Evaluation of IR Models

Reference: Introduction to Information Retrieval
by C. Manning, P. Raghavan, H. Schutze

Evaluating an IR system

- Note: the **information need** is translated into a **query**
- Relevance is assessed relative to the **information need** *not* the **query**
- E.g., <u>Information need</u>: *I'm looking for information on whether drinking red wine is more effective at reducing your risk of heart attacks than white wine.*
- <u>Query</u>: ***wine red white heart attack effective***
- You evaluate whether the doc addresses the information need, not whether it has these words

Data Supporting Evaluation

- Relevant measurement requires 3 elements:
  1. A benchmark document collection
  2. A benchmark suite of queries
  3. A usually binary assessment of either <u>Relevant</u> or <u>Nonrelevant</u> for each query and each document
     - Some work on more-than-binary

Standard relevance benchmarks

- TREC - National Institute of Standards and Technology (NIST) has run a large IR test bed for many years
- Some other benchmark doc collections have been used
- "Retrieval tasks" specified
  - sometimes as queries
- Human experts mark, for each query and for each doc, Relevant or Nonrelevant
  - or at least for subset of docs that some system returned for that query

Should we instead use the accuracy measure for evaluation?

- Given a query, an engine classifies each doc as "Relevant" or "Nonrelevant"
- Equivalently, it returns a set of "Relevant" doc as the output result.
- The **accuracy** of an engine: the fraction of these classifications that are correct
- **Accuracy** is a commonly used evaluation measure in machine learning classification work

- Why is this not a very useful evaluation measure in IR?

Unranked Retrieval Evaluation

	Relevant	Nonrelevant
Retrieved	tp	fp
Not Retrieved	fn	tn

- Accuracy = (tp + tn) / (tp +fp + tn + fn)

A Sample Scenario

- In an IR system that handles 1,000 documents in a document collection.
- Suppose that given a particular query, the number of true relevant documents is 10.
- Consider a poor retrieval method that only returns 1 document and this document is relevant.

	Relevant	Nonrelevant
Retrieved	1	0
Not Retrieved	9	990

- The accuracy is (1+990)/1,000 = 0.991
- It is quite easy for a poor retrieval system to get high accuracy if it just returns an extremely small number of documents.

Metric - Recall

- To address the above problem, we may define a metric known as recall defined as:

  recall = fraction of gold standard relevant docs that can be retrieved

An Extreme Example

- What is the recall score of the following retrieval system?

- The recall score is 1.
- Intuitively, such retrieval system is not desirable.

Precision and Recall

- **Precision**: fraction of retrieved docs that are relevant
- **Recall**: fraction of relevant docs that are retrieved

	Relevant	Nonrelevant
Retrieved	tp	fp
Not Retrieved	fn	tn

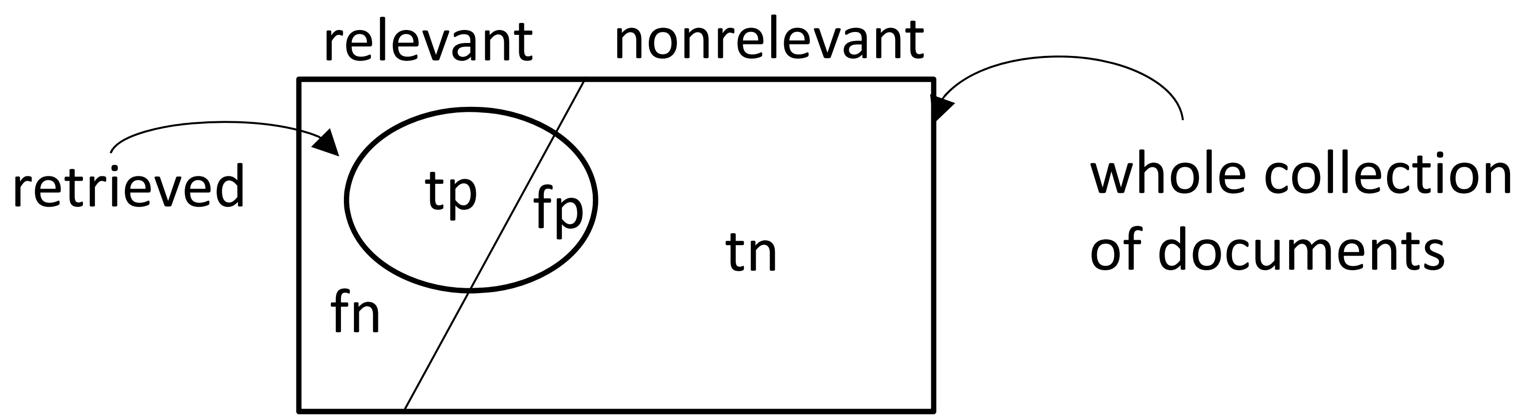
- Precision P = tp/(tp + fp)
- Recall R = tp/(tp + fn)

Precision and Recall

	Relevant	Nonrelevant
Retrieved	tp	fp
Not Retrieved	fn	tn

Precision P = tp/(tp + fp)

Recall   R = tp/(tp + fn)



Precision/Recall

- You can get high recall (but low precision) by retrieving all docs for all queries!
- Recall is a non-decreasing function of the number of docs retrieved
- In a good system, precision decreases as either the number of docs retrieved or recall increases
  – This is not a theorem, but it is just a general trend and it has been observed with strong empirical confirmation

A combined measure: *F*

- Combined measure that assesses precision/recall tradeoff is **F measure** (weighted harmonic mean):

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

- People usually use balanced F_1 measure
  - i.e., with $\beta = 1$ or $\alpha = ½$
- Harmonic mean is a conservative average
  - See CJ van Rijsbergen, *Information Retrieval*

Rank-Based Measures

Evaluating ranked results

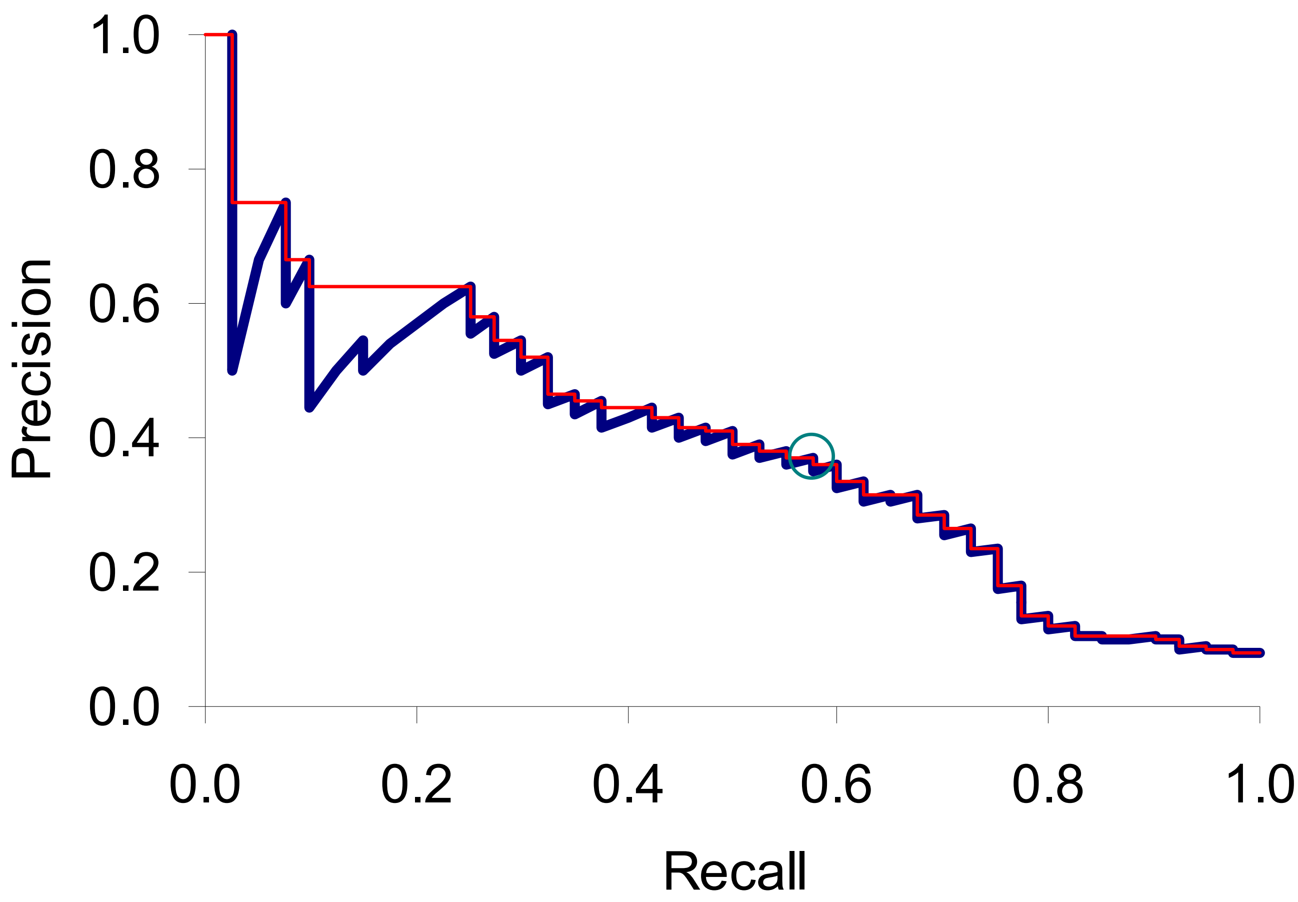
- Suppose that all the results are ranked:
  - The system can return any number of results
  - By taking various numbers of the top returned documents (levels of recall), the evaluator can produce a *precision-recall curve*

An example

Suppose that there are 20 relevant documents in the collection

ranked results		**precision**	**recall**
d14	Relevant	1.0	0.05
d3	Relevant	1.0	0.1
d26	Nonrelevant	0.67	0.1
d2	Relevant	0.75	0.15
d12	Nonrelevant	0.6	0.15
:	:	:	:

A precision-recall curve

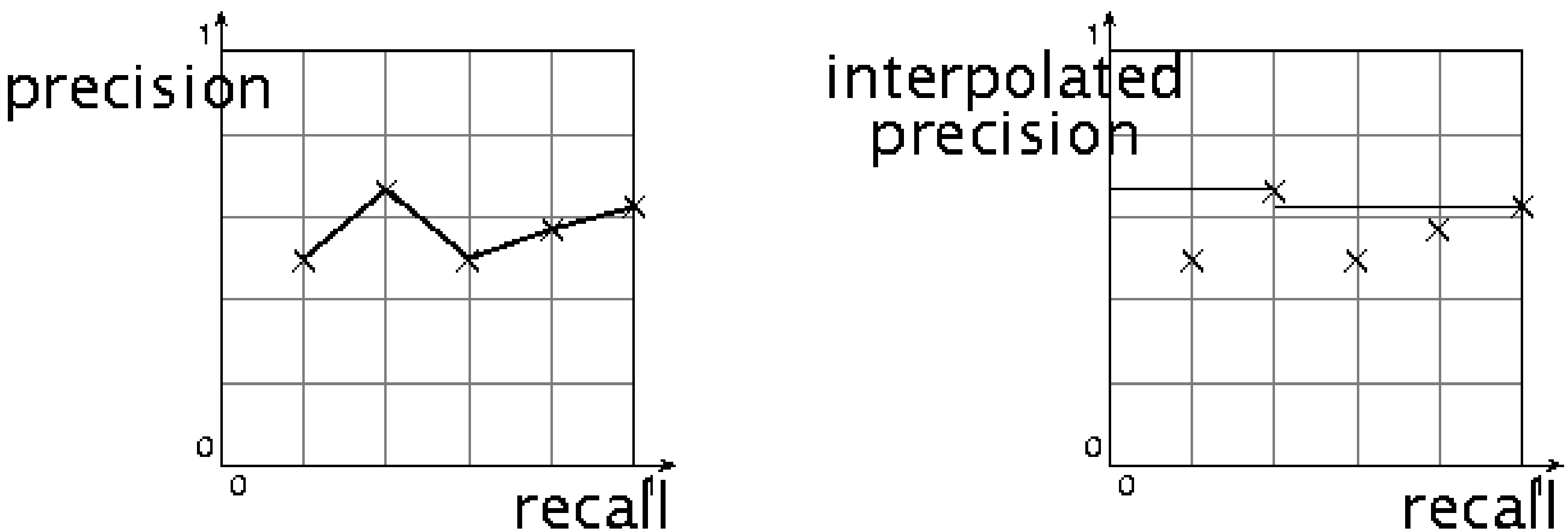


Averaging over queries

- A precision-recall graph for one query isn't a very sensible thing to look at
- You need to average performance over a whole bunch of queries.
- But there's a technical issue:
  - Precision-recall calculations place some points on the graph
  - How do you determine a value (interpolate) between the points?

Interpolated precision

- Idea: If locally precision increases with increasing recall, then you should get to count that…

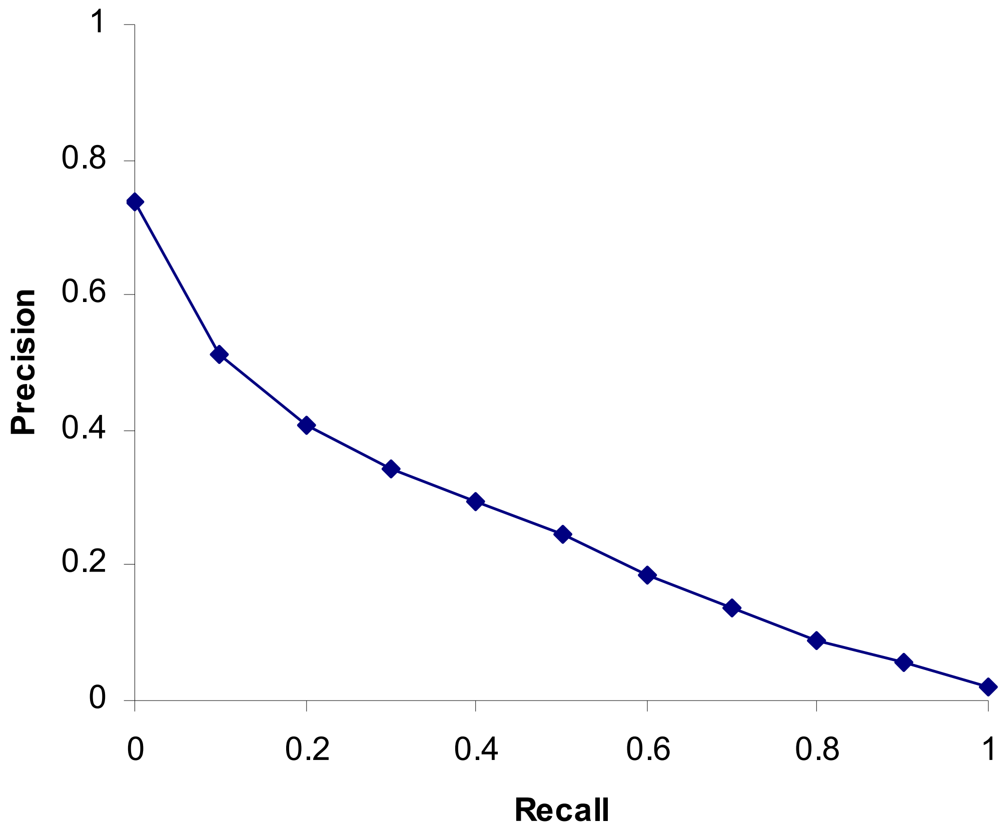


11-point Interpolated Average Precision

- Graphs are good, but people want summary measures!
- The standard measure in the early TREC competitions: you take the precision at 11 levels of recall varying from 0 to 1 by tenths of the documents, using interpolation (the value for 0 is always interpolated!), and average them
- Evaluates performance at all recall levels

Typical (good) 11 point precisions

- An example



Variance

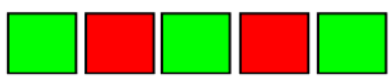
- For a test collection, it is usual that a system may perform poorly on some information needs (e.g., F = 0.1) and excellently on others (e.g., F = 0.7)
- Indeed, it is usually the case that the variance in performance of the same system across queries is much greater than the variance of different systems on the same query.
- That is, there are easy information needs and hard ones!

Other 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

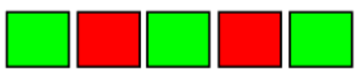
Mean Average Precision (MAP)

- Average of the precision value obtained for the top *k* documents, each time a relevant doc is retrieved
- Avoids interpolation, use of fixed recall levels
- MAP for query collection is arithmetic average.
  – Macro-averaging: each query counts equally

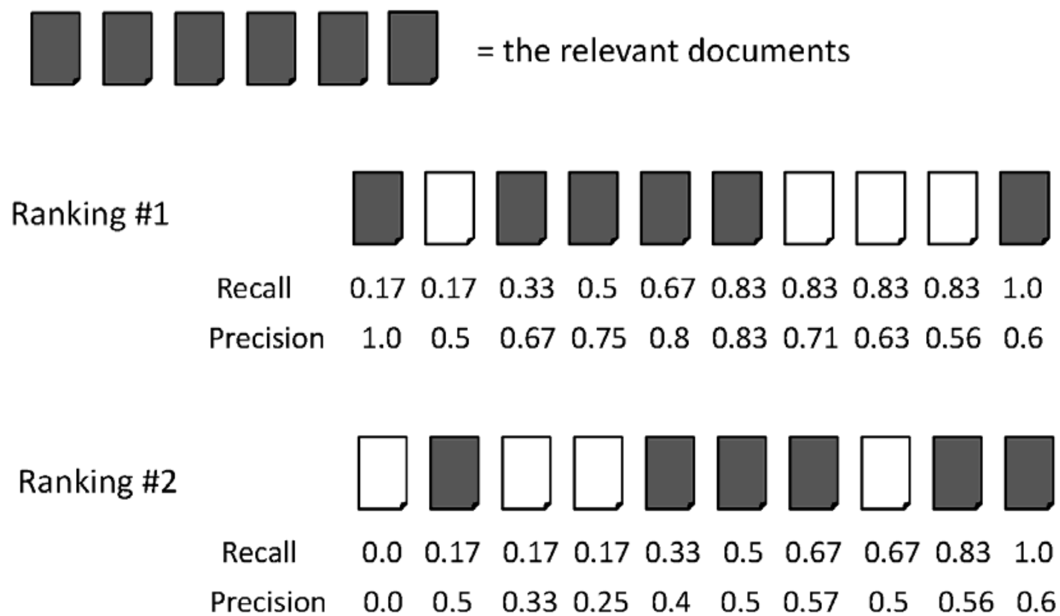
If the set of gold standard relevant documents for a query $q_j \in Q$ is $\left\{d_1, \cdots, d_{m_j}\right\}$ and R_{jk} is the set of ranked retrieval results from the top result until you get to document d_k, then

$$\mathrm{MAP}(Q) = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\left(\frac{1}{m_j}\sum_{k=1}^{m_j}\mathrm{Precision}\left(R_{jk}\right)\right)$$

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 $\frac{1}{3} \cdot \left(\frac{1}{1} + \frac{2}{3} + \frac{3}{5} \right) \approx 0.76$
- MAP is Average Precision across multiple queries/rankings

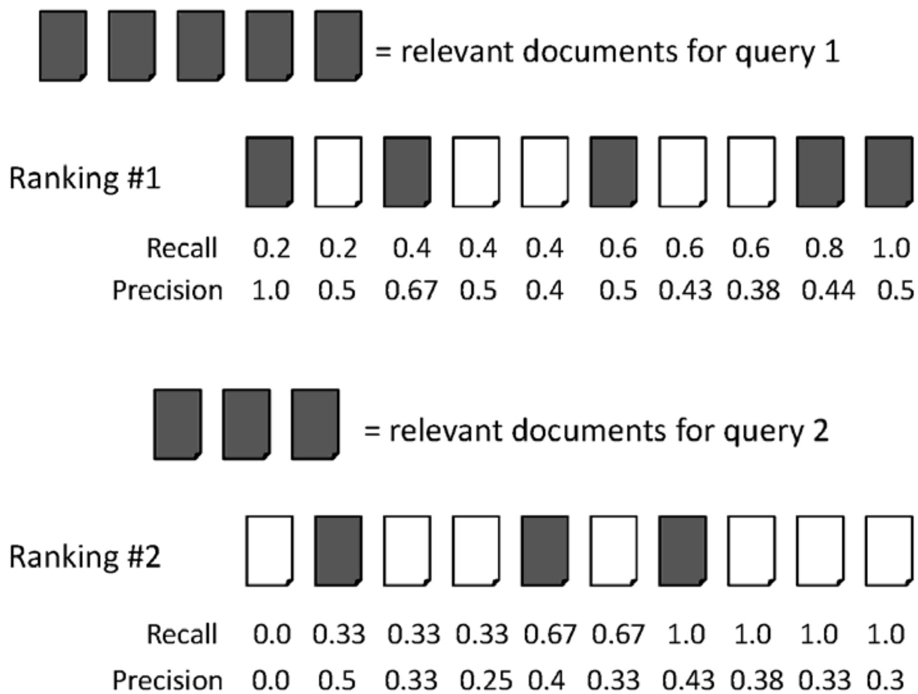
Average Precision



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

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

MAP



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

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

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
- 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

When There's only 1 Relevant Document

- Scenarios:
  - known-item search
  - navigational queries
  - looking for a fact
- Search Length = Rank of the answer
  - measures a user's effort

Mean Reciprocal Rank

- Consider rank position, K, of first relevant doc
- Reciprocal Rank score = $\dfrac{1}{K}$
- MRR is the mean RR across multiple queries

Discounted Cumulative Gain

- Popular measure for evaluating web search and related tasks

- Two assumptions:
  - Highly relevant documents are more useful than marginally relevant document
  - 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 $1/\log(\mathrm{rank})$
  - 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, …r_n (in ranked order)
  - CG = $r_1+r_2+\ldots r_n$
- Discounted Cumulative Gain (DCG) at rank n
  - DCG = $r_1 + r_2/\log_2 2 + r_3/\log_2 3 + \ldots\ r_n/\log_2 n$
    - We may use any base for the logarithm, e.g., base=b

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

Summarize a Ranking: NDCG

- Normalized 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
  - Compute the precision (at rank) where each (new) relevant document is retrieved => p(1),…,p(k), if we have k rel. docs

- 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$_1$		Ranking Function$_2$	
	Document Order	r_i	Document Order	r_i	Document Order	r_i
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
	$NDCG_{GT}$=1.00		$NDCG_{RF1}$=1.00		$NDCG_{RF2}$=0.9203	

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

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

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

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

$$\frac{4.2619}{4.6309} = 0.9203$$

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
- 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

Evaluation at large search engines

- Search engines have test collections of queries and hand-ranked results
- Recall is difficult to measure on the web
- Search engines often use precision at top k, e.g., k = 10
- . . . or measures that reward you more for getting rank 1 right than for getting rank 10 right.
  - NDCG (Normalized Cumulative Discounted Gain)
- 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