional index size 35-50% of volume of original text.

For some popular phrases, we can use biword indexes.

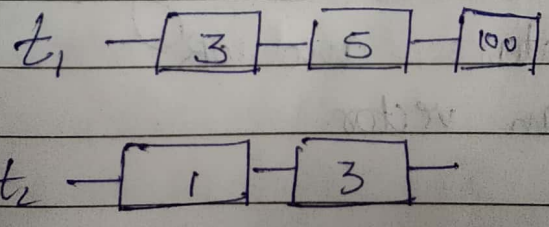
~~Combiens~~ Combination :-

It required 26% more space than having a positional index alone.

t_1 and t_2 and --- t_{n-1} and t_n match
$n-1$ match = 0 match.

OR t_1 OR t_2 OR ---- t_n.
n matches ≡ 1 match.

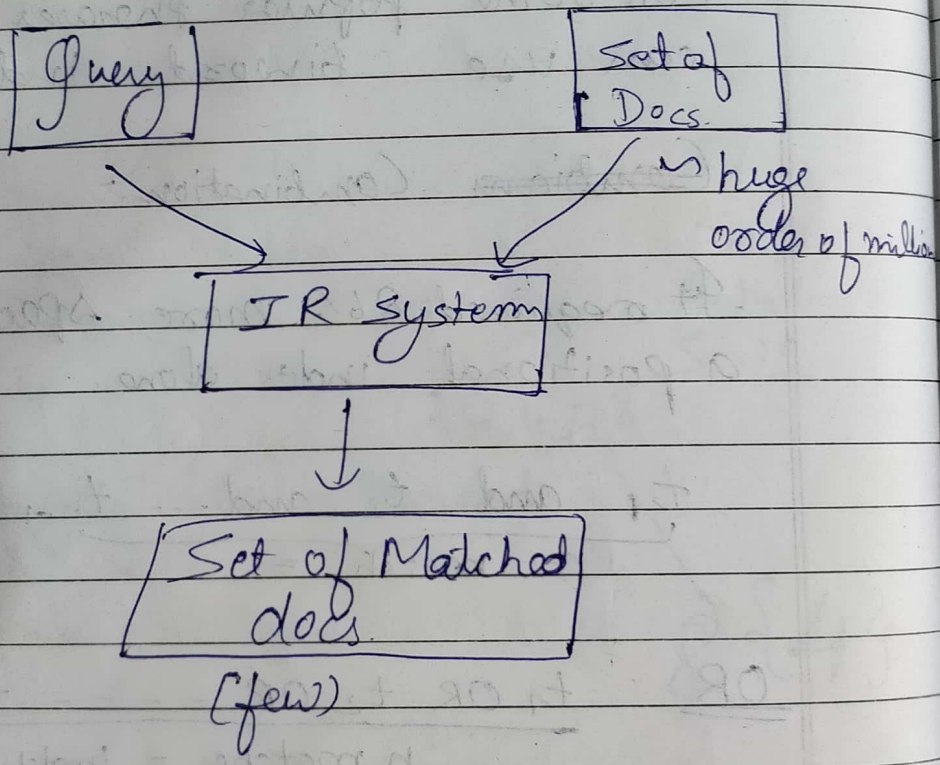
Skip pointers



PAGE NO.:
DATE: / /

Retrieval Models

- Boolean → Set based retrieval/ ranking not possible.
- Vector Space Models
- Probablistic



$Q = t_1, t_2 \ldots t_n$

t_i's connected by boolean operators and OR

= Boolean vector.

SISLPA

PAGE NO.:
DATE: / /

	t_1	t_2	...	t_n
D_1	1	0	0	1
D_2	0	1	0	0
D_3	:			
D_n	0	...		1

Vector space model.

- Both docs & query are considered to be vectors with non-negative term weights. (importance of term in the doc.)
- Terms are dimensions in the vector space.

- Term-weight $\propto$ term-frequency.

D_3 1000 words

50 t_1 40 t_i 60 t_j

- Term-weight $\propto \frac{1}{\text{doc-length}}$.
- Term-weight $\propto$ rarity of the term.
$\propto$ inverse doc-frequency (idf) $\propto \frac{1}{\text{document-frequency (df)}}$

= # docs where a term is present.

query → verbose

PAGE NO.:
DATE: / /

$$tw \propto tf \cdot idf$$
$$\propto \frac{1}{dl}$$

$Q = \langle 0, w_{iq}, w_{kq}, \ldots 0 \rangle$

w_{kq} → weight of the term i in query q

$D_1 = \langle w_{11}, w_{21}, \ldots w_{t1} \rangle$
$D_2 = \langle w_{12}, w_{22}, \ldots w_{t2} \rangle$
$D_j = \langle w_{1j}, w_{2j}, \ldots w_{tj} \rangle$

Closness check:-
1) Euclidean Distance.

Angle $\angle^{D}_{q}$ will represent the closeness b/w D & Q.

$Sim(Q, D_i) = \angle^{D_i}_{Q}$

0° represent max. closeness.
90° represent min. closeness.

PAGE NO.:
DATE: / /

~~Similarity~~

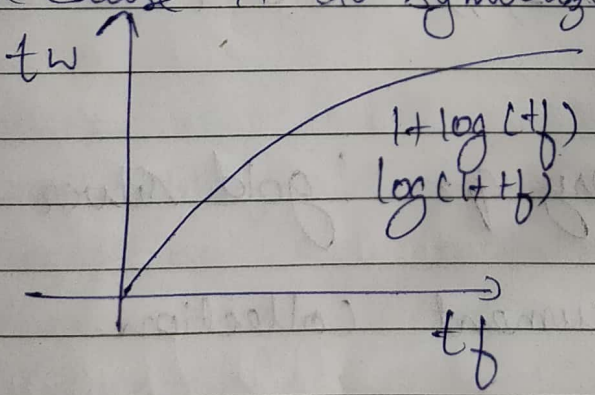
Similarity,

$$Sim(Q, D_j) = \cos(\angle_{Q}^{D_j})$$

$$= \frac{\vec{D_j} \cdot \vec{Q}}{|\vec{D_j}| \cdot |\vec{Q}|}$$

$$= \frac{w_{1j} * w_{1q} + w_{2j} * w_{2q} + \dots w_{tj} * w_{tq}}{\sqrt{w_{1j}^2 + w_{2j}^2 \dots w_{tj}^2} \times \sqrt{w_{1q}^2 + w_{2q}^2 + \dots w_{tq}^2}}$$

We don't use raw term weights because it do symbolize real importance.



Similarly $\log\left(\frac{N}{df}\right)$

other technique:

$$tf = 0.5 * + 0.5 \times \frac{tf}{max.tf}$$

PAGE NO.: _____
DATE: / /

Weight = tf x idf/normalization.

Q. Consider two Documents D_1, D_2 & query Q. $D1 = (0.5, 0.8, 0.3)$
$D_2 = (0.9, 0.4, 0.2)$, $Q = (1.5, 1.0, 0)$.

$$\cos(D1, Q) = \frac{0.5 \times 1.5 + 0.8 \times 1}{\sqrt{1.5^2 + 1} \quad \sqrt{0.5^2 + 0.8^2 + 0^2}}$$

$$\cos(D2, Q) = \frac{0.9 \times 1.5 + 0.4}{\sqrt{1.5^2 + 1} \times \sqrt{0.9^2 + 0.4^2 + 0.2^2}}$$

$$\cos(D1, Q) = \underline{0.87}$$

$$\cos(D2, Q) = 0.965 = 0.97$$

Q2. Query q_2 = 'gold silver truck'.

Document collection.

d_1 = 'shipment of gold damaged in a
d_2 = 'Delivery of silver arrived in a truck
d_3 = 'shipment of gold arrived in a truck'.

	D_1	D_2	D_3
Shipment	1/2	0	1/2
of	1/3	1/3	1/3
gold	1/2	0	1/2
damaged	1	0	0
in	1/3	1/3	1/3
a	1/3	1/3	1/3
fire	1	0	0
delivery	0	1	0
silver	0	2	0
arrived	0	1/2	1/2
truck.	0	1.1/2	1/2

$$(1 + \ln(1+f)) \times \log\left(\frac{N}{df}\right)$$

N = 3

Pros:-

i)

Cons:-

ii) Impossibility of formulation of structured qu

iii) Assumption of stochastic independency between term. get rid off (all are orthogonal)

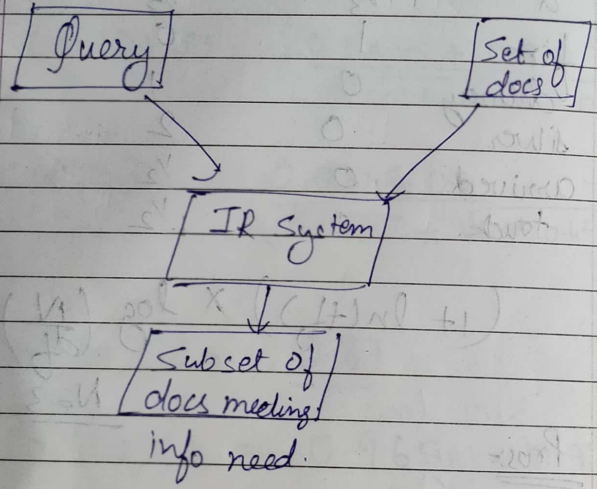
iii) Terms are axes.

10.00

PAGE NO.:
DATE: / /

- Evaluation

IR
- Retrieval



In evaluation, user's satisfaction on the information need met by the system-output.

1. Set based output / Retrieval.
2. Ranked Retrieval

O/P: A set of docs. $\{d_1^R, d_2^{NR}, d_3^R \text{ -- } d_{10}^R\}$

5 R

5 NR.

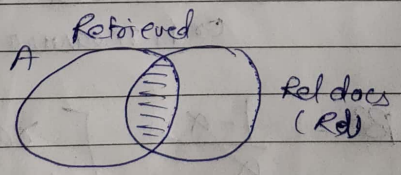
PAGE NO.:
DATE: / /

Relevance is decided by the user particularly the user who posed the query

Precision (P) = $\frac{\#\text{ relevant retrieved}}{\#\text{ retrieved}}$

Recall (R) = $\frac{\#\text{ relevant retrieved}}{\#\text{ Total relevant}}$

3 System.	Total relevant. 50.	
5R	9R	12R
5NR	11NR	18NR
P score $\frac{5}{10}$	$\frac{9}{20}$	$\frac{12}{30}$
R score $\frac{5}{50}$	$\frac{9}{50}$	$\frac{12}{50}$



$$P = \frac{|A \cap Rel|}{|A|}$$

$$R = \frac{|A \cap Rel|}{|Rel|}$$

PAGE NO.:
DATE: / /

f_1 - measure

= H.M of P & R.

$$= \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P+R}$$

F-measure = $$\frac{1}{\frac{\alpha}{P} + \frac{1-\alpha}{R}}$$

$0 \leq \alpha \leq 1$

$\alpha > 0.5$ means more importance is given to precision term.

$\alpha = 0.5$ balanced F-measure or f_1 measure.

$\beta^2 = \frac{1-\alpha}{\alpha}$ ← ratio of importance

F_1 → means β is 1.

Obfuscate

$$F_1 = \frac{2PR}{P+R} = \frac{2 \times \frac{1}{10} \times \frac{5}{10}}{\frac{1}{10} + \frac{1}{10}}$$

$$= \frac{\frac{1}{5}}{1 + \frac{1}{5}} = \boxed{\frac{1}{6}}$$

$$= 0.16$$

$$F_1 = \frac{2 \times \frac{1}{10} \times \frac{9}{50}}{\frac{1}{20} + \frac{9}{50}} = \frac{\frac{9}{50}}{\frac{1}{5} + \frac{2}{10}}$$

$$= \frac{9}{35} = 0.25$$

$$F_1 = \frac{2 \wedge 0.4 \times 0.4}{0.64} = 0.3$$

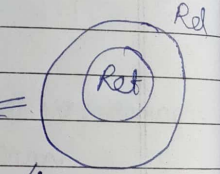
AM is influenced more by outliers while HM does not hence we take HM.

PAGE NO.:
DATE: / /

Basic Metrics

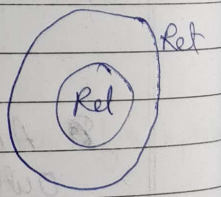
	Sys1	P	R
1	NR	0/1	0/50
2	R	1/2	1/50
3	NR	1/3	1/50
4	R	2/4	2/50
5	R	3/5	3/50
6	R	4/6	4/50
7	NR	4/7	4/50
8	NR	4/8	4/50
9	NR	4/9	4/50
10	R	5/10	5/50

$$P = \frac{|Rel \cap Ret|}{|Ret|}, \quad 1 \Leftarrow$$



$$R = \frac{|Rel \cap Ret|}{|Rel|} < 1$$

$$R = 1 \Rightarrow$$



$$F = \frac{1}{\frac{\alpha}{P} + \frac{1-\alpha}{R}}$$

$$0 \le \alpha \le 1$$

$\alpha = 0.5$ (Balanced).

Tot. Rel. =

PAGE NO.:
DATE: / /

Average Prec = $\frac{1}{|Rel|} \sum (\frac{1}{2} + \frac{2}{4} + \frac{3}{5} + \frac{4}{7} + \frac{5}{9})$

$AP_1 = \frac{1}{50}$

Average. Prec = $\frac{1}{(\text{Total Rel.})} \sum_{i \in \text{Ranks where Reldoc is present}} \text{Prec}_i$

$AP_2 = \frac{1}{50} \left(1 + \frac{2}{5} + \frac{3}{6} + \frac{4}{7} + \frac{5}{9}\right)$

AP is a top-heavy measure, high Prec. gives high AP score.

But it also takes into account recall. So it is reasonably balanced.

Missing terms implies 0 Prec. In other words, those rel docs are retrieved at rank infinity.

PAGE NO.:
DATE: / /

Interpolated Precision:

→ To compare individual precision values at diff. ranks is difficult sometimes.

→ Better to compare prec. at diff. recall-values.

$$Prec_{int}(r) = \max_{r' \geq r} Prec(r')$$

$$0 \leq r', r \leq 1$$

$P_{int}(0)$ = highest p-value

- Non increasing function.

	Relevant	Non-rel.
Ret.	TP	FP
Non-Retr.	FN	TN

$$Prec. = \frac{TP}{TP+FP}$$

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

PAGE NO.:
DATE: / /

Fall-out = $\frac{FP}{FN+-}$

Interpolated precision (r) = $\max_{r' \geq r}$ prec. (r')

$P_{int}(r) = \max_{r' \geq r} P_{nat.}(r')$

interpolated precision — natural precision.

If there are 2 queries Q_1 = 10 relv. Q_2 = 7 relv. we cannot compare them by natural precision as they will have diffent recall value, at that point but by interpolated precision we can do it.

11- Standard recall points.

= {0, 0.1, 0.2 . . . 1.0}

101- standard recall points.

= {0, 0.01, . . . 0.99, 1.0}

$AP_{net.} = \frac{1}{|Rel|} \sum Prec.(r_i)$

$r_i \in$ ranks where Reldoc is found.

PAGE NO.:
DATE: / /

Rel = set of reldocs.

$$AP_{int} = \frac{1}{|Rel|} \sum_{d_i \in Rel} P(r_i)$$

$$AP_{int} = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots 1.0\}} P_{int}(r)$$

Mean Avg. Prec. (MAP)

$$= \frac{1}{|Q|} \sum_{t \in Q} AP(t)$$

Q = Set of Queries/Topics.

$\frac{MAP_{not}}{MAP_{int}}$ } $|Q| = 50, 100.$

P@K = Precision at rank K.

P@r = (r-precision)
= P@50

P@R = Recall at R.

Mean average precision gives priority to top heavy documents.

MAP cannot be used for web-search because we cannot get no. of relevant documents.

Cumulative Grain (Jarvelin & Kekalainen) Tois, 2002.

- 4 grades of rel
  - 3 (highly rel)
  - 2 (moderately rel)
  - 1 (Marginally rel)
  - 0 (Non-relevant).

Ranks	Grain	CG.	Ideal (CG. I)
1	3	3	3 3
2	2	5	3 6
3	3	8	3 9
4	0	8	2 11
5	0	8	2 13
6	1	9	2 15
7	2	11	1 16
8	2	13	0 16
9	3	16	0 16
10	0	16	0 16

PAGE NO.:
DATE: / /

n.C.G = CG/CG$_I$

3/3 = 1

5/6

8/9

8/11

8/13

9/15

11/16

13/16

16/16

16/16

Discounted Cumulative Gain

Logarithmic discount base 2

Discounted gain (r) = $\frac{\text{Gain}(r)}{\log_2(r)}$ where $r \geq 2$

Gain (r) otherwise

DCG	DCG$_i$
3	3
5	6
5 + 3/log 3	6 + 3/log$_2$ 3
DCG(3) + 0	DCG(3) + 2/log 4
DCG(4) + 0/log 4	DCG(4) + 2/log 5
DCG(5) + 1/log 6	DG(5) + 2/log 5
DCG(6) + 2/log 7	DCG(6) + 1/log 6
DCG(7) + 2/log 8	DG(7)
	~~DCG(8)~~
	~~DCG(9)~~

Scanned by CamScanner

5th

PAGE NO.:
DATE: / /

DCG(8) + 3/log 9 | DCG(7)
DCG(9) + 0 | DCG(7)

NDCG.
1
5/6
6.89/7.89
6.89/8.89

Evaluation
- Set Based — Precision, Recall, F-measure
- Ranked
  - Binary relevance — P@K, MRR, AP, Reciprocal Rank (RR) = 1 / rank of first rel doc.
  - Graded rel — CG, nCG, DCG, nDCG.

Mean Reciprocal Rank.

$$MRR = \frac{1}{|Q|} \sum_{t \in Q} \frac{1}{RR(t)}$$

t is a topic in set Q.

Scanned by CamScanner

B-pref.
RBP
Stat AP.

Relevance Judgement

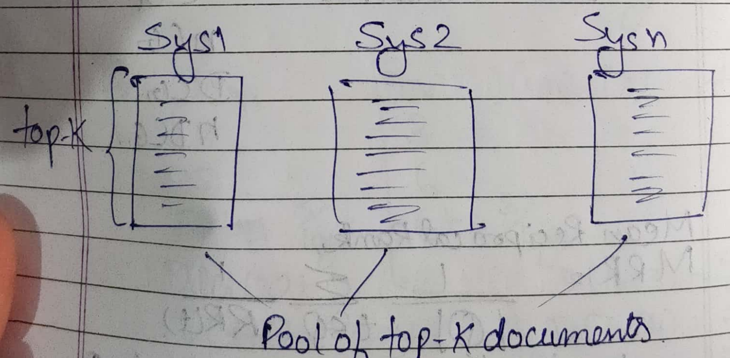
Test bed : 1) A set of documents (D)
2) A set of topics (Q)
3) Relevance Judged info

For each query (t) what are the docs which are relevant.

50 x 10,000 manual labelling } infeasible
queries   (no. of document)

System no. >50 — less no. of system won't work (3)

Sampling: top-heavy sampling.



PAGE NO.:
DATE: / /

Assumptions of Cranfield Evaluation

1) All the systems are arranging relevant docs in decreasing order of relevance (or preferance of the system).

2) All the systems can collectively identify all the relevant docs in the collection (D) for a given query within top-k ranks.

(system no. → 50; less no. of system can't work)

3) There is no relevant doc outside the pool created based on top-k pooling.

Topic creators are the best (assessors).

Inter-assessor agreement

$$K \text{ (Kappa)} = \frac{P(A) - P(E)}{1 - P(E)}$$

$$= \frac{\frac{325}{400} - \left(\frac{275}{400} \times \frac{300}{400} + \frac{125}{400} \times \frac{100}{400}\right)}{1 - \frac{825}{1600}}$$

$$= \frac{\frac{325}{400} - \frac{825}{1600}}{1 - \frac{825}{1600}} = \frac{1300 - 825}{1600 - 825} = \frac{475}{775} = 0.61$$

PAGE NO.:
DATE: / /

Retrieval Model

- Boolean
- VSM
- Probabilistic

Probability

Event

Random variable → It's a function that maps an event to a real number.

$0 \leq P(X = \quad) \leq 1$

Joint Probability: A, B... $P(A \cap B) = P(AB)$

A, B, C... $= P(A \cap B \cap C \ldots) = P(ABC \ldots)$

Conditional Probability:

$P(A|B) = \frac{P(A \cap B)}{P(B)}$

Occurrence of A ← | → has occured

$P(A \cap B) = P(A|B)P(B) = P(B|A).P(A)$

$P(A_1 A_2 \ldots A_n) = P(A_1 | A_2 \ldots A_n) \times P(A_2 | A_3 \ldots A_n) \ldots P(A_{n-1} | A_n) \times P(A_n)$.

Bayes' Rule:

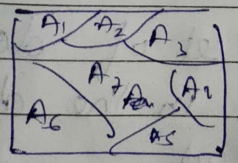
$A_1, A_2, \ldots A_n$ are a set of mutually exclusive exhaustive set of event & B is another event then

$$P(A_i | B) = \frac{P(A_i \cap B)}{P(B)}$$

$$= \frac{P(B|A_i) \times P(A_i)}{\sum_{j=1}^{n} P(B \cap A_j)}$$

$$= \frac{P(B|A_i) \times P(A_i)}{\sum_{j=1}^{n} P(B|A_j) \times P(A_i)}$$

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B|A)\,P(A) + P(B|A')\,P(A')}$$



$B = B \cap U$

$B = B \cap (A_1 \cup A_2 \ldots A_n)$

$P(B) = P(B \cap A_1) + P(B \cap A_2) + \ldots P(B \cap A_n)$.

PAGE NO.:
DATE: / /

odd of an event:

$$O(A) = \frac{P(A)}{P(\bar{A})}$$

$$= \frac{P(A)}{1 - P(A)}$$

Probability Ranking principle:- (PRP)

For any given query (q) we can consider a set of reldocs (R=1) and a set of non-rel docs (R=0) in the collection.

Each document will belong to (R=1) with some prob. $P(R=1|d,q)$ and to R=0 with prob. $P(R=0|d,q)$

s.t. $P(R=1|d,q) + P(R=0|d,q) = 1$.

We can rank a set of docs for a given query q in the decreasing order of $P(R=1|d,q)$.

PAGE NO.:
DATE: / /

Binary independence model (BIM).

A doc is an ordered set of terms. $\vec{d} = (x_1, x_2 \ldots x_m)$ where each x_i is 0 or 1.

Similarly $\vec{q} = (x_1, x_2, \ldots x_n)$.

$$P(R=1|\vec{d},\vec{q}) = \frac{P(R=1|\vec{d},\vec{q})}{P(\vec{d},\vec{q})}$$

$$= \frac{P(\vec{d}|R=1,\vec{q}) \times P(R=1|\vec{q}) \times P(\vec{q})}{P(\vec{d}|\vec{q}) \cdot P(\vec{q})}$$

$$= P(R=1|\vec{q}) \times \frac{P(\vec{d}|R=1,\vec{q})}{P(\vec{d}|\vec{q})}$$

$$O(R=1|\vec{d},\vec{q}) = \frac{P(R=1|\vec{d},\vec{q})}{P(R=0|\vec{d},\vec{q})}$$

$$O(R=1|\vec{q}) = O(R=1|\vec{q}) \times \frac{P(\vec{d}|R=1,\vec{q})}{P(\vec{d}|R=0,\vec{q})}$$

$$= O(R=1|\vec{q}) \times \frac{P(x_1, x_2 \ldots x_m|R=1,\vec{q})}{P(x_1, x_2 \ldots x_m|R=0,\vec{q})}$$

$$= O(R=1|\vec{q}) \times \frac{\prod P(x_i|R=1,\vec{q})}{\prod (P(x_i|R=0,\vec{q}))}$$

[x_i's are occuring independ

PAGE NO.: DATE: / /

$P(x_i = 1 \mid R=1, q)$
↳ Probability of presence of term x_i in Relevant doc.

$= O(R=1 \mid q) \times \prod_{x_i=1} \frac{P(x_i \mid R=1, q)}{\ldots} \times \prod_{x_i=0} \frac{P(x_i \ldots}{\ldots}$

	R = 1	R = 0
x_i present $x_i = 1$	p_i	u_i
x_i absent $x_i = 0$	$1-p_i$	$1-u_i$

$O(R=1 \mid d, q) = O(R=1 \mid \vec{q}) =$

$$= \frac{\prod_{x_i=1} p_i \times \prod_{x_i=0} (1-p_i)}{\prod_{x_i=1} u_i \times \prod_{x_i=0} (1-u_i)}$$

When $q_i = 0$ $p_i = u_i$ [Assumption

~~$= O(R=1 \mid q) \times \prod$~~

$$= O(R=1 \mid q) \times \frac{\prod_{x_i=1, q_i=1} p_i \times \prod_{x_i=0, q_i=1} (1-p_i)}{\prod_{x_i=1, q_i=1} u_i \times \prod_{x_i=0, q_i=1} (1-u_i)}$$

Scanned by CamScanner

PAGE NO.:
DATE: / /

$$= O(R=1|q) \cdot \frac{\prod_{x_i=q_i=1} p_i}{\prod_{x_i=q_i=1} (1-p_i)} \times \frac{\prod_{x_i=q_i=1} (1-u_i)}{\prod_{x_i=q_i=1} u_i} \times$$

$$\frac{\prod_{q_i=1} (1-p_i)}{\prod_{q_i=1} (1-u_i)}$$

constant indepent of document.

$$O(R=1|d,q) = \text{const.} \times \prod_{x_i=1=q_i} \frac{p_i}{1-p_i} \times \frac{1-u_i}{u_i}$$

$$\log(O(R=1|d,q)) = \sum_{x_i=1=q_i} \log\left(\frac{p_i}{1-p_i}\right) + \log\left(\frac{1-u_i}{u_i}\right)$$

For a query (q).

	$R=1$	$R=0$	
x_i present ($x_i=1$)	s	$df_i - s$	df_i → no. of documents in which term is present
x_i absent ($x_i=0$)	$S-s$	$N-df_i-(S-s)$	$N-df_i$
	S	$N-S$	N → Total no of document

PAGE NO.: _____
DATE: / /

$$P_i = \frac{s}{S}$$

$$U_i = \frac{df_i - s}{N - S}$$

$$C_i = \frac{P_i}{1 - P_i} \times \frac{1 - U_i}{U_i}$$

$$= \frac{s/S}{(S-s)/s} \times \frac{(N-S-df_i+s)/N-S}{(df_i - s)/N-S}$$

$$= \frac{s}{S-s} \times \frac{N-S-df_i+s}{df_i - s}$$

$S \ll N$
$s \ll df_i$ [Statistical assumption]

$$\frac{s}{S-s} \times \frac{N-df_i}{df}$$

$\frac{s}{S-s}$ represent term frequency some kind of.

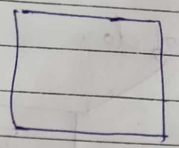
$\frac{N-df_i}{df}$ represent idf

Scanned by CamScanner

PAGE NO.:
DATE: / /

To get an idea of S_i

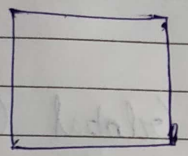
init q



V ≈ 10-100

VR: VR_i

V



V ≈ 10-100

VR : VR_i

VNR = V - VR

$$\hat{p}_i = \frac{VR_i + \left(\frac{1}{2}\right)}{VR + (1)}$$ for smoothing.

Probablistic Relevance feedback.

$$p_i^{(k+1)} = \frac{VR_i + \alpha p_i^{(k)}}{VR + \alpha}$$

$$u_i = \frac{df_i - s}{N - s}$$

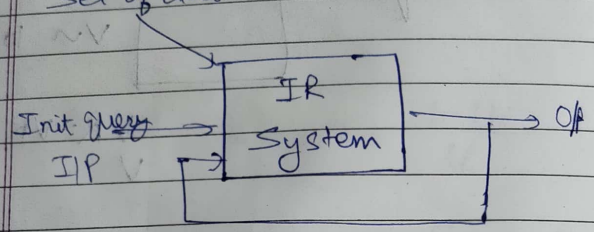
≈ $\frac{df_i}{N}$

To stop when

$|p_i^{(k+1)} - p_i^{(k)}| <$ threshold

PAGE NO.: _____
DATE: / /

Retrieval Models



Feedback
- Global (independent of query collection)
- Local

Global
- Thesaurus / Dictionary (WordNet)
- Manual.

Local
- Relevance feedback.

PAGE NO.:
DATE: / /

(VSM)

$$\vec{q}_{opt} = \arg\max_{\vec{q}} \left[sim(\vec{q}, D_r) - sim(\vec{q}, D_{nr}) \right]$$

Modify $\vec{q}$ and find out q_{opt}.

Rocchio Relevance feedback.

$$q_{mod} = \alpha \vec{q}_{mod} + \beta \times \frac{1}{|D_r|} \sum_{d_r \in D_r} \vec{d_r} - \gamma \times \frac{1}{|D_{nr}|} \times \sum_{d_{nr} \in D_{nr}} d_{nr}$$

for estimation we'll see in ~~retrieval~~ docs. ideally should be taken from whole set.

α, β, γ tuning param.

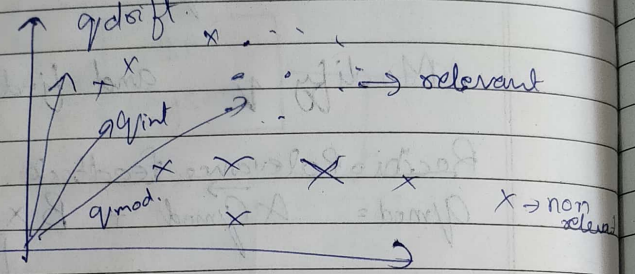
$\alpha = 1$
$\beta = 0.6 - 0.85$
$\gamma = 0 - 0.15$

Pseudo-Relevance feedback (Blind Query Expansion).

Initial 20 docs (say)
(Rank 1 — Rank 20) - do.
The rest are d_{nr}.

PAGE NO.: _____
DATE: / /

In pseudo relevance feedback
query drift = If your modified query goes to non-relevant doc side.



RF increases recall. } ⇒ MAP
also it can increase prec }

Relevance feedback

q_0 = cheap DVDs cheap CDs extremely cheap CDs

d_1 = CDs cheap software CDs.

d_2 = cheap thrills DVDs (NR)

cheap
CD
DVD
extreme
software
thrill

PAGE NO.:
DATE: / /

$$\vec{q}_{mod} = \alpha q_{mod} + \frac{\beta}{D_r} \sum_{d \in D_r} \vec{d_r} - \gamma \frac{1}{|D_{nr}|} \sum_{d_{nr} \in D_{nr}} \vec{d_{nr}}$$

	q_0		d_r		d_{nr}		
cheap	3		1		1		3.4
CD	2		2		0		3.4
DVD	1	+ 0.7	0	- 0.3	1	=	0.7
extreme	1		0		0		1
software	0		1		0		0.7
thrill	0		0		1		-0.3
							≈ 0

q_{mod}

3) Language Models

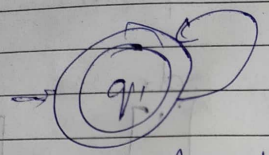
Σ = Set of Symbols = {a, b, ...}

P = Set of rules.

Σ^* = Set of strings

PAGE NO.:
DATE: / /

$P(t_1 t_2 t_3) = P(t_1)\ P(\text{continue} \mid t_1)$
$P(t_2)\ P(\text{cont} \mid t_1 t_2)$
$P(t_3)\ P(\text{stop} \mid t_1 t_2 t_3)$



A probablistic finite automata.

$P(\text{cont} \mid t_i) + P(\text{stop} \mid t_i) = 1$

$\sum P(t_i) = 1$

We can rank the document by the probablity by which a document generates a query.

Types of Lang Models.

- Unigram:- Consider a particular order.

$P_{uni}(t_1 t_2 t_3 t_4) = P(t_1) P(t_2) P(t_3) P(t_4)$

- Bigram:-

$P_{bi}(t_1 t_2 t_3 t_4) = P(t_1) P(t_2 \mid t_1) P(t_3 \mid t_2)$
$P(t_4 \mid t_3)$

Each document is a model a universe for news.

Probability of document in universe of language.

Multinomial:- considers every permutation

$$P(d) = \frac{|d|!}{tf_1 d!\, tf_2 d! \ldots tf_n d!} \times P(t_1)^{tf_1} \times P(t_2)^{tf_2}$$

Query Likelihood Model.

$$P(d|q) = P(q|d) \times P(d)/P(q)$$

$$P(q|M_d) = K_q \prod_{t \in V} P(t|M_d)^{tf_{t,d}}$$

where $K_q = |q|!/(tf_1 d!) \times (tf_2 d!)$

Each document is a model a universe for now.

$$P(d_i|q) = \frac{P(q|d_i) \times P(d_i)}{P(q)} \propto P(q|d_i) \times P(d_i)$$

Estimating Probabilities :-

Maximum likelihood estimate :-

$$\hat{P}(q|M_d) = \prod_{t \in q} \hat{P}_{mle}(t|M_d) = \prod_{t \in q} \frac{tf_{t,d}}{|d|}$$

Smoothing $tf_d = 0$ then

$$\hat{P}(t|M_d) \leq cf_t/T$$

Scanned by CamScanner

PAGE NO.:
DATE: / /

$$\hat{P}(t|d) = \lambda P_{mle}(t|M_d) + (1-\lambda) P_{mle}(t|M_c)$$

Where $0 < \lambda < 1$ & M_c is the model built from entire document collection.

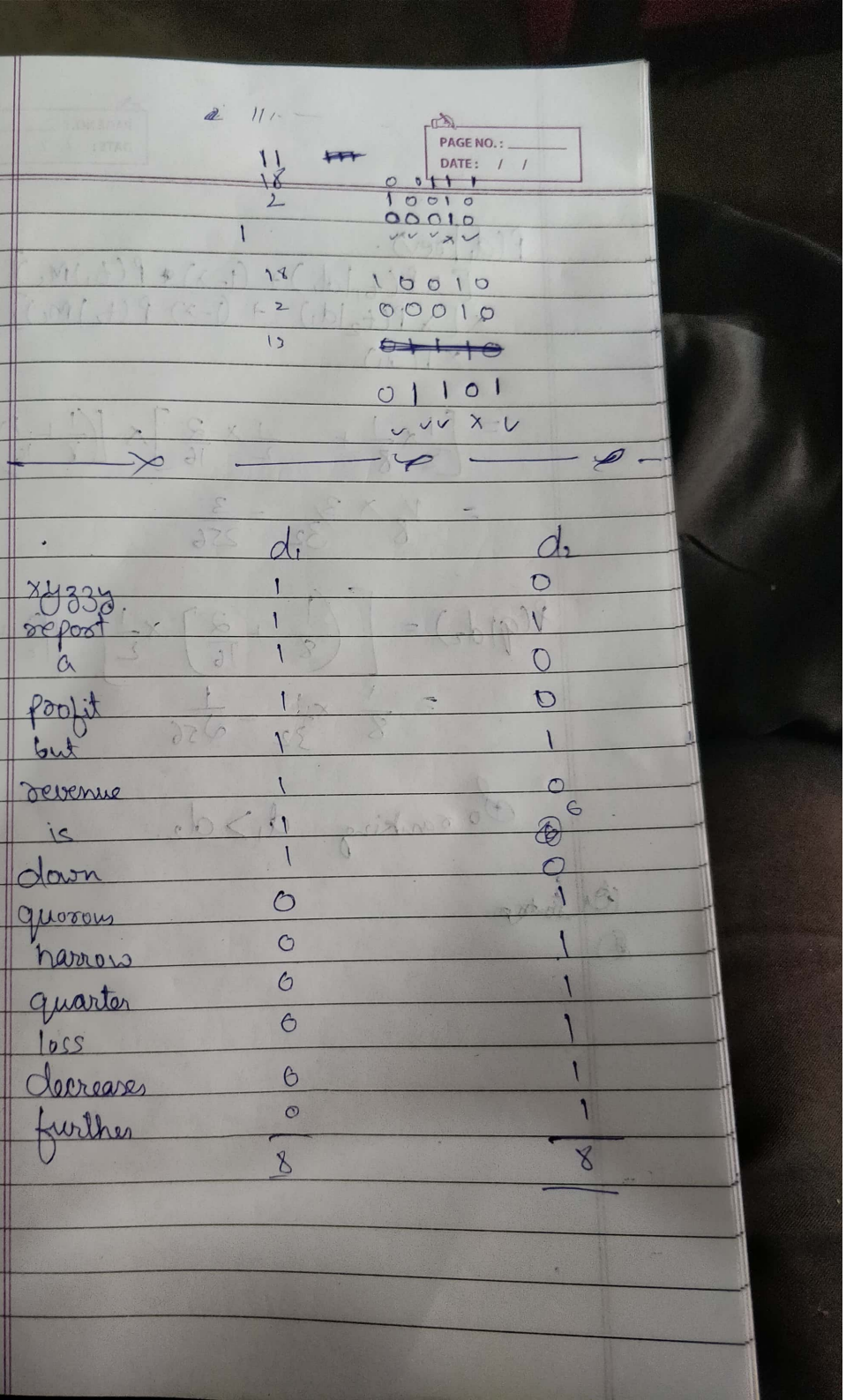
Bayesian Smoothening

$$\hat{P}(t|d) = \frac{tf_{t,d} + \alpha \hat{P}(t|M_c)}{L_d + \alpha}$$

- Small lamda or large alpha means more smoothing.
- Low smoothing good for short queries, high smoothing for large queries.

Example

d_1 = Xyzzy reports a profit but revenue is down
d_2 = Quorus narrows quarter loss but revenue decreases further



PAGE NO.:
DATE: / /

	d_1	d_2
xyzzy	1	0
report	1	1
a	1	0
profit	1	0
but	1	1
revenue	1	0
is	1	0
down	1	0
quarter	0	1
narrow	0	1
quarter	0	1
loss	0	1
decreases	0	1
further	0	1
	8	8

PAGE NO.:
DATE: / /

$$P(d_1|rev) = [\lambda P(t_1|d_1) + (1-\lambda) P(t_1|M_c)] \times [\lambda P(t_2|d_1) + (1-\lambda) P(t_2|M_c)] \times P(d_1)$$

$$= \left[\frac{1}{2} \times \frac{1}{8} + \frac{1}{2} \times \frac{2}{16}\right] \times \left[\left(\frac{1}{8} + \frac{1}{16}\right) \times \frac{1}{2}\right]$$

$$= \frac{1}{8} \times \frac{3}{32} = \frac{3}{256}$$

$$P(q|d_2) = \left[\left(\frac{1}{8} + \frac{2}{16}\right) \times \frac{1}{2}\right] \times \left[\left(\frac{0}{8} + \frac{1}{16}\right) \times \frac{1}{2}\right]$$

$$= \frac{1}{8} \times \frac{1}{32} = \frac{1}{256}$$

So ranking $d_1 > d_2$.

(1) Smoothing.

(2)