Information Retrieval: Evaluation

Information Retrieval: Evaluation

- **How to systematically evaluate/compare different IR methods**
  - which variant of TF*IDF performs best?
  - does stemming help? How about stopword removal?
- We need a **document collection**, a **set of topics** and **relevance assessments**, and **effectiveness measures**
- IR evaluation has been driven a lot by benchmark initiatives (e.g., TREC http://trec.nist.gov)

Documents, Topics, and Relevance Assessments

- **Document collection** (e.g., a collection of newspaper articles)
- Topics are **descriptions of concrete information needs**

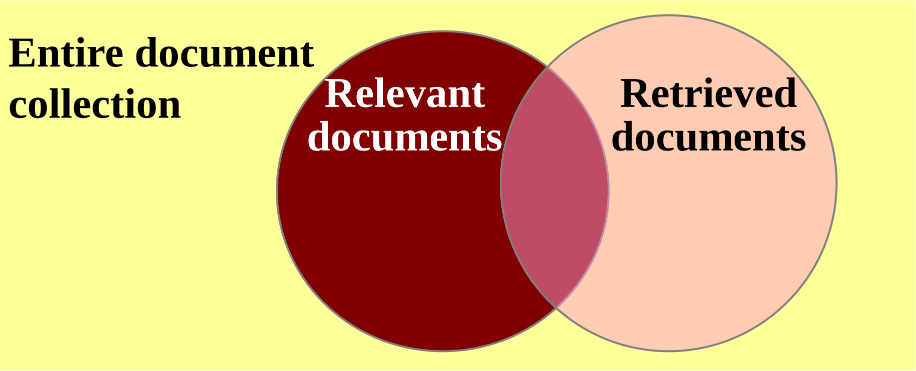
```
<num> Number: 310
<title> Radio Waves and Brain Cancer

<desc> Description:
Evidence that radio waves from radio towers or car phones affect
brain cancer occurrence.

<narr> Narrative:
Persons living near radio towers and more recently persons using
car phones have been diagnosed with brain cancer.  The argument
rages regarding the direct association of one with the other.
The incidence of cancer among the groups cited is considered…
```

- **Queries are derived from topics** (e.g., using only the title)
- **Relevance assessments** are **(topic, document, label) tuples** with binary (1 : relevant, 0 : irrelevant) or graded labels often determined by trained experts
- Parameter tuning mandates splitting into training & test topics

Precision and Recall



	retrieved	not retrieved
irrelevant	retrieved & irrelevant	Not retrieved & irrelevant
relevant	retrieved & relevant	not retrieved but relevant

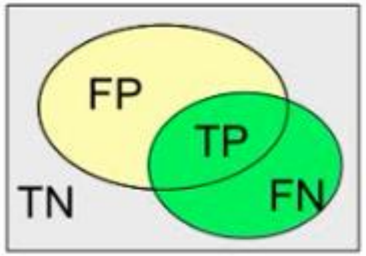
$$recall = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ relevant\ documents}$$

$$precision = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ documents\ retrieved}$$

Precision and Recall

- Precision
  - The ability to retrieve top-ranked documents that are mostly relevant.
- Recall
  - The ability of the search to find ***all*** of the relevant items in the corpus.

Why not accuracy?



- Retrieval ... a kind of classification
  - document → {relevant, non-relevant}
  - standard measure: $$Accuracy = \frac{correct}{total} = \frac{TP + TN}{N}$$
  - or use Error = 1 - Accuracy
- Meaningless:
  - accuracy 99.99% for any search algorithm
    - for any query, almost all documents are non-relevant
    - often best strategy is to retrieve nothing:

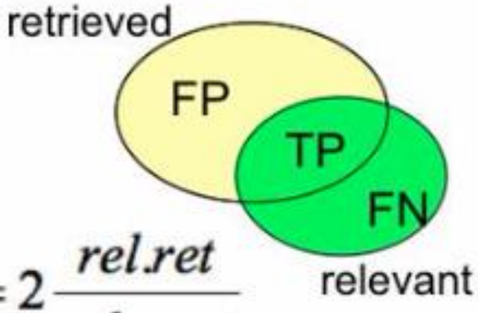
$$\underbrace{\overbrace{\text{NNRNRNR}}^{\text{retrieved}}\overbrace{\text{NNNNNNNNNNNNNNNNNNNNNNN}}^{\text{not retrieved}}}_{\text{retrieve nothing}}$$

F-measure

- A variant of accuracy not affected by negatives
  - single-value measure (compare, tune systems)
- Harmonic mean of P and R: $F_\beta = \frac{(\beta^2+1) \cdot P \cdot R}{\beta^2 P + R}$
  - β … relative importance of recall and precision
  - popular setting: β=1, which gives: $F_1 = \frac{2PR}{P+R}$
  - heavily penalizes small values of P and R
- Geometric interpretation:
  - %overlap between relevant, retrieved

F-measure

- A variant of accuracy not affected by negatives
  - single-value measure (compare, tune systems)
- Harmonic mean of P and R: $F_\beta = \frac{(\beta^2+1)\cdot P\cdot R}{\beta^2 P + R}$
  - β … relative importance of recall and precision
  - popular setting: β=1, which gives: $F_1 = \frac{2PR}{P+R}$
  - heavily penalizes small values of P and R
- Geometric interpretation:
  - %overlap between relevant, retrieved



$$F_1 = \frac{2PR}{P+R} = 2\left(\frac{1}{P}+\frac{1}{R}\right)^{-1} = 2\left(\frac{TP+FP}{TP}+\frac{TP+FN}{TP}\right)^{-1} = 2\frac{rel.ret}{rel+ret}$$

aka Dice coefficient

Rank-Based Measures

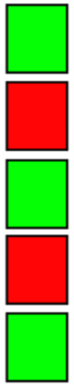
- Binary relevance
  - Precision@K (P@K)
  - Mean Average Precision (MAP)
  - Mean Reciprocal Rank (MRR)

- Multiple levels of relevance
  - Normalized Discounted Cumulative Gain (NDCG)

Binary relevance

Precision@K

- Set a rank threshold K
- Compute % relevant in top K
- Ignores documents ranked lower than K
- Ex:
  - Prec@3 of 2/3
  - Prec@4 of 2/4
  - Prec@5 of 3/5



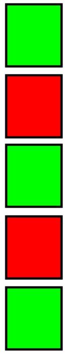
- In similar fashion we have Recall@K

Mean Average Precision

- Consider rank position of each ***relevant*** doc
  - $K_1, K_2, \ldots K_R$

- Compute Precision@K for each $K_1, K_2, \ldots K_R$

- Average precision = average of P@K

- Ex:  has AvgPrec of

AP

$$\frac{1}{3} \cdot \left(\frac{1}{1} + \frac{2}{3} + \frac{3}{5} \right) \approx 0.76$$

- MAP is Average Precision across multiple queries/rankings

Mean Average Precision

- **Average precision** (AP) averages over retrieved relevant results (=computed Precision at all "Recall levels")
  - Let $\{d_1, \ldots, d_{mj}\}$ be the set of relevant results for the query q_j
  - Let R_{jk} be the set of ranked retrieval results for the query q_j from top until you get to the relevant result d_k

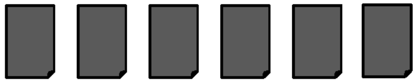
$$\mathrm{AP}(q_j) = \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk})$$

If a relevant doc is not retrieved at all, the Precision(…) is considered 0

- **Mean average precision** (MAP) averages over multiple queries

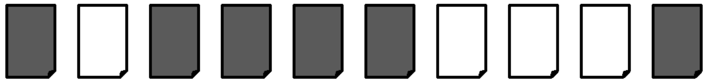
$$\mathrm{MAP}(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \mathrm{AP}(q_j)$$

Average Precision



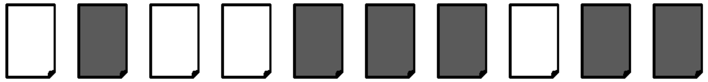
= the relevant documents

Ranking #1



Recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	0.83	1.0
Precision	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.6

Ranking #2

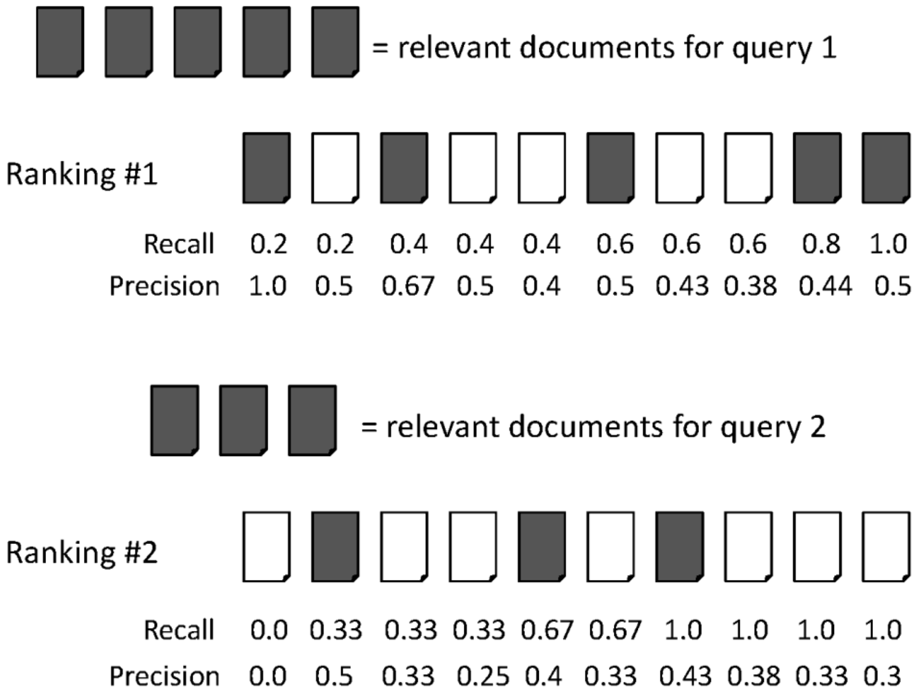


Recall	0.0	0.17	0.17	0.17	0.33	0.5	0.67	0.67	0.83	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.5	0.57	0.5	0.56	0.6

$$\text{Ranking \#1:}\ (1.0 + 0.67 + 0.75 + 0.8 + 0.83 + 0.6)/6 = 0.78$$

$$\text{Ranking \#2:}\ (0.5 + 0.4 + 0.5 + 0.57 + 0.56 + 0.6)/6 = 0.52$$

Mean Average Precision



$$\textit{average precision query 1} = (1.0 + 0.67 + 0.5 + 0.44 + 0.5)/5 = 0.62$$
$$\textit{average precision query 2} = (0.5 + 0.4 + 0.43)/3 = 0.44$$

$$\textit{mean average precision} = (0.62 + 0.44)/2 = 0.53$$

Mean average precision

- If a relevant document never gets retrieved, we assume the precision corresponding to that relevant doc to be zero
- MAP is macro-averaging: each query counts equally
- Now perhaps most commonly used measure in research papers
- MAP assumes user is interested in finding many relevant documents for each query
- MAP requires many relevance judgments in text collection

What if the results are not in a list?

- Suppose there's only one Relevant Document
- Scenarios:
  - known-item search
  - navigational queries
  - looking for a fact
- Search duration ~ Rank of the answer
  - measures a user's effort

Mean Reciprocal Rank

- Consider rank position, K, of first relevant doc
  - Could be – only clicked doc
- Reciprocal Rank score = $\frac{1}{K}$ RR Score
- MRR is the mean RR across multiple queries

Multiple levels of relevance

Discounted Cumulative Gain

- Popular measure for evaluating web search and related tasks

- Two assumptions:
  - Highly relevant documents are more useful than marginally relevant documents
  - the lower the ranked position of a relevant document, the less useful it is for the user, since it is less likely to be examined

Cumulative Gain

- With graded relevance judgments, we can compute the *gain* at each rank.
- **Cumulative Gain** at rank n:

$$CG_n = \sum_{i=1}^{n} rel_i$$

(Where rel_i is the graded relevance of the document at position i)

n	doc #	relevance (gain)	CG_n
1	588	1.0	1.0
2	589	0.6	1.6
3	576	0.0	1.6
4	590	0.8	2.4
5	986	0.0	2.4
6	592	1.0	3.4
7	984	0.0	3.4
8	988	0.0	3.4
9	578	0.0	3.4
10	985	0.0	3.4
11	103	0.0	3.4
12	591	0.0	3.4
13	772	0.2	3.6
14	990	0.0	3.6

Discounting Based on Position

- Users care more about high-ranked documents, so we **discount** results by *1/log₂(rank)*

- **Discounted Cumulative Gain:**

$$\boxed{\mathrm{DCG}_n = rel_1 + \sum_{i=2}^{n} \frac{rel_i}{\log_2 i}}$$

n	doc #	rel (gain)	CG_n	$\log_n$	DCG_n
1	588	1.0	1.0	-	1.00
2	589	0.6	1.6	1.00	1.60
3	576	0.0	1.6	1.58	1.60
4	590	0.8	2.4	2.00	2.00
5	986	0.0	2.4	2.32	2.00
6	592	1.0	3.4	2.58	2.39
7	984	0.0	3.4	2.81	2.39
8	988	0.0	3.4	3.00	2.39
9	578	0.0	3.4	3.17	2.39
10	985	0.0	3.4	3.32	2.39
11	103	0.0	3.4	3.46	2.39
12	591	0.0	3.4	3.58	2.39
13	772	0.2	3.6	3.70	2.44
14	990	0.0	3.6	3.81	2.44

Normalized Discounted Cumulative Gain (NDCG)

- To compare DCGs, normalize values so that an *ideal ranking* would have a **Normalized DCG** of 1.0
- Ideal ranking:

n	doc #	rel (gain)	CG_n	$\log_n$	DCG_n
1	588	1.0	1.0	0.00	1.00
2	589	0.6	1.6	1.00	1.60
3	576	0.0	1.6	1.58	1.60
4	590	0.8	2.4	2.00	2.00
5	986	0.0	2.4	2.32	2.00
6	592	1.0	3.4	2.58	2.39
7	984	0.0	3.4	2.81	2.39
8	988	0.0	3.4	3.00	2.39
9	578	0.0	3.4	3.17	2.39
10	985	0.0	3.4	3.32	2.39
11	103	0.0	3.4	3.46	2.39
12	591	0.0	3.4	3.58	2.39
13	772	0.2	3.6	3.70	2.44
14	990	0.0	3.6	3.81	2.44

n	doc #	rel (gain)	CG_n	$\log_n$	$IDCG_n$
1	588	1.0	1.0	0.00	1.00
2	592	1.0	2.0	1.00	2.00
3	590	0.8	2.8	1.58	2.50
4	589	0.6	3.4	2.00	2.80
5	772	0.2	3.6	2.32	2.89
6	576	0.0	3.6	2.58	2.89
7	986	0.0	3.6	2.81	2.89
8	984	0.0	3.6	3.00	2.89
9	988	0.0	3.6	3.17	2.89
10	578	0.0	3.6	3.32	2.89
11	985	0.0	3.6	3.46	2.89
12	103	0.0	3.6	3.58	2.89
13	591	0.0	3.6	3.70	2.89
14	990	0.0	3.6	3.81	2.89

Normalized Discounted Cumulative Gain (NDCG)

- Normalize by DCG of the ideal ranking:

$$\mathrm{NDCG}_n = \frac{DCG_n}{IDCG_n}$$

- NDCG ≤ 1 at all ranks
- NDCG is comparable across different queries

n	doc #	rel (gain)	DCG_n	$IDCG_n$	$NDCG_n$
1	588	1.0	1.00	1.00	**1.00**
2	589	0.6	1.60	2.00	**0.80**
3	576	0.0	1.60	2.50	**0.64**
4	590	0.8	2.00	2.80	**0.71**
5	986	0.0	2.00	2.89	**0.69**
6	592	1.0	2.39	2.89	**0.83**
7	984	0.0	2.39	2.89	**0.83**
8	988	0.0	2.39	2.89	**0.83**
9	578	0.0	2.39	2.89	**0.83**
10	985	0.0	2.39	2.89	**0.83**
11	103	0.0	2.39	2.89	**0.83**
12	591	0.0	2.39	2.89	**0.83**
13	772	0.2	2.44	2.89	**0.84**
14	990	0.0	2.44	2.89	**0.84**