IR system for E-commerce platform.

- search for products
- Whether relevant items are being recommended in the context of searched item.

- Relevant products being shown.
- are the searches getting materialized to money ($, ₹)
- Repeat buyers (in days/weeks/months)
  - Dwell time.

Evaluation of IR systems.

→ How fast can we index?
  → # of documents / hour
  → 10K docs are added per day
→ How fast can we search?
  CPU speed — query size
  → latency
→ Whether "related" items are featuring?

None of these are the correct metrics for the judgement at the PRIMARY SPOT.

query → information need → IR system should be judged on the basis of this need.

50K sample queries.

5M products.

5M products

50K sample queries

relevance judgement

Binary

0/1

graduated

fair (0)

Good (1)

Excellent (2)

Each judgement takes 2.5 secs. for a human.

✓ 10^{11} secs ← 3000 + years.

✓ \$10 per hour to a person. \$300M

Cyril Cleverdon → Cranfield experiments

Make some generic assumptions about an IR systems.

A test collection:

↳ Three elements

(1) A benchmark collection of documents

(2) A benchmark suite of queries.

(3) An assessment whether a benchmark ~~query~~ document is relevant / not relevant to a benchmark query (human judges).

benchmark docs: products

benchmark queries: ?

Judgements whether a product is relevant to a query.

Relevance → as per the user need/intent
&
NOT the query itself

(Information need – "My swimming pool bottom is becoming black & needs to be cleaned!"

Query → "pool cleaner".

crowd source the judgement task.

AMT → Amazon Mechanical Turk platform.

<u>Benchmark</u>
<u>query</u>

- query should be appropriate to the corpus.
- query ≡ actual info need of the user.
- random query terms are not a good choice
- What is a good choice?

"query logs" → Internet capital these days

Binary assessments.

Precision: No. of relevant docs out of all the retrieved docs.

Recall. No. of relevant docs out of all the relevant docs.

recall.

	Relevant	Not relevant
Retrieved	true positive (tp)	false positive (fp) → P
Not retrieved	false negative (fn)	true negative (tn)

$$\text{Precision} = \frac{tp}{(tp + fp)}$$

$$\text{Recall} = \frac{tp}{(tp + fn)}$$

Why care about precision & recall?

When is precision important

- precision medicine "~~safte~~ safety critical system"

When is recall important

- covid testing.
- covid vaccination. (flu shot)

" CV screening"

Rank based measures:

Precision@K
Mean Average Precision (MAP)
Mean Reciprocal Rank (MRR).

Graduated Relevance scores:

Normalized Discounted Cumulative Gain (NDCG).

Set a rank threshold = K.
% of relevant docs in top K.
Ignores all documents that are ranked below K.

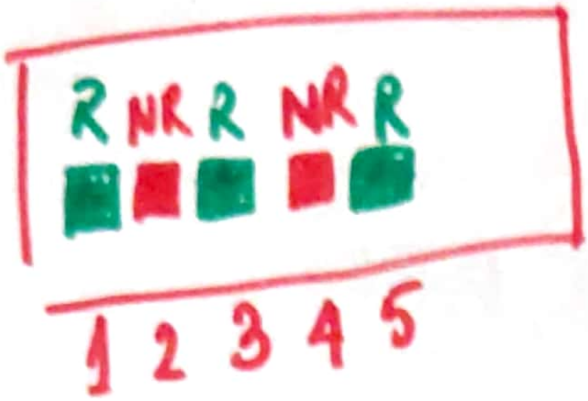
Avg(Pr@1, Pr@3, Pr@5)

$$\frac{1}{3}\left(1+\frac{2}{3}+\frac{3}{5}\right) \simeq 0.76.$$

Estimate whether relevant items are at "<u>better</u>" rank positions.

<u>MAP.</u>

Mean of avg precisions across different queries.



$Pr@3 = \frac{2}{3}$

$Pr@5 = \frac{3}{5}$

Average Precision:

the rank positions of the <u>relevant</u> docs are at K_1, K_2, K_3 $\ldots, K_r$.

(Compute $Pr@K$ for each $K_1, K_2, K_3 \ldots K_r$)

Average

■ ■ ■ ■ ■ = relevant docs for query 1.

	R	NR	R	NR	NR	R	NR	NR	R	R
	■	□	■	□	□	■	□	□	■	■
Precision	1	0.5	0.66	0.5	0.4	. . .				
Recall	0.2	0.2	0.4	0.4	. . .					

■ ■ ■ = relevant docs for query 2.

	NR	R	NR	NR	R	NR	R	NR	NR	NR
	□	■	□	□	■	□	■	□	□	□
Precision	0	0.5	0.33	. . .						
Recall	0	0.33	0.33	0.33	0.67	. . .				

AP for q1 = 0.62

AP for q2 = 0.44

MAP = (0.62 + 0.44)/2 = 0.53.

Mean Reciprocal Rank.

rank R of the first relevant document in the list

$$RR = \frac{1}{R}.$$

MRR → RR across different queries.

RR@K.

↳ inspect upto rank K.

Graded relevance.

→ fair } — 1
→ good — 2
→ Ex — 3

→ NR → 0.

→

Highly relevant docs should be at the top.

↳ if a document comes very low in the rank list then it isn't so important.

Discounted gain.

Discount the gain

$$\frac{1}{\log_2(\text{rank})}$$

↳ discount at rank

$4 = \frac{1}{2}$

$8 = \frac{1}{3}$

⋮

$[0 \ldots K]$, $K >= 2$.

relevance based ratings of 'n' docs

$r_1, r_2, \ldots, r_n$ (in ranked order.)

$CG = r_1 + r_2 + r_3 \ldots \ldots + r_n$

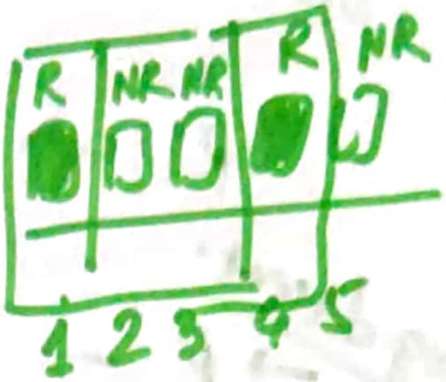
$DCG = r_1 + r_2/\log 2 + r_3/\log 3 + r_4/\log 4 + \ldots \ldots + r_n/\log n$

$$\underline{\underline{DCG}} = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$$

↳ normalization

$\boxed{NDCG}$

Precision at a rank. (K)



$Pr@3 = \frac{1}{3}$

Avg. Precision

AP.

Pr@K → only those k's rank positions where we have a relevant doc

$Avg(Pr@1, Pr@4) = \frac{1}{2}\left(\frac{1}{1} + \frac{2}{4}\right)$

q_1 → AP

q_2 → AP

q_m →

MAP. (mean avg precision).

MRR. → RR = inverse of the rank of the first relevant doc → (R) = $\frac{1}{R}$.

Σ Mean of RR over multiple queries

Discounted cumulative gain. (DCG).

R → R_1. (full score)

log discount. → R → R_4 ($\frac{\text{score}}{\log_2(\text{rank})}$ ← discounted)

$$DCG = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$$

DCG@p. (upto rank p).

Ideal DCG.

[0-3] ←

IR. → 3, 2, 3, 0, 0, 1, 2, 2, 3, 0.

Sort → 3, 3, 3, 2, 2, 1, 0, 0, 0.

Actual DCG: 3, 5, 6.89, 6.89 . . . .

Ideal DCG 3, 6, 7.89, 8.89 . . . .

NDCG : 1, 0.83, 0.87, 0.76 . . . .

1 2 3 4

Intrinsic eval technique.

Extrinsic eval technique. (task based evaluation).

User judgements for that task at your disposal. → X

inform another task (T_2)

unevalulated task with no user judgement

T_1 output

user judgement is there

Based on this addl. info → does T_2's performance improve?

Improve the recall of an IR system?

- Relevance feedback.
- Query expansion.

term.

q: [aircraft].

- [plane]
- [airplane].
-
-
-

Can we leverage this info. - to improve the overall performance?

- RF ← local method. some feedback from the user & change the query.
- QE
  - ↳ Global method.
    - ↳ Dictionary/ Thesaurus. (to expand your query)

Relevance feedback.

$u \xrightarrow{\text{fires}} q_0$ (initial query).

SE → returns some documents
(search engine) related to q_0
(relevant to)

feedback [u → marks some of the retrieved
docs as R OR NR.

SE → computes a new representation
of the info.

A q_1 is run on SE
Expectation: recall should improve.

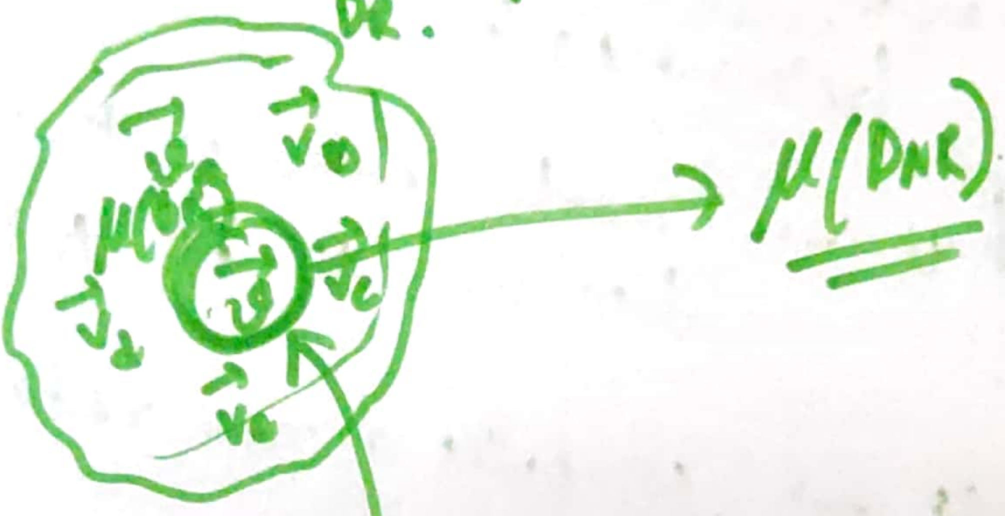
Rocchio's Technique.

- Initial query q_0
- u ← relevant documents (D_R)
  non-relevant documents (D_{NR})

→ having a vector ($\vec{v}_{opt}$) that maximally
separates D_R from D_{NR} in your vector space.

cosine sim.

$$\vec{v}_{opt} = \underset{\vec{v}}{\arg\max} \left[sim(\vec{v}, \underline{D_R}) - sim(\vec{v}, \underline{D_{NR}}) \right].$$



$\vec{\mu}(D_R)$ [centroid of the vectors in D_R].

$$= \frac{1}{|D_R|} \sum_{d_j \in D_R} d_j$$

Compute: $\vec{\mu}(D_{NR})$. $\rightarrow$ $\vec{v}_{opt} = \underline{\vec{\mu}(D_R) - \vec{\mu}(D_{NR})}$

Shift q_0 with $\vec{v}_{opt}$ to get the optimal query.

$$\vec{q}_{opt} = \boxed{\vec{\mu}(D_R)} + \left[\vec{\mu}(D_R) - \vec{\mu}(D_{NR}) \right]$$