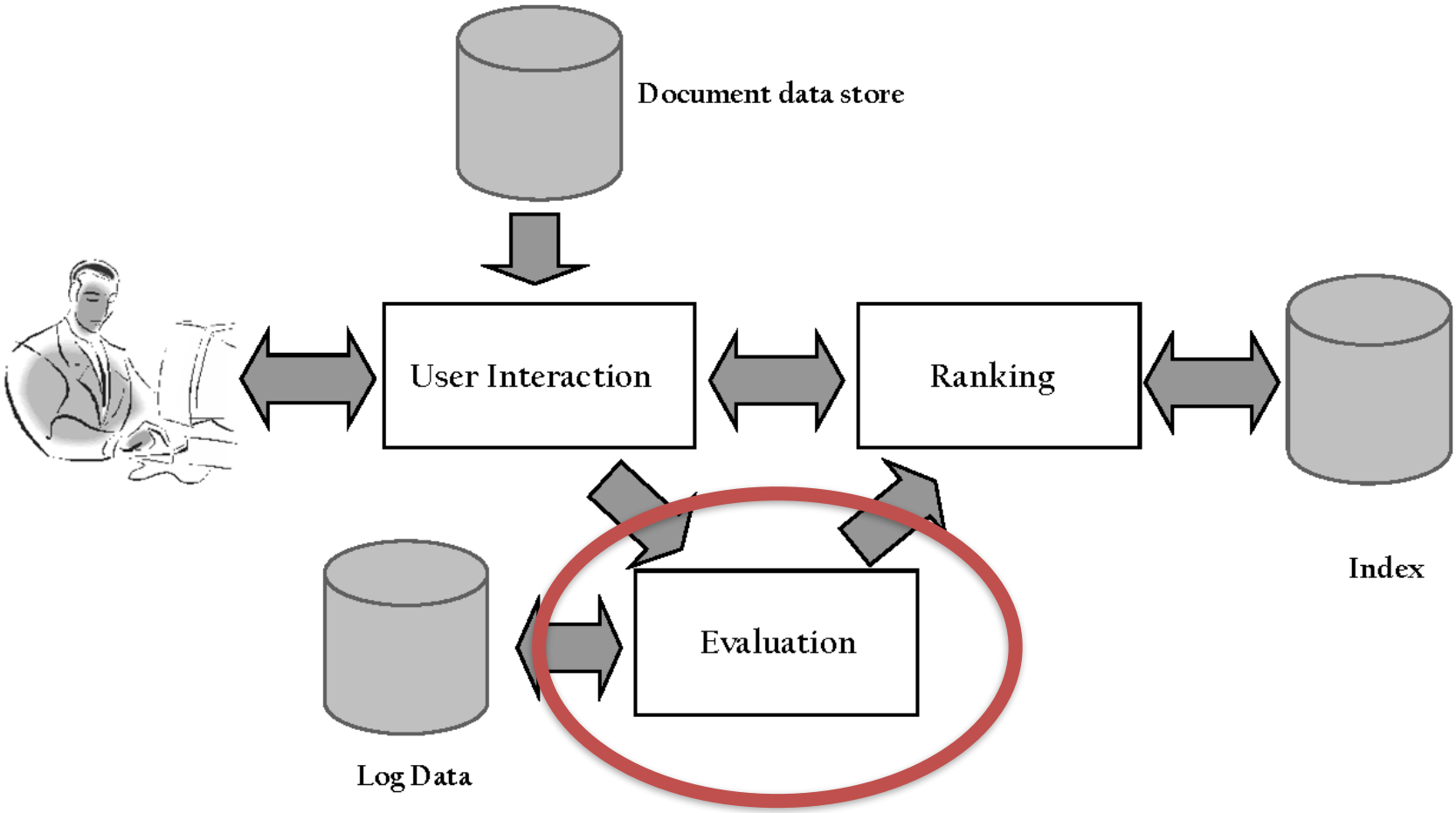
CS6200
Information Retrieval

Jesse Anderton
College of Computer and Information Science
Northeastern University

Query Process



IR Evaluation

- **Evaluation** is any process which produces a quantifiable measure of a system's performance.
- In IR, there are many things we might want to measure:
  - ➡ Are we presenting users with relevant documents?
  - ➡ How long does it take to show the result list?
  - ➡ Are our query suggestions useful?
  - ➡ Is our presentation useful?
  - ➡ Is our site appealing (from a marketing perspective)?

IR Evaluation

- The things we want to evaluate are often subjective, so it's frequently not possible to define a "correct answer."

- Most IR evaluation is comparative: "Is system A or system B better?"

  ➡ You can present system A to some users and system B to others and see which users are more satisfied ("A/B testing")

  ➡ You can randomly mix the results of A and B and see which system's results get more clicks

  ➡ You can treat the output from system A as "ground truth" and compare system B to it

Binary Relevance

Retrieval Effectiveness

- Retrieval effectiveness is the most common evaluation task in IR
- Given two ranked lists of documents, which is better?
  - ➡ A better list contains *more relevant documents*
  - ➡ A better list has relevant documents *closer to the top*
- But what does “relevant” mean and how can we measure it?

Relevance

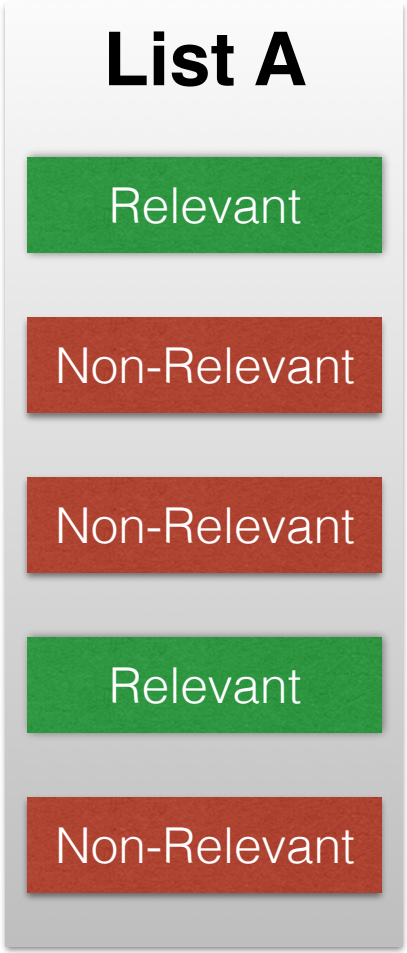
- The meaning of relevance is actively debated, and effects how we build rankers and choose evaluation metrics.
- In general, it means something like how “useful” a document is as a response to a particular query.
- In practice, we adopt a working definition in a given setting which approximates what we mean.
  - ➡ Page-finding queries: there is only one relevant document; the URL of the desired page.
  - ➡ Information gathering queries: a document is relevant if it contains any portion of the desired information.

Ambiguity of Relevance

- The ambiguity of relevance is closely tied to the ambiguity of a query's underlying information need
- Relevance is not independent of the user's language fluency, literacy level, etc.
- Document relevance may depend on more than just the document and the query. (Isn't true information more relevant than false information? But how can you tell the difference?)
- Relevance might not be independent of the ranking: if a user has already seen document A, can that change whether document B is relevant?

Binary Relevance

- For now, let's assume that a document is entirely relevant or entirely non-relevant to a query.
- This allows us to represent a ranking as a vector of bits representing the relevance of the document at each rank.
- Binary relevance metrics can be defined as functions of this vector.



$$\vec{r} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

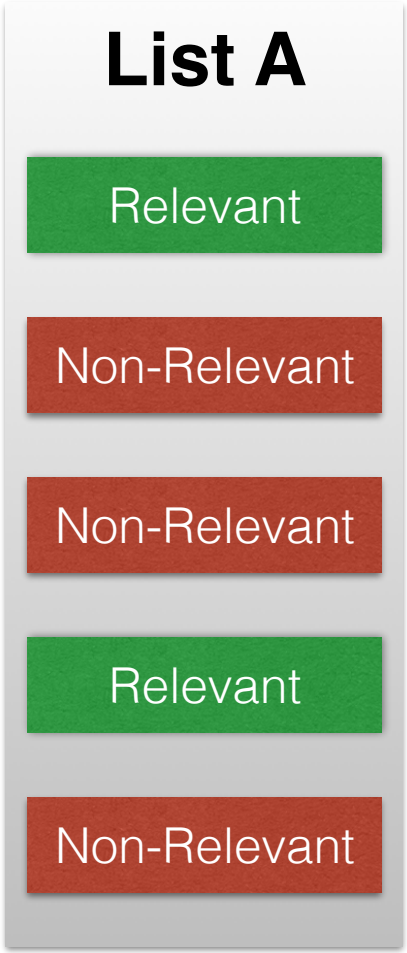
Recall

- **Recall** is the fraction of all possible relevant documents which your list contains.

$$
\begin{aligned}
recall(\vec{r}) &= \frac{1}{R}\sum_i r_i \\
&= \frac{rel(\vec{r})}{R} \\
&= Pr(retrieved|relevant)
\end{aligned}
$$

- **Recall@K** is almost identical, but truncates your list to the top K elements first.

$$recall@k(\vec{r}, k) = \frac{1}{R}\sum_i^k r_i$$



$$\vec{r} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$R = 10$$

$$recall(\vec{r}) = \frac{2}{10}$$

$$recall@k(\vec{r}, 3) = \frac{1}{10}$$

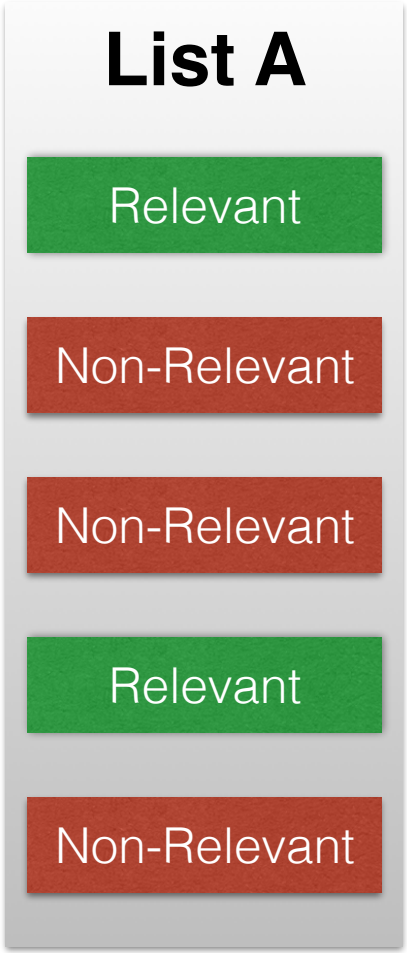
Precision

- **Precision** is the fraction of your list which is relevant.

$$prec(\vec{r}) = \frac{1}{|\vec{r}|}\sum_i r_i$$
$$= \frac{rel(\vec{r})}{|\vec{r}|}$$
$$= Pr(relevant|retrieved)$$

- **Precision@K** truncates your list to the top K elements.

$$prec@k(\vec{r}, k) = \frac{1}{k}\sum_i^k r_i$$



$$\vec{r} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$prec(\vec{r}) = \frac{2}{5}$$

$$prec@k(\vec{r}, 3) = \frac{1}{3}$$

Recall vs. Precision

- Neither recall nor precision is sufficient to describe a ranking's performance.
  - ➡ How to get perfect recall: retrieve all documents
  - ➡ How to get perfect precision: retrieve the one best document
- Most tasks find it relatively easy to get high recall or high precision, but doing well at both is harder.
- We want to evaluate a system by looking at how precision and recall are related.

F Measure

- The **F Measure** is one way to combine precision and recall into a single value.

$$F(\vec{r}, \beta) = \frac{(\beta^2 + 1) \cdot prec(\vec{r}) \cdot recall(\vec{r})}{\beta^2 \cdot prec(\vec{r}) + recall(\vec{r})}$$

- We commonly use the F1 Measure:

$$F1(\vec{r}) = F(\vec{r}, \beta = 1) = \frac{2 \cdot prec(\vec{r}) \cdot recall(\vec{r})}{prec(\vec{r}) + recall(\vec{r})}$$

- F1 is the *harmonic mean* of precision and recall.

- This heavily penalizes low precision and low recall. Its value is closer to whichever is smaller.

R-Precision

- Instead of using a cutoff based on the number of documents, use a cutoff for precision based on the recall score (or vice versa)

$$prec@r(\vec{s}, r) = prec@k(\vec{s}, k : recall@k(\vec{s}, k) = r)$$

$$recall@p(\vec{s}, p) = recall@k(\vec{s}, k : prec@k(\vec{s}, k) = p)$$

- As you move down the list:
  - ➡ recall increases monotonically
  - ➡ precision goes up and down, with an overall downward trend
- **R-Precision** is the precision at the point in the list where the two metrics cross.

$$rprec(\vec{s}) = prec@k(\vec{s}, k : recall@k(\vec{s}, k) = prec@k(\vec{s}, k))$$

Average Precision

- **Average Precision** is the mean of prec@k for every k which indicates a relevant document.

$$\Delta recall(\vec{s}, k) = recall@k(\vec{s}, k) - recall@k(\vec{s}, k-1)$$

$$ap(\vec{s}) = \sum_{k:rel(s_k)} prec@k(\vec{s}, k) \cdot \Delta recall(\vec{s}, k)$$

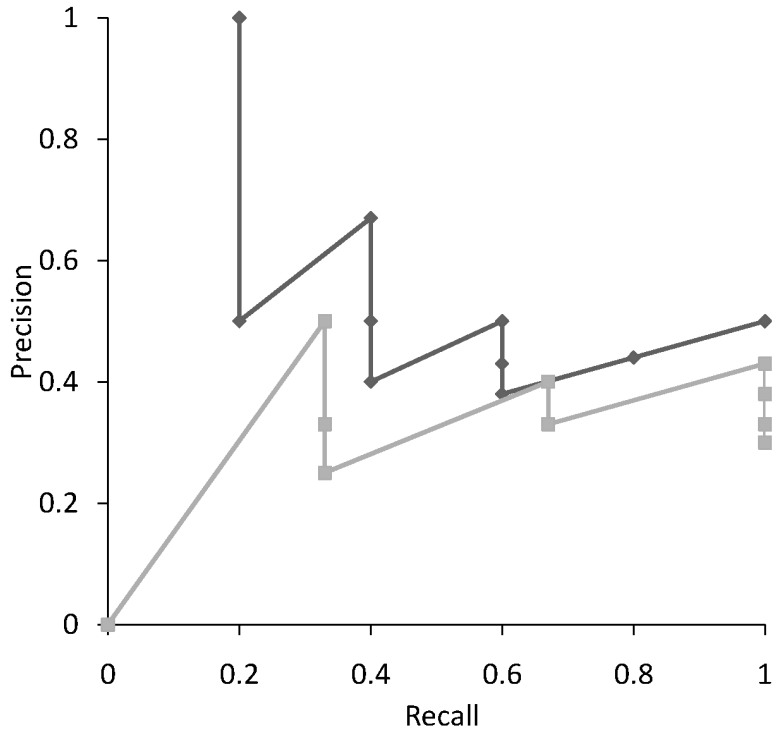
- Example:

$$\vec{r} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad prec@k = \begin{pmatrix} 1 \\ 1/2 \\ 1/3 \\ 1/2 \\ 2/5 \end{pmatrix} \quad \Delta recall = \begin{pmatrix} 0.5 \\ 0 \\ 0 \\ 0.5 \\ 0 \end{pmatrix}$$

$$\begin{aligned} ap &= (1 \cdot 0.5) + (1/2 \cdot 0.5) \\ &= 0.5 + 0.25 \\ &= 0.75 \end{aligned}$$

Precision-Recall Curves

- A **Precision-Recall Curve** is a plot of precision versus recall at the ranks of relevant documents.
- Average Precision is the area beneath the PR Curve.



Graded Relevance

Binary Relevance | **Graded Relevance** | Multiple Queries
Test Collections | Ranking for Web Search

Graded Relevance

- So far, we have dealt only with binary relevance
- It is sometimes useful to take a more nuanced view: two documents might both be relevant, but one might be better than the other.
- Instead of using relevance labels in {0,1}, we can use different values to indicate more relevant documents.
- We commonly use {0, 1, 2, 3, 4}

Ambiguity of Graded Relevance

- This adds its own ambiguity problems.
- It's hard enough to define "relevant vs. non-relevant," let alone "somewhat relevant" versus "relevant" versus "highly relevant."
- Expert human judges often disagree about the proper relevance grade for a document.
  - ➡ Some judges are stricter, and only assign high grades to the very best documents.
  - ➡ Some judges are more generous, and assign higher grades even to weaker documents.

A Graded Relevance Scale

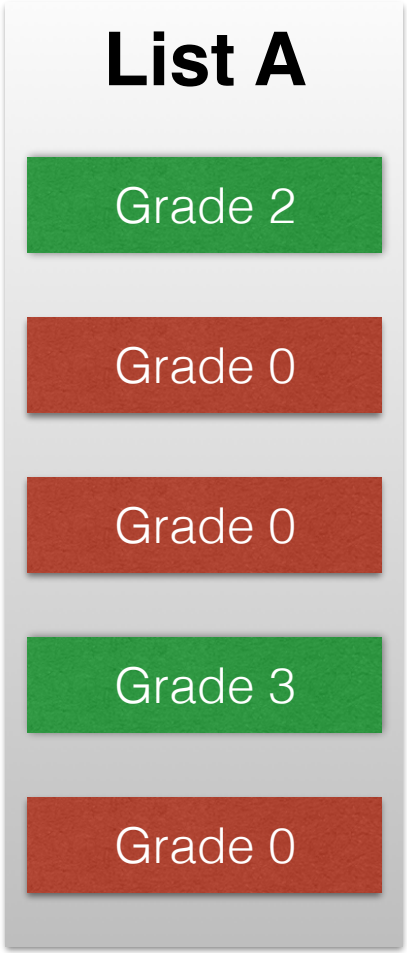
- Here is one possible scale to use.
  - ➡ Grade 0: Non-relevant documents. These documents do not answer the query at all (but might contain query terms!)
  - ➡ Grade 1: Somewhat relevant documents. These documents are on the right topic, but have incomplete information about the query.
  - ➡ Grade 2: Relevant documents. These documents do a reasonably good job of answering the query, but the information might be slightly incomplete or not well-presented.
  - ➡ Grade 3: Highly relevant documents. These documents are an excellent reference on the query and completely answer it.
  - ➡ Grade 4: Nav documents. These documents are the “single relevant document” for navigational queries.

Cumulative Gain

- **Cumulative Gain** is the total relevance score accumulated at a particular rank.

$$CG(\vec{r}, k) = \sum_{i=1}^{k} r_k$$

- This tries to measure the *gain* a user collects by reading the documents in the list.
- Problems: CG doesn't reflect the *order* of the documents, and treats a 4 at position 100 the same as a 4 at position 1.



$$\vec{r} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 3 \\ 0 \end{pmatrix}$$

$$CG(\vec{r}, 3) = 2$$

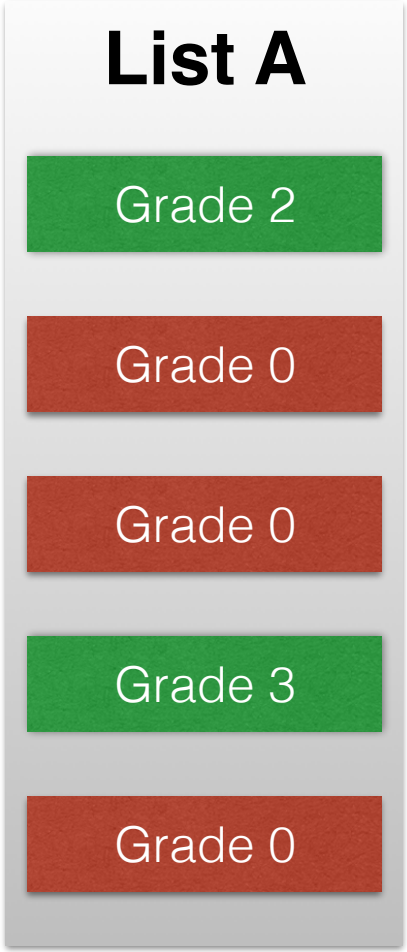
$$CG(\vec{r}, 5) = 5$$

Discounted Cumulative Gain

- **Discounted Cumulative Gain** applies some discount function to CG in order to punish rankings that put relevant documents lower in the list.

$$DCG(\vec{r}, k) = r_1 + \sum_{i=2}^{k} \frac{r_k}{\log_2 k}$$

- Various discount functions are used, but log() is fairly popular.
- A problem: the maximum value depends on the distribution of grades for this particular query, so comparing across queries is hard.



$$\vec{r} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 3 \\ 0 \end{pmatrix}$$

$$DCG(\vec{r}, 3) = 2$$

$$DCG(\vec{r}, 5) = 2 + \frac{3}{2}$$

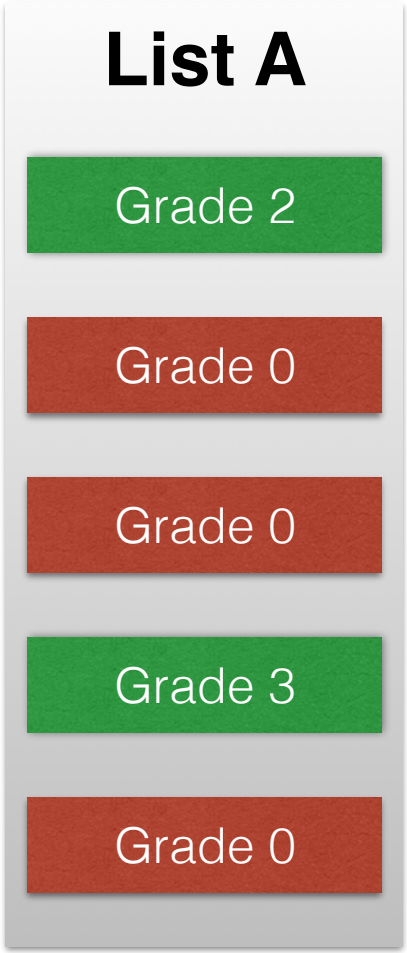
$$= 3.5$$

Normalized Discounted Cumulative Gain

- **Normalized Discounted Cumulative Gain** divides DCG by the best possible value for that query, the Ideal DCG (IDCG).

$$nDCG(\vec{r}, k) = \frac{DCG(\vec{r}, k)}{IDCG(k)}$$

- IDCG(k) is calculated by sorting all the documents in the collection in order of decreasing relevance grade, and then calculating DCG at cutoff k.



$$\vec{r} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 3 \\ 0 \end{pmatrix} \qquad \vec{c} = \begin{pmatrix} 3 \\ 3 \\ 2 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{aligned} IDCG(3) &= DCG(\vec{c}, 3) \\ &= 3 + \frac{3}{\log_2 2} + \frac{2}{\log_2 3} \\ &\approx 7.26 \end{aligned}$$

$$\begin{aligned} nDCG(\vec{r}, 3) &= \frac{DCG(\vec{4}, 3)}{IDCG(3)} \\ &\approx 2/7.26 \\ &\approx 0.275 \end{aligned}$$

Multiple Queries

Binary Relevance | Graded Relevance | **Multiple Queries**
Test Collections | Ranking for Web Search

Using Multiple Queries

- It isn't usually fair to compare system performance on a single query. What if the better system just got lucky?
- Instead, we commonly run both systems on a collection of different queries and compare metric values across all queries.
  - ➡ Individual queries can still be useful. Look for distinctive queries: a system's best or worst query, the queries for which the overall worse system beats the overall better system, etc.

Mean Metric Values

- One common way to combine information across queries is simply to take the mean of the metric over the queries.

- **Mean Average Precision** (MAP) is the average AP value for a system across many queries.

  ➡ This is one of the most popular evaluation metrics when using binary relevance.

Significance Tests

- Suppose System A beats System B on just one query. Do we believe it's better?
- Maybe System B would beat System A on some other query.
- How many queries do we need to try before we can be confident of the result?
  - ➡ Empirical results show that 25 queries are often enough
  - ➡ TREC generally uses at least 50 queries
- What if the systems are identical for all but one query, for which A is better? A would have a higher average than B…
- What if A's average is just 0.0001% higher than B's average? Is it better?

Significance Tests

- Statistical significance tests help us determine whether the observed differences in two systems are likely to be due to chance (or “luck”).
- One-Sample Tests: “Is the system’s response time under one second?”
- Two-Sample Tests: “Does the system perform equally well on these two queries?”
- Paired-Sample Tests: “Is System A better than System B?”

Statistical Terminology

- **Populations** are sets of objects of interest
  - ➡ e.g. all possible queries
- **Samples** are objects drawn from the population
  - ➡ e.g. the particular queries you're testing with
- **Statistics** are functions of data
  - ➡ e.g. A system's AP on a particular query
- We calculate our statistics on a sample of the population to test a hypothesis (e.g. "System A is better than System B") for the entire population.

Hypothesis Testing

- A significance test allows us to measure the probability that a result we observe happened by chance.
- We compare the probability of two possible hypotheses:
  - ➡ The **null hypothesis**: “Systems A and B are not different”
  - ➡ The **alternative hypothesis**: “System A is better than System B”
- The **power** of a hypothesis test is the probability that it will correctly reject the null hypothesis.
- A test’s power can be increased by increasing the number of queries in the experiment.

Hypothesis Testing

1. Compute the effectiveness measure for every query for both systems.

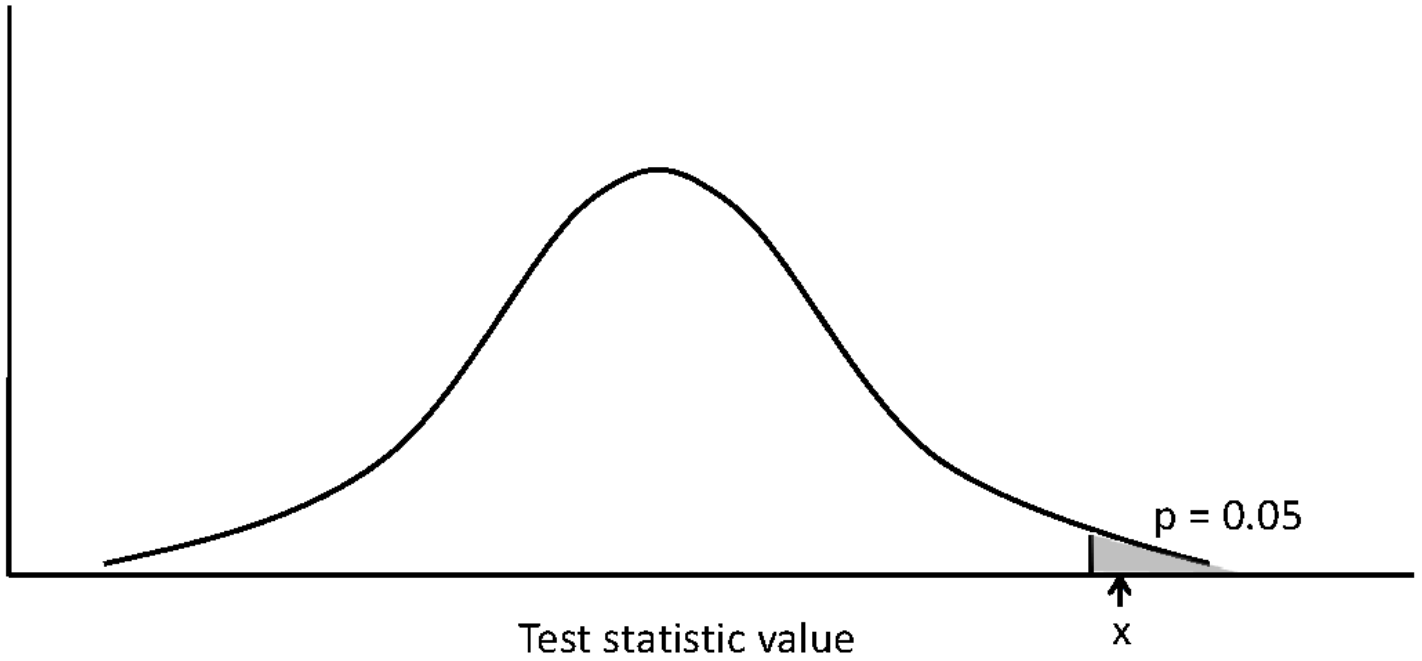
2. Compute a **test statistic** based on comparing the two systems' measures for each query. The details of this step depend on the particular test you're using.

3. The test statistic is used to compute a **P-value**: the probability that a test statistic value at least that extreme could be observed if the null hypothesis were true. The smaller the p-value is, the more confidently we can reject the null hypothesis.

4. We reject the null hypothesis if the p-value is smaller than some predetermined value, the **significance level**. The significance level is small: the smaller, the better. It should be at most 0.05.

One-Sided Test

- The distribution of possible test statistic values, assuming that the null hypothesis is true:



- The shaded area is the **region of rejection**

Example Experimental Results

Query	A	B	B-A
1	25	35	10
2	43	84	41
3	39	15	-24
4	75	75	0
5	43	68	25
6	15	85	70
7	20	80	60
8	52	50	-2
9	49	58	9
10	50	75	25

t-Test

- Assumes that the difference between the effectiveness values is a sample from a normal distribution
- The null hypothesis is that the mean of the distribution of differences is zero
- The test statistic is: $t = \dfrac{\overline{B-A}}{\sigma_{B-A}} \cdot \sqrt{N}$
- Example:

$\overline{B-A} = 21.4, \sigma_{B-A} = 29.1; t = 2.33, p\text{-}value = 0.02$

Wilcoxon Signed-Ranks Test

- A nonparametric test based on the differences between effectiveness scores
- Test statistic is: $w = \sum_{i=1}^{N} R_i$
- N is the number of differences. R_i is a *signed-rank*.
- To compute the signed-ranks, the differences are ordered by their absolute values (increasing) and then assigned rank values.
- Rank values are then given the sign of the original difference.

Wilcoxon Example

- 9 non-zero differences are (in rank order of absolute value):

  2, 9, 10, 24, 25, 25, 41, 60, 70

- Signed-ranks:

  -1, +2, +3, -4, +5.5, +5.5, +7, +8, +9

- Test statistic:

  w = 35, p-value = 0.025

Test Collections

Test Collections

- Several organizations have built standard collections of documents, queries, and relevance judgements for use in IR evaluation.
- These test collections allow the comparison of systems across many teams and publications by providing a standard measure of performance.
- These collections are used more in research than industry, as we'll see later.

TREC

- The Text Retrieval Conference was established in 1992 to construct large-scale IR test collections
  - ➡ Run by NIST's Information Access Division
  - ➡ Initially sponsored by DARPA as part of Tipster program
- Probably the best-known IR evaluation setting, with participants from dozens of countries
- Proceedings are available from http://trec.nist.gov

TREC Tracks

- TREC is organized into roughly a dozen independent research tracks each year, often run by volunteers outside of NIST.
  - ➡ November: tracks approved by TREC community
  - ➡ Winter: track members finalize format for track
  - ➡ Spring: researchers train systems based on track specification
  - ➡ Summer: researchers carry out formal evaluation (usually “blind” – the researchers do not know the answer)
  - ➡ Fall: NIST carries out evaluation
  - ➡ November: Group meeting (at NIST) to find out how well your submission did, and what other track members tried

TREC Tracks

- Examples of TREC tracks:
  - ➡ Ad-hoc retrieval: classic keyword document search.
  - ➡ Question answering: responding to questions with factoids instead of with documents.
  - ➡ Crowdsourcing test collections: can we collect accurate relevance grades from anonymous crowd workers?
  - ➡ Temporal summarization: How much was known about event *e* at time *t*?

TREC Topic Example

<top>
<num> Number: 794

<title> pet therapy

<desc> Description:
How are pets or animals used in therapy for humans and what are the benefits?

<narr> Narrative:
Relevant documents must include details of how pet- or animal-assisted therapy is or has been used. Relevant details include information about pet therapy programs, descriptions of the circumstances in which pet therapy is used, the benefits of this type of therapy, the degree of success of this therapy, and any laws or regulations governing it.

</top>

Historically Important Collections

- CACM: titles and abstracts from the Communications of the ACM from 1958-1979. Queries and relevance judgements generated by computer scientists.

- AP: Associated Press newswire documents from 1988-1990 (from TREC disks 1-3). Queries are the title fields from TREC topics 51-150. Topics and relevance judgements generated by government information analysts.

- GOV2: Web pages crawled from websites in the .gov domain during early 2004. Queries are the title fields from TREC topics 701-850. Topics and relevance judgements generated by government analysts.

Historically Important Collections

Collection	Number of documents	Size	Average number of words/doc.
CACM	3,204	2.2 Mb	64
AP	242,918	0.7 Gb	474
GOV2	25,205,179	426 Gb	1073

Collection	Number of queries	Average number of words/query	Average number of relevant docs/query
CACM	64	13.0	16
AP	100	4.3	220
GOV2	150	3.1	180

Recent Collections

- TREC8 (1999): A very thoroughly-evaluated collection of documents and queries. Considered to have very accurate relevance scores for the documents, but the documents and queries are not ideal for modern web search.

- CLUEWEB09 (2009): A 25TB crawl of the web containing 1,040,809,705 web pages in 10 languages. Fewer queries and relevance grades available (largely because of its scale).

- CLUEWEB12 (2012): A collection of 733,019,372 English web pages crawled in early 2012. Fewer queries and relevance grades available. Used by many current TREC tracks.

Pooling

- The large size of recent collections makes judging all documents for a query impractical.
- At TREC, a technique called **pooling** is used to compare the performance of several submitted runs.
  - ➡ Each team submits one or more rankings produced by their system(s).
  - ➡ The top k results from each ranking are merged into a pool.
  - ➡ Duplicates are removed.
  - ➡ The documents are presented to human judges in random order.
- This produces a large number of relevance judgements for each query, although still incomplete

Ranking for Web Search

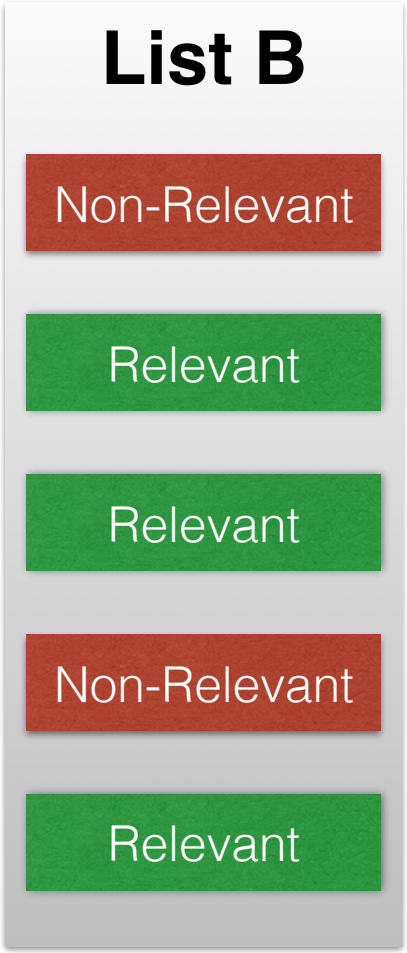
Binary Relevance | Graded Relevance | Multiple Queries
Test Collections | **Ranking for Web Search**

Search Engine Evaluation

- Consider the context of a web search engine.
  - ➡ Recall is not very important: there are usually far too many relevant documents for a user to see or process all of them.
  - ➡ In most cases, the user won't even see the rankings after the first page.
- Search engines are often interested in precision at the top few ranks: prec@10, or even prec@3.
- Search engines also have access to different kinds of data, which allows them to develop custom (proprietary, often secret) metrics.

Reciprocal Rank

- The Reciprocal Rank (RR) is the reciprocal of the rank of the first relevant document.
- The Mean Reciprocal Rank (MRR) is the RR averaged across many queries.
- This is very sensitive to rank position, and useful when the user will only see a few documents.



$$\vec{r} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

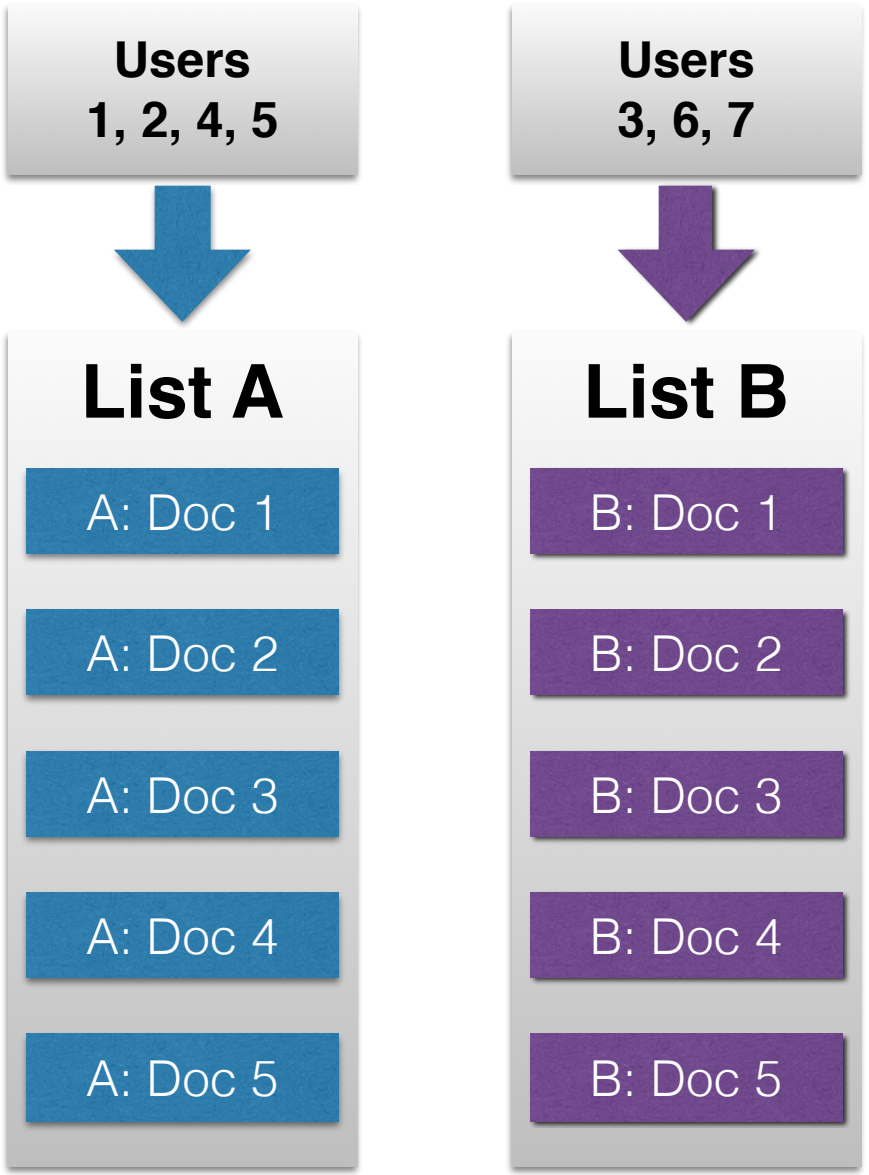
$$RR = \frac{1}{2}$$

Leveraging Users

- Search engines also have a resource most researchers don't: massive numbers of daily users
- This allows them to more directly compare user satisfaction of different systems:
  - ➡ Do users click on the top documents, or further down the list?
  - ➡ Do users come back to the results and click other documents?
  - ➡ How often do users reformulate their queries?
- These values can be averaged across many users and queries for each system to compare the systems.

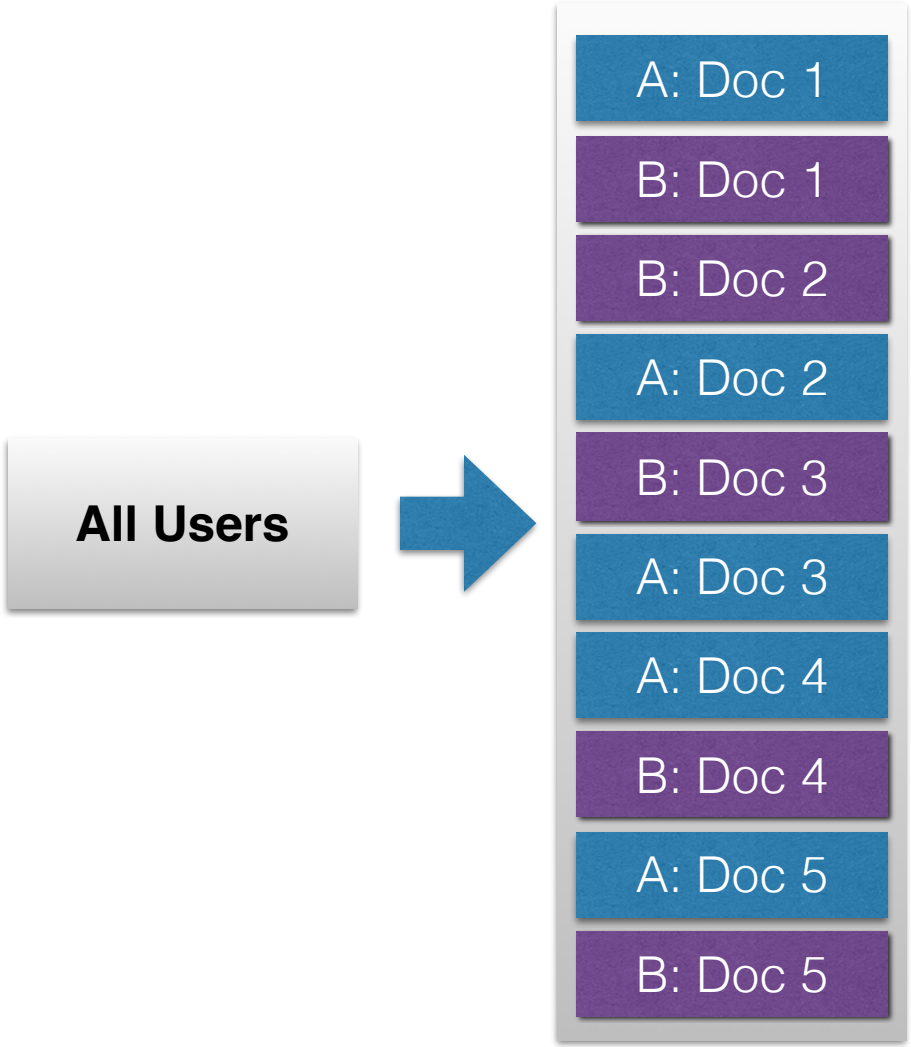
A/B Testing

- One way to compare two systems is to randomly assign users to one of the systems and compare user satisfaction between groups.
- This is known as A/B Testing, and can be used to compare whatever metrics you desire.



Interleaving Results

- Another way to compare to systems is to randomly interleave their results, and measure which system's results get clicked more often.
- A new random interleaving is chosen for each user, so we can average out the benefits a system may gain from one particular ordering.



Search Engine Performance

- Many other metrics are of interest to search engines:
  - ➡ Elapsed indexing time: How long does it take to index a document?
  - ➡ Indexing processor time: How much CPU time does the indexing process take? (Ignores time spent waiting for I/O.)
  - ➡ Indexing temporary space: The amount of transient disk space used when creating an index.
  - ➡ Index size: The amount of disk space used for the index overall.
  - ➡ Query throughput: number of queries processed per second.
  - ➡ Query latency: The amount of time a user must wait before receiving a response to a query.

Summary

- No single metric is ideal for every situation.
- You usually want to look at a combination of metrics to examine different aspects of your system.
- It's important to use aggregated metrics across many queries and use statistical significance tests.
- It's also important to analyze performance on individual queries to understand where your system has the most trouble