

# Information Retrieval

Evaluation Methods

# Measures for a search engine

---

- How fast does it index?
  - Number of documents/hour
  - Incremental indexing
- How fast does it search?
  - Latency as a function of index size
- Does it recommend related products?
- This is all good, but it says nothing about the *quality* of the search
  - You want the users to be happy with the search experience

# How do you tell if users are happy?

---

- Search returns products relevant to users
  - How do you assess this at scale?
- Search results get clicked a lot
  - Misleading titles/summaries can cause users to click
- Users buy after using the search engine
  - Or, users spend a lot of \$ after using the search engine
- Repeat visitors/buyers
  - Do users leave soon after searching?
  - Do they come back within a week/month/... ?

# Measuring relevance

---

- Three elements:
  1. A benchmark document collection
  2. A benchmark suite of queries
  3. An assessment of either Relevant or Nonrelevant for each query and each document

# Evaluating an IR system

---

- Note: **user need** is translated into a **query**
- Relevance is assessed relative to the **user need**, *not* the **query**
- E.g., Information need: *My swimming pool bottom is becoming black and needs to be cleaned.*
- Query: ***pool cleaner***
- Assess whether the doc addresses the underlying need, not whether it has these words

# Precision and Recall

- **Binary assessments**

**Precision:** fraction of retrieved docs that are relevant =  $P(\text{relevant} | \text{retrieved})$

**Recall:** fraction of relevant docs that are retrieved =  $P(\text{retrieved} | \text{relevant})$

	Relevant	Nonrelevant
Retrieved	tp	fp
Not Retrieved	fn	tn

- Precision  $P = tp / (tp + fp)$
- Recall  $R = tp / (tp + fn)$

# 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)

# Precision@K

- Set a rank threshold K
- Compute % relevant in top K
- Ignores documents ranked lower than K

■ Ex:

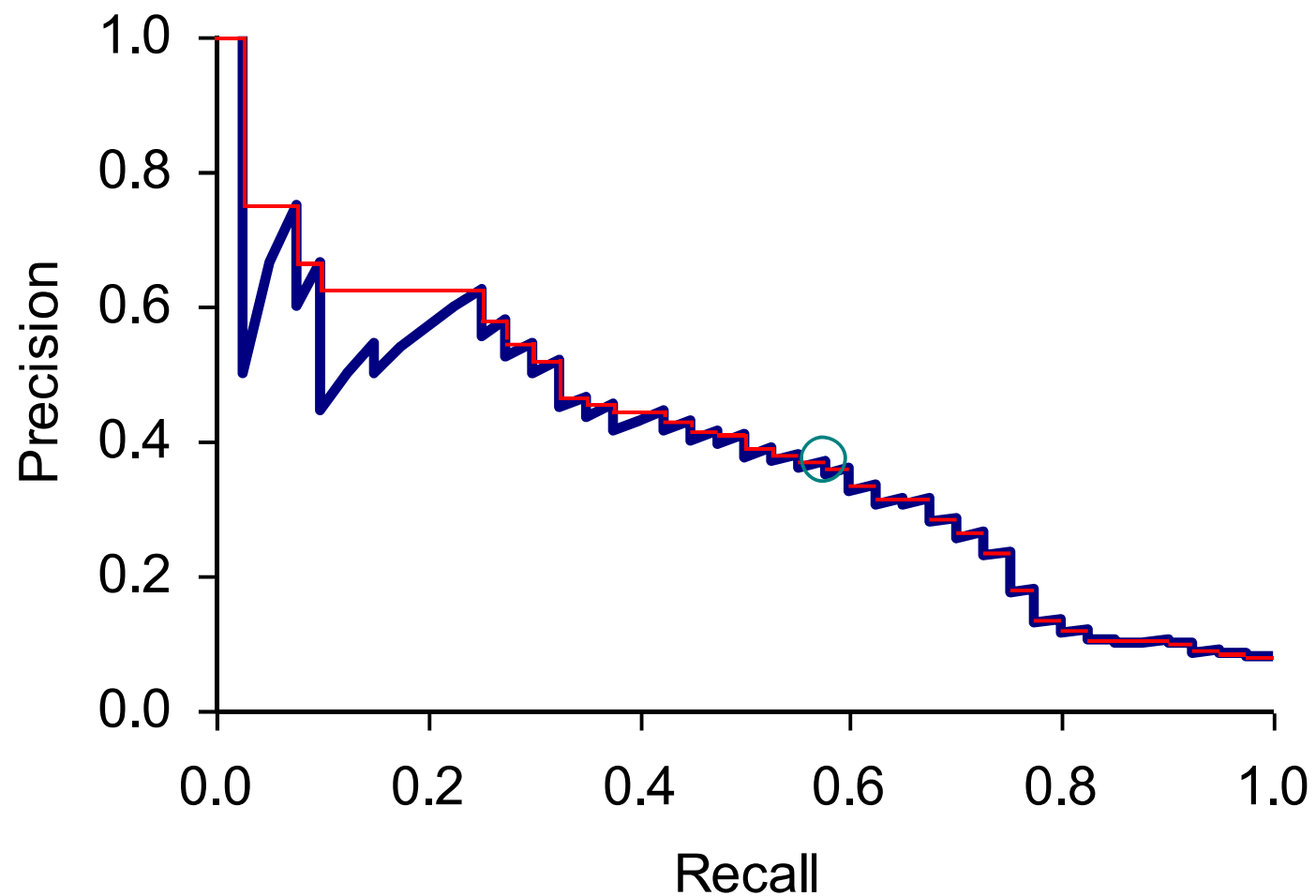
- Prec@3 of 2/3
- Prec@4 of 2/4
- Prec@5 of 3/5



- In similar fashion we have Recall@K



# A precision-recall curve



# Mean Average Precision











- Consider rank position of each **relevant** doc
  - $K_1, K_2, \dots K_R$
- Compute Precision@K for each  $K_1, K_2, \dots K_R$
- Average precision = average of P@K

- Ex:  has AvgPrec of  $\frac{1}{3} \cdot \left( \frac{1}{1} + \frac{2}{3} + \frac{3}{5} \right) \approx 0.76$











- MAP is Average Precision across multiple queries/rankings

# 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











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


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

# MAP











 = relevant documents for query 1

Ranking #1

										
Recall	0.2	0.2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	1.0
Precision	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5

 = relevant documents for query 2

Ranking #2

										
Recall	0.0	0.33	0.33	0.33	0.67	0.67	1.0	1.0	1.0	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.33	0.43	0.38	0.33	0.3

$$\text{average precision query 1} = (1.0 + 0.67 + 0.5 + 0.44 + 0.5)/5 = 0.62$$

$$\text{average precision query 2} = (0.5 + 0.4 + 0.43)/3 = 0.44$$

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

# 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
- Good for web search?
- MAP assumes user is interested in finding many relevant documents for each query
- MAP requires many relevance judgments in text collection



# **BEYOND BINARY 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

# Discounted Cumulative Gain

---

- Uses *graded relevance* as a measure of usefulness, or *gain*, from examining a document
- Gain is accumulated starting at the top of the ranking and may be reduced, or *discounted*, at lower ranks
- Typical discount is  $1/\log(\text{rank})$ 
  - With base 2, the discount at rank 4 is  $1/2$ , and at rank 8 it is  $1/3$



# Summarize a Ranking: DCG

---

- What if relevance judgments are in a scale of  $[0, r]$ ?  $r > 2$
- Cumulative Gain (CG) at rank  $n$ 
  - Let the ratings of the  $n$  documents be  $r_1, r_2, \dots, r_n$  (in ranked order)
  - $CG = r_1 + r_2 + \dots + r_n$
- Discounted Cumulative Gain (DCG) at rank  $n$ 
  - $DCG = r_1 + r_2 / \log_2 2 + r_3 / \log_2 3 + \dots + r_n / \log_2 n$ 
    - We may use any base for the logarithm

# Discounted Cumulative Gain

- *DCG* is the total gain accumulated at a particular rank  $p$ :

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

- Alternative formulation:

$$DCG_p = \sum_{i=1}^p \frac{2^{rel_i} - 1}{\log(1+i)}$$

- used by some web search companies
- emphasis on retrieving highly relevant documents

# DCG Example

---

- 10 ranked documents judged on 0–3 relevance scale:  
3, 2, 3, 0, 0, 1, 2, 2, 3, 0
- discounted gain:  
 $3, 2/1, 3/1.59, 0, 0, 1/2.59, 2/2.81, 2/3, 3/3.17, 0$   
 $= 3, 2, 1.89, 0, 0, 0.39, 0.71, 0.67, 0.95, 0$
- DCG:  
3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61

# NDCG for summarizing rankings

---

- Normalized Discounted Cumulative Gain (NDCG) at rank  $n$ 
  - Normalize DCG at rank  $n$  by the DCG value at rank  $n$  of the ideal ranking
  - The ideal ranking would first return the documents with the highest relevance level, then the next highest relevance level, etc
- Normalization useful for contrasting queries with varying numbers of relevant results
- NDCG is now quite popular in evaluating Web search

# NDCG - Example

4 documents:  $d_1, d_2, d_3, d_4$

i	Ground Truth		Ranking Function <sub>1</sub>		Ranking Function <sub>2</sub>	
	Document Order	r <sub>i</sub>	Document Order	r <sub>i</sub>	Document Order	r <sub>i</sub>
1	d4	2	d3	2	d3	2
2	d3	2	d4	2	d2	1
3	d2	1	d2	1	d4	2
4	d1	0	d1	0	d1	0
	NDCG <sub>GT</sub> =1.00		NDCG <sub>RF1</sub> =1.00		NDCG <sub>RF2</sub> =0.9203	

$$DCG_{GT} = 2 + \left( \frac{2}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.6309$$

$$DCG_{RF1} = 2 + \left( \frac{2}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.6309$$

$$DCG_{RF2} = 2 + \left( \frac{1}{\log_2 2} + \frac{2}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.2619$$

$$MaxDCG = DCG_{GT} = 4.6309$$

# Mean Reciprocal Rank

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