Search Evaluation

Tao Yang

CS293S Fall 2020

Slides partially based on text book [CMS] [MRS]

Table of Content

Difficulties in Evaluating IR Systems

- **Effectiveness is related to the *relevancy* of matched items.**
  - Relevancy is not typically binary but continuous.
  - Relevancy, from a human standpoint, is:
    - Subjective/cognitive: Depends upon user's judgment, human perception and behavior
  - Situational and dynamic:
    - Relates to user's current needs. Change over time.
    - CMU. US Open
- **Measure happiness of users**
  - Web engine: A user finds what they want and uses again
    - Measure rate of return users
  - eCommerce site: user finds what they want and make a purchase
    - Measure time to purchase, or fraction of searchers who become buyers?

Aspects of Search Quality

- **Relevancy**
- **Freshness& coverage**
  - Latency from creation of a document to time in the online index. (Speed of discovery and indexing)
  - Size of database in covering data coverage
- **User effort and result presentation**
  - Work required from the user in formulating queries, conducting the search
  - Expressiveness of query language
  - Influence of search output format on the user's ability to utilize the retrieved materials.

System Aspects of Evaluation

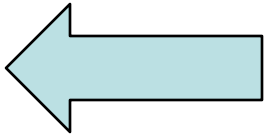
- **Response time:**
  - Time interval between receipt of a user query and the presentation of system responses.
  - Average response time
    - at different traffic levels (queries/second)
    - When # of machines changes, the size of database changes, and there is a failure of machines
- **Throughputs**
  - Maximum number of queries/second that can be handled
    - without dropping user queries
    - Or meet Service Level Agreement (SLA)
      - For example, 99% of queries need to be completed within a second.
  - How does it vary when the size of database changes

System Aspects of Evaluation

- **Others**
  - Time from crawling to online serving.
  - Percentage of results served from cache
  - Stability: number of abnormal response spikes per day or per week.
  - Fault tolerance: number of failures that can be handled.
  - Cost: number of machines needed to handle
    - different traffic levels
    - host a DB with different sizes

Table of Content

- **Search Engine Evaluation**
- **Metrics for relevancy**
  - Precision/recall
  - F-measure
  - MAP
  - NDCG
- **Creating Test Collections for IR Evaluation**
  - A/B testing



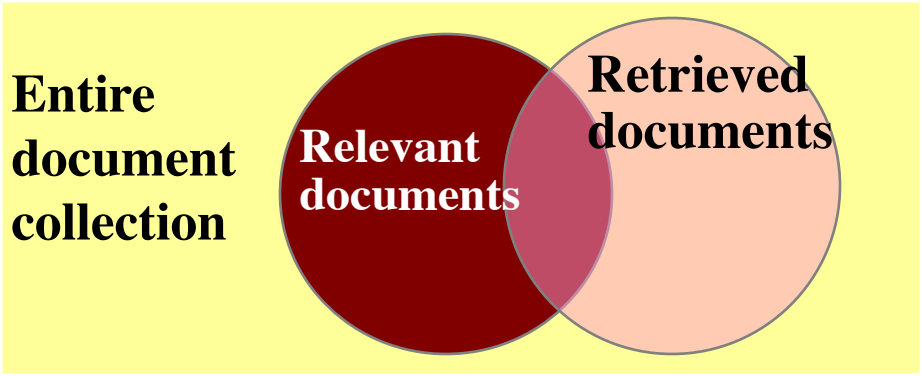
Unranked retrieval evaluation: Precision and Recall

- **Precision: fraction of retrieved docs that are relevant = P(relevant|retrieved)**
- **Recall: fraction of relevant docs that are retrieved = P(retrieved|relevant)**

	Relevant	Not Relevant
Retrieved	tp (True positive)	fp (false positive)
Not Retrieved	fn (false negative)	tn (true negative)

- **Precision P = tp/(tp + fp)** Row-wise
- **Recall R = tp/(tp + fn)** Column-wise

Precision and Recall: Another View



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

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

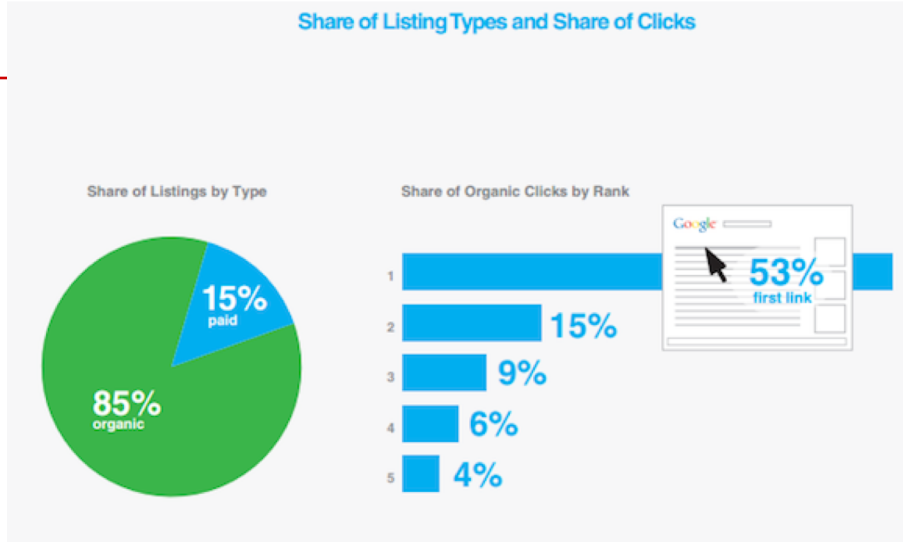
$$precision = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ documents\ retrieved}$$

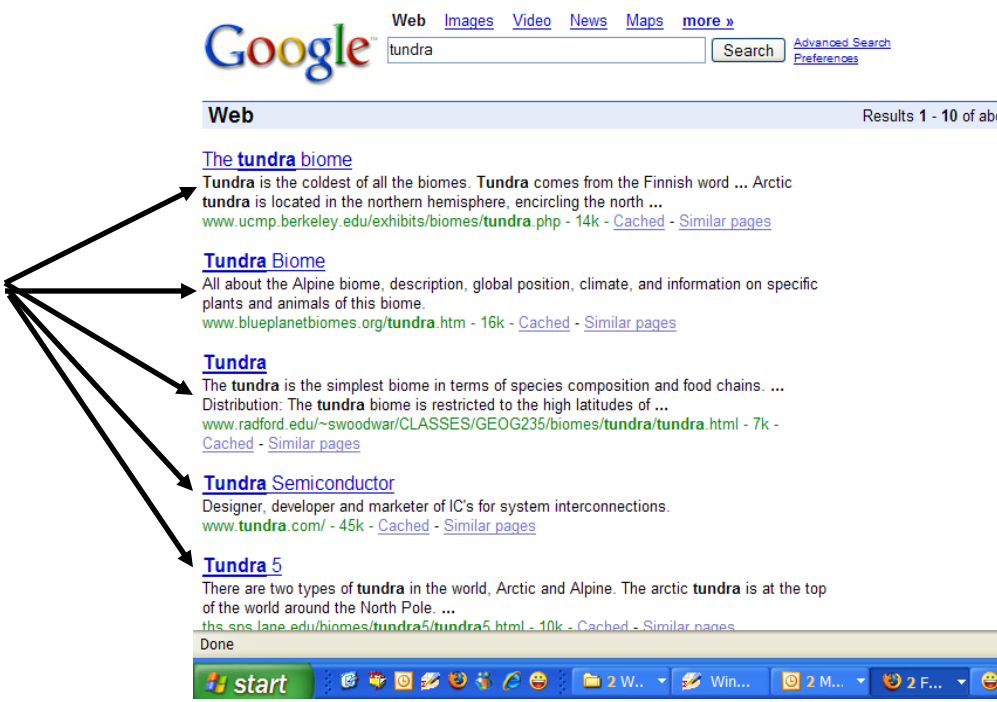
Determining Recall is Difficult

- Sometime it is not easy to estimate the total number of relevant items available.
  - Example: Santa Barbara
- A search engine needs to do well for queries that only have few relevant documents available.
  - Rare queries
  - Example: movie theater in camino real marketplace

Computing Recall/Precision Points for Ranked Results

- For a given query, produce the ranked list of retrievals.
- Mark each document in the ranked list that is relevant
- Compute a recall/precision pair for each position in the ranked list that contains a relevant document.





R- Precision (at Position R)

- **Precision at the R-th position in the ranking of results for a query that has R relevant documents.**

n	doc #	relevant
1	588	x
2	589	x
3	576	
4	590	x
5	986	
6	592	x
7	984	
8	988	
9	578	
10	985	
11	103	
12	591	
13	772	x
14	990	

R = # of relevant docs = 6

R-Precision = 4/6 = 0.67

Precision@6

Computing Recall/Precision Points: An Example

n	doc #	relevant
1	588	x
2	589	x
3	576	
4	590	x
5	986	
6	592	x
7	984	
8	988	
9	578	
10	985	
11	103	
12	591	
13	772	x
14	990	

Let total # of relevant docs = 6
Check each new recall point:

R=1/6=0.167; P=1/1=1

R=2/6=0.333; P=2/2=1

Recall@2
Precision@2

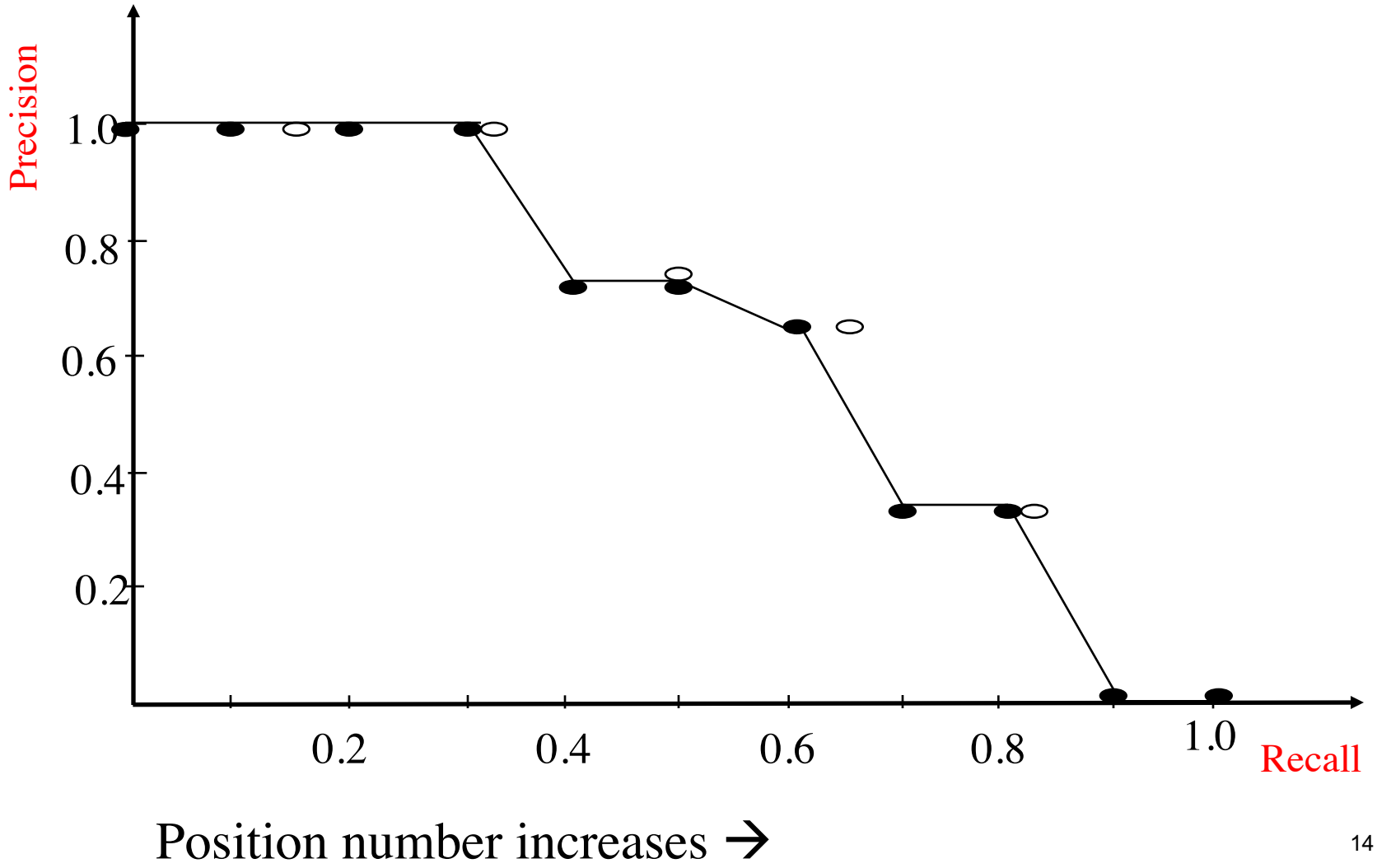
R=3/6=0.5; P=3/4=0.75

R=4/6=0.667; P=4/6=0.667

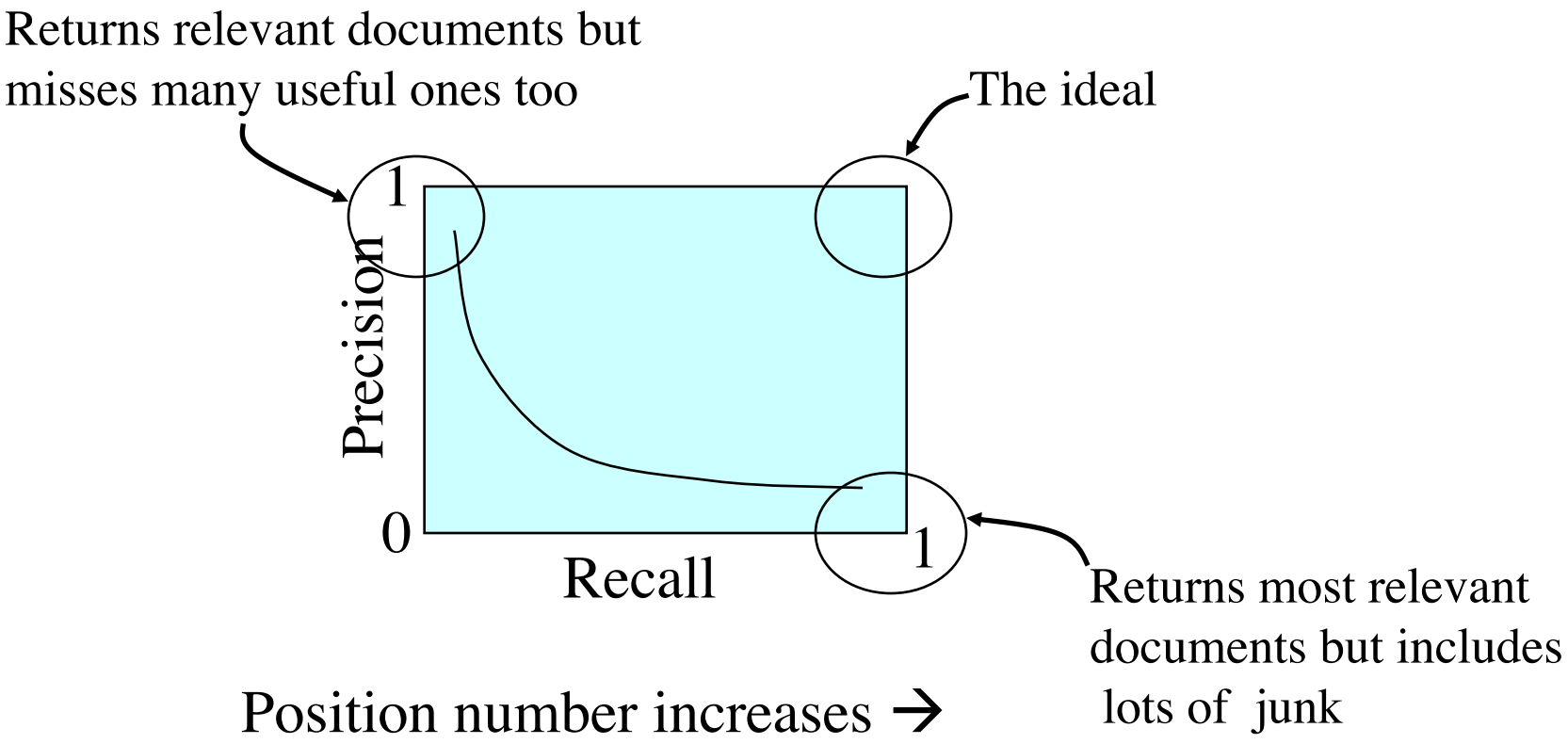
R=5/6=0.833; p=5/13=0.38

Missing one relevant document. Never reach 100% recall

Interpolating a Recall/Precision Curve: An Example



Trade-off between Recall and Precision



F-Measure

- One measure of performance that takes into account both recall and precision.
- Harmonic mean of recall and precision:

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

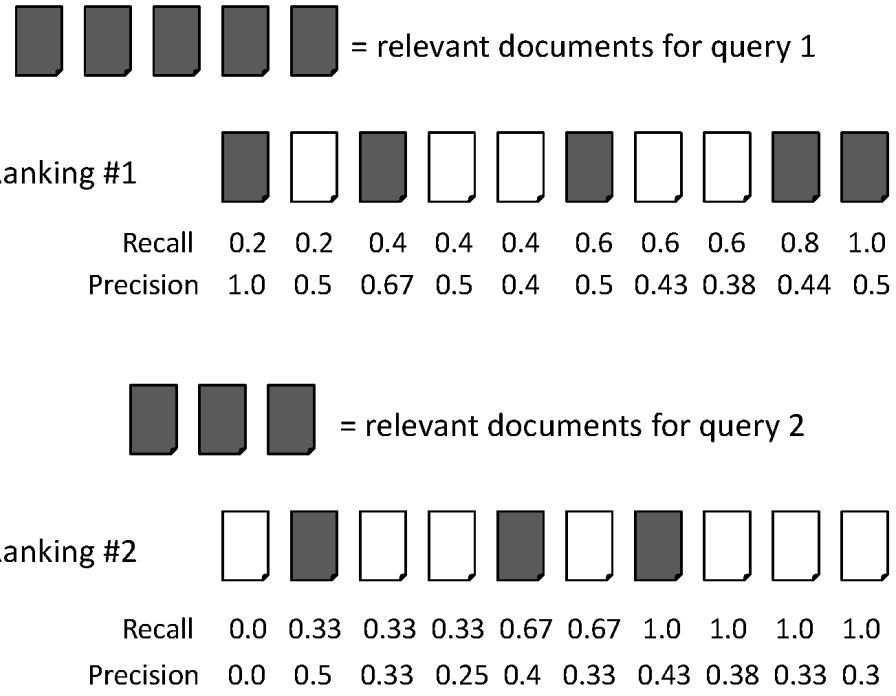
- A variant of F measure that allows weighting emphasis on precision over recall:

$$E = \frac{(1+\beta^2)PR}{\beta^2 P + R} = \frac{(1+\beta^2)}{\frac{\beta^2}{R}+\frac{1}{P}}$$

- Value of β controls trade-off:
  - β = 1: Equally weight precision and recall (E=F).
  - β > 1: Weight precision more.
  - β < 1: Weight recall more.

Averaging across Queries: MAP

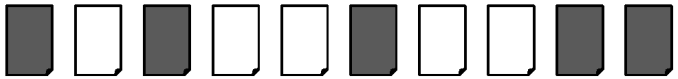
- ***How to evaluate when there are many queries***
- ***Mean Average Precision* (MAP)**
  - summarize rankings from multiple queries by averaging average precision
  - assumes user is interested in finding many relevant documents for each query



MAP Example:


= relevant documents for query 1

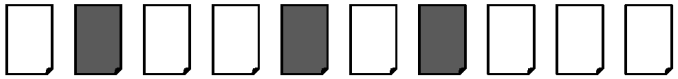
Ranking #1



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

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

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

Discounted Cumulative Gain

- **Popular measure for evaluating web search and related tasks**
- **Two assumptions:**
  - Highly relevant documents are more useful than marginally relevant document
    - Support relevancy judgment with multiple levels
  - 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
- **Gain is *discounted*, at lower ranks, e.g. 1/*log (rank)***
  - With base 2, the discount at rank 4 is 1/2, and at rank 8 it is 1/3

Discounted Cumulative Gain

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

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

  - Used in the rest of slides and our exercises
  - Alternative formulation:

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

  - 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@1, @2, @3 etc:**

  3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61

  DCG@3= sum of DCG at top 3 = 3+2+1.89 = 6.89

Normalized DCG

- **DCG values are often *normalized* by comparing the DCG at each rank with the DCG value for the *perfect ranking***
  - *Example:*
    - *DCG@5 = 6.89*
    - *Ideal DCG@5=9.75*
    - *NDCG@5=6.89/9.75=0.71*

- **NDCG numbers are averaged across a set of queries at specific rank values**

NDCG Example with Normalization

- **Perfect ranking:**

  3, 3, 3, 2, 2, 2, 1, 0, 0, 0
- **Ideal DCG@1, @2, …:**

  3, 6, 7.89, 8.89, 9.75, 10.52, 10.88, 10.88, 10.88, 10
- **My ranking:**

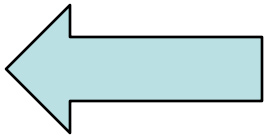
  3, 2, 3, 0, 0, 1, 2, 2, 3, 0
  - DCG@1, @2, etc:

    3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61
- **NDCG@1, @2, …**
  - normalized values (divide actual by ideal):

  1, 0.83, 0.87, 0.76, 0.71, 0.69, 0.73, 0.8, 0.88, 0.88
  - NDCG $\leq$ 1 at any rank position

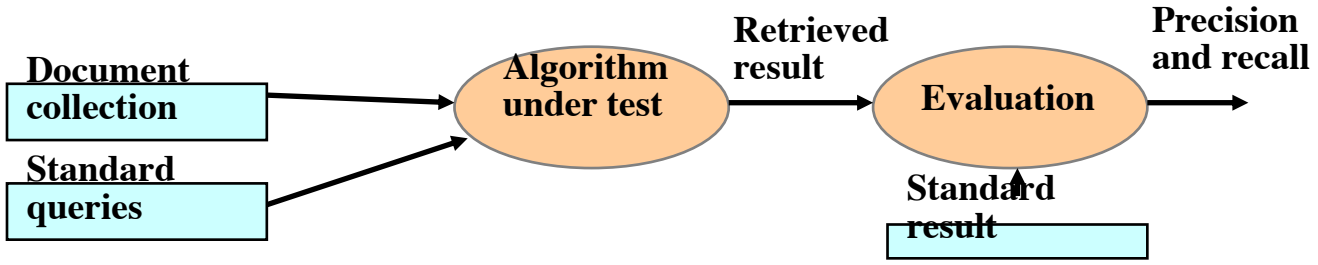
Table of Content

- **Search Engine Evaluation**
- **Metrics for relevancy**
  - Precision/recall
  - F-measure
  - MAP
  - NDCG
- **Creating Test Collections for IR Evaluation**
  - A/B testing



Relevance benchmarks

- **Relevant measurement requires 3 elements:**
  1. A benchmark document collection
  2. A benchmark suite of queries
  3. Editorial assessment of query-doc pairs
     - Relevant vs. non-relevant
     - Multi-level: Perfect, excellent, good, fair, poor, bad



- **Public benchmarks**
  - TREC: http://trec.nist.gov/
  - Microsoft/Yahoo published learning benchmarks

From document collections to test collections

- **Still need**
  - Test queries
  - Relevance assessments
- **Test queries**
  - Must be germane to docs available
  - Best designed by domain experts
  - Random query terms generally not a good idea
- **Relevance assessments**
  - Human judges, time-consuming
  - Are human panels perfect?

TREC

- **TREC Ad Hoc task from first 8 TRECs is standard IR task**
  - Human evaluation of pooled results returned
  - More recently other related things: Web track, HARD
- **A TREC query (TREC 5)**

<top>

<num> Number: 225

<desc> Description:

What is the main function of the Federal Emergency Management Agency (FEMA) and the funding level provided to meet emergencies? Also, what resources are available to FEMA such as people, equipment, facilities?

</top>

Standard relevance benchmarks: Others

- **GOV2**
  - Another TREC/NIST collection
  - 25 million web pages
  - Largest collection that is easily available
  - But still 3 orders of magnitude smaller than what Google/Yahoo/MSN index
- **ClueWeb**
  - Upto 1 billion web pages.
- **NTCIR**
  - East Asian language and cross-language information retrieval
- **Cross Language Evaluation Forum (CLEF)**
  - This evaluation series has concentrated on European languages and cross-language information retrieval.
- **Many others**

Kappa measure for inter-judge (dis)agreement

- **Kappa measure**
  - Agreement measure among judges
  - Designed for categorical judgments
  - Corrects for chance agreement
- **Kappa = [P(A) – P(E)] / [1 – P(E)]**
- **P(A) – proportion of time judges agree**
  - Relative observed agreement of judges
- **P(E) – what agreement would be by chance**
  - hypothetical probability of chance agreement
- **Kappa = 0 for chance agreement, 1 for total agreement.**

Kappa Measure: Example

P(A)? P(E)?

Number of docs	Judge 1	Judge 2
300	Relevant	Relevant
70	Nonrelevant	Nonrelevant
20	Relevant	Nonrelevant
10	Nonrelevant	Relevant

30

Kappa Example

#docs	Judge 1	Judge 2
300	Relevant	Relevant
70	Nonrel.	Nonrel.
20	Relevant	Nonrel.
10	Nonrelevant	Relevant

P(A) – proportion of time judges agree
P(E) –probability of chance agreement

- **P(A) = 370/400 = 0.925**
- **P(nonrelevant) = (10+20+70+70)/800 = 0.2125**
- **P(relevant) = (10+20+300+300)/800 = 0.7878**
- **P(E) = 0.2125^2 + 0.7878^2 = 0.665**
- **Kappa = (0.925 – 0.665)/(1-0.665) = 0.776**

- **Kappa > 0.8 = good agreement**
- **0.67 < Kappa < 0.8 -> “tentative conclusions” (Carletta ’96)**
- **Depends on purpose of study**
- **For >2 judges: average pairwise kappas**

Can we avoid human judgment?

- **No**
  - But once we have test collections, we can reuse them (so long as we don't overtrain too badly)
  - Makes experimental work hard
    - Especially on a large scale
- **In some very specific settings, can use proxies**
  - E.g.: for approximate vector space retrieval, use cosine distance closeness
- **Search engines also use non-relevance-based measures.**
  - Clickthrough on first result
    - Not very reliable if you look at a single clickthrough … but pretty reliable in the aggregate.
  - Studies of user behavior in the lab
  - A/B testing

A/B testing

- Purpose: Test a single innovation (variation)
- Prerequisite: Website with large traffic
- Have most users use old system
  - Divert a small proportion of traffic (e.g., 1%) to the new system that includes the innovation
- Evaluate with an “automatic” measure
  - Clickthrough.
  - Now we can directly see if the innovation (variation) does improve user happiness.