# Evaluation

Module Introduction

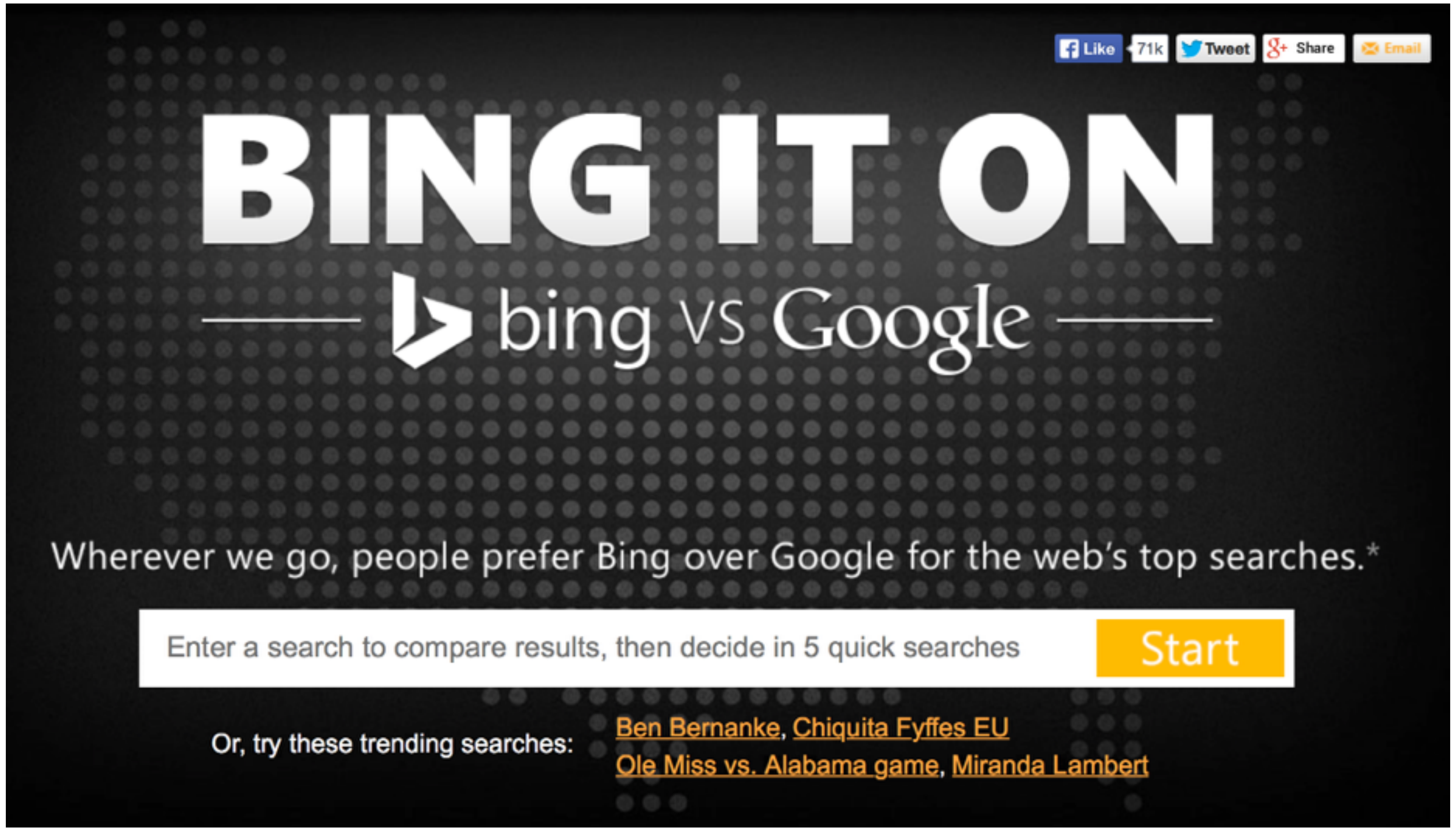
# Comparing Search Engines

How should we evaluate IR systems?

A user-driven side by side comparison?

By calculating automatic scores on a query-by-query basis?

Which documents and queries should we use, and how do we decide which system is better?



Wherever we go, people prefer Bing over Google for the web's top searches.*

Enter a search to compare results, then decide in 5 quick searches

Or, try these trending searches: Ben Bernanke, Chiquita Fyffes EU, Ole Miss vs. Alabama game, Miranda Lambert

Screenshot: http://www.bingiton.com

# User-Driven Evaluation

IR systems are designed to help people find information, so we should ideally measure their effectiveness with actual users.

Challenges include selecting appropriate test users, choosing natural search topics for those users, and removing potential sources of bias from the search interface.

It’s also important to get IRB approval.

# Automatic Evaluation

IR research has usually favored automatic evaluation on standard collections of documents and queries. It’s faster, cheaper, and easier to replicate.

Challenges include building or choosing an appropriate collection, collecting relevance judgements for your collection, and choosing the right effectiveness measures based on the user task you’re evaluating.

Let’s get started!

# Relevance, Precision, and Recall

Evaluation, session 2

# 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.

Here, we focus mostly on *retrieval effectiveness*.

**IR Evaluation Questions**

- 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)?

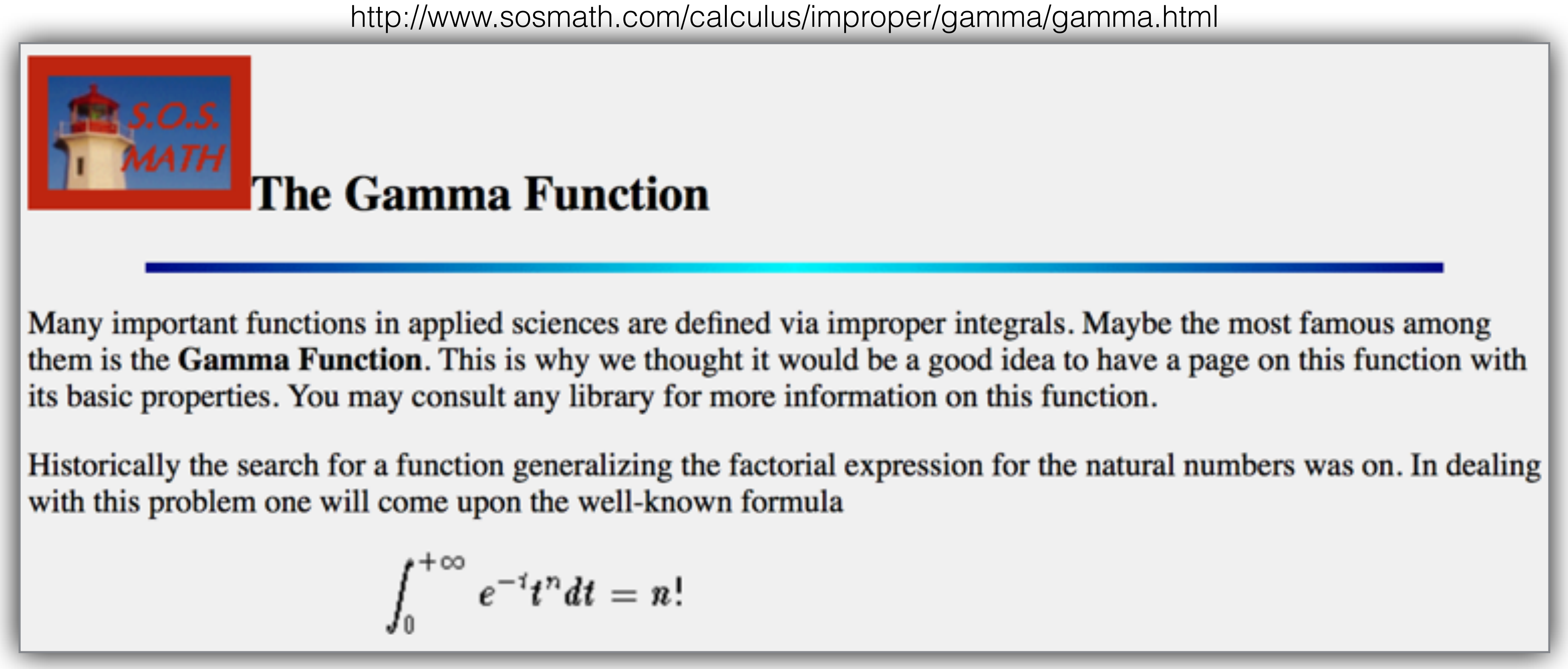
# Retrieval Effectiveness

Retrieval effectiveness is inherently subjective, because the relevance of a document to a query is subjective.

*Relevance* roughly means “satisfying the information need,” but for a precise evaluation we need a precise definition.

The appropriate definition depends on the task you are evaluating.

http://www.sosmath.com/calculus/improper/gamma/gamma.html



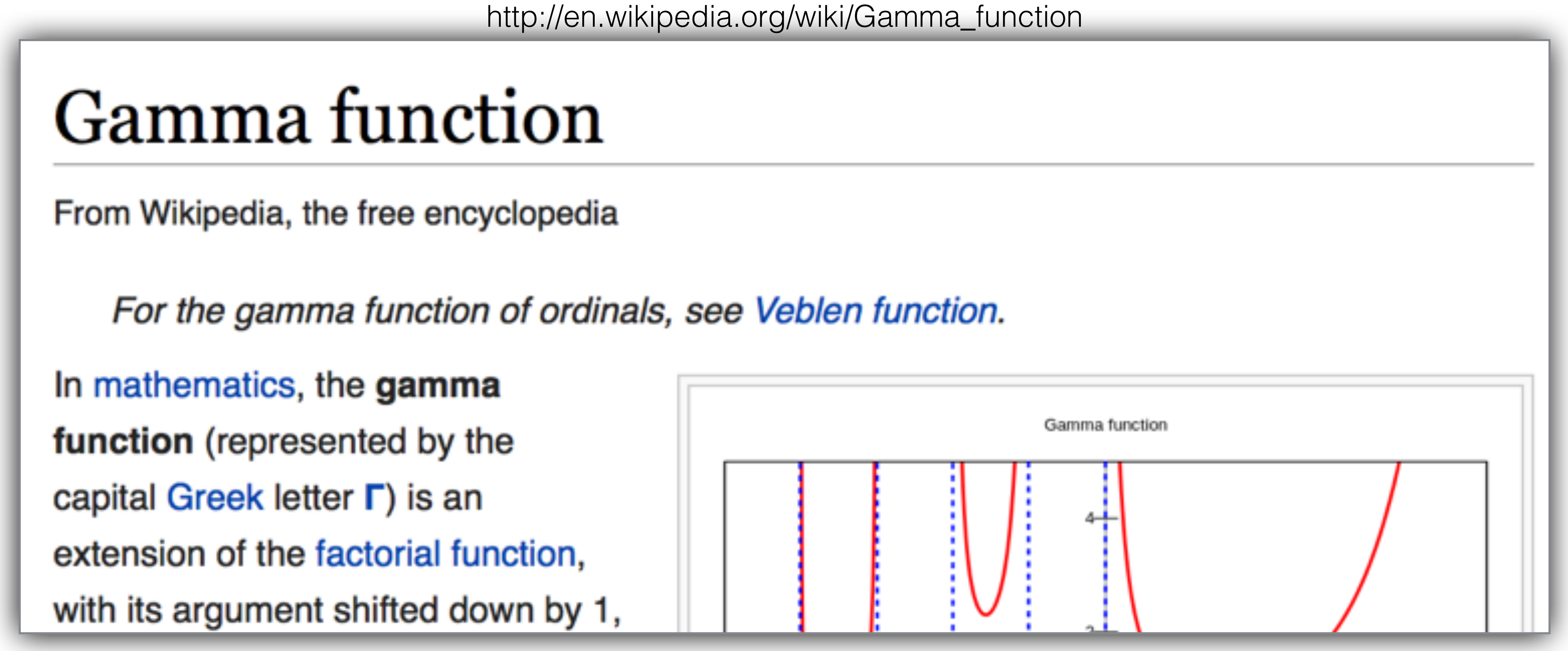
**The Gamma Function**

Many important functions in applied sciences are defined via improper integrals. Maybe the most famous among them is the **Gamma Function**. This is why we thought it would be a good idea to have a page on this function with its basic properties. You may consult any library for more information on this function.

Historically the search for a function generalizing the factorial expression for the natural numbers was on. In dealing with this problem one will come upon the well-known formula

$$\int_0^{+\infty} e^{-t} t^n dt = n!$$

http://en.wikipedia.org/wiki/Gamma_function



Gamma function

From Wikipedia, the free encyclopedia

*For the gamma function of ordinals, see Veblen function.*

In mathematics, the **gamma function** (represented by the capital Greek letter **Γ**) is an extension of the factorial function, with its argument shifted down by 1,

**Which is better? It depends.**

# Evaluating a Ranking

Given a ranking of documents, we can create a *confusion matrix* that counts the correct and incorrect answers of each type.

- *True Positives* are relevant documents in the ranking
- *False Positives* are non-relevant documents in the ranking
- *True Negatives* are non-relevant documents missing from the ranking
- *False Negatives* are relevant documents missing from the ranking

| | **Relevant** | **Non-Relevant** |
|---|---|---|
| **Retrieved** | TP | FP |
| **Not Retrieved** | FN | TN |

**Confusion Matrix**

# Recall

*Recall* is the fraction of relevant documents retrieved by the system.

*Recall@k* is the fraction of relevant documents in the top $k$ results.

A task is said to be *recall-oriented* when the user wants to make sure they have not missed any relevant detail (e.g. legal discovery).

| | Relevant | Non-Relevant |
|---|---|---|
| **Retrieved** | TP | FP |
| **Not Retrieved** | FN | TN |

**Confusion Matrix**

$$\begin{aligned} recall &:= \frac{num(\textbf{retrieved relevant})}{num(\textbf{relevant})} \\ &= \frac{TP}{TP + FN} \end{aligned}$$

# Precision

*Precision* is the fraction of retrieved documents that were relevant.

*Precision@k* is the fraction of the top $k$ results that were relevant.

A task is said to be *precision-oriented* when the user wants just a few high-quality documents (e.g. most web search).

| | Relevant | Non-Relevant |
|---|---|---|
| **Retrieved** | TP | FP |
| **Not Retrieved** | FN | TN |

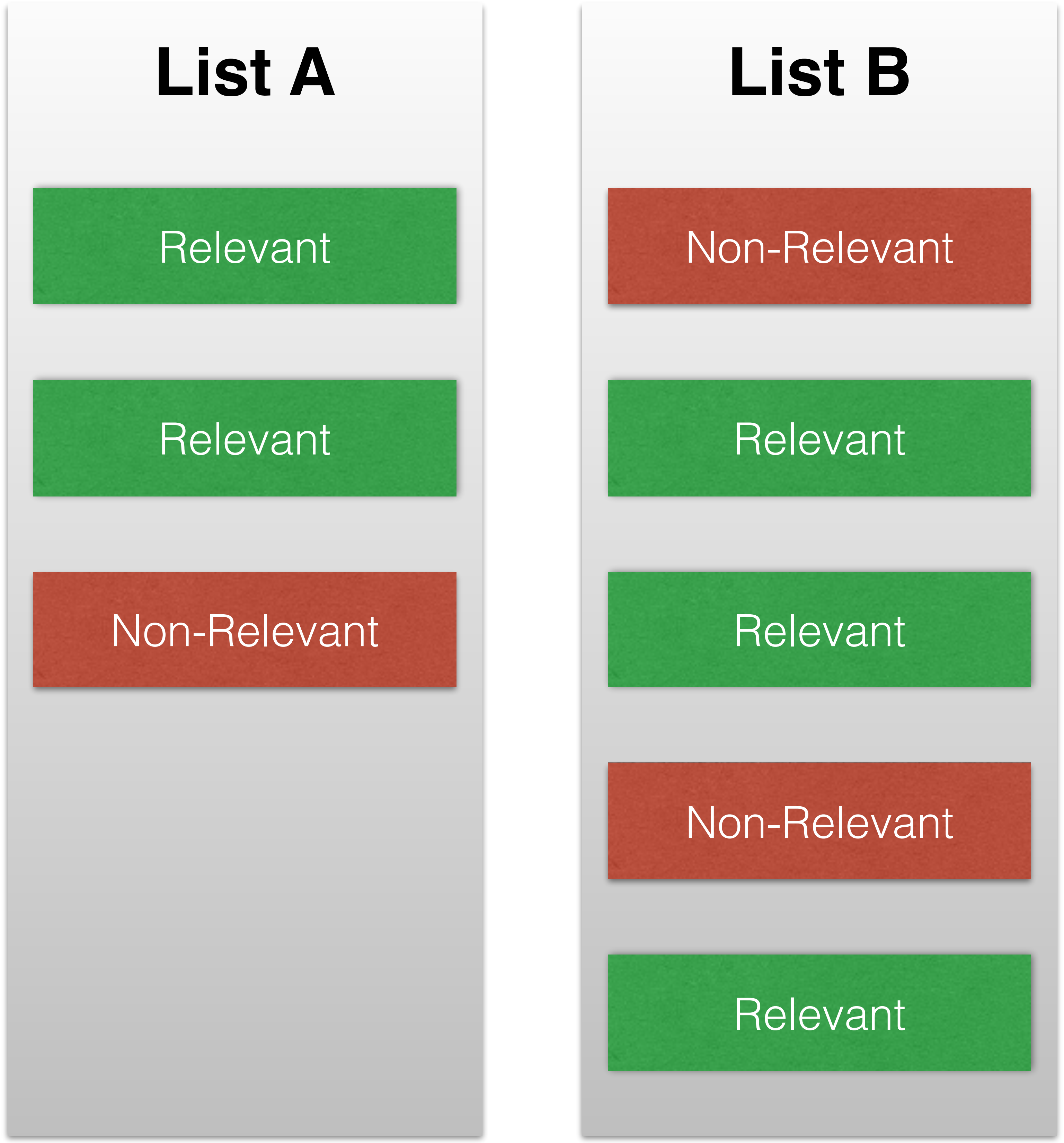
**Confusion Matrix**

$$
\begin{aligned}
precision &:= \frac{num(\textbf{retrieved relevant})}{num(\textbf{retrieved})} \\
&= \frac{TP}{TP + FP}
\end{aligned}
$$

# Precision vs. Recall

There is a tradeoff between recall and precision: usually increasing one will decrease the other.

The quality of a given ranking depends on whether your task is recall- or precision-oriented.



**Which is better? It depends.**

# Wrapping Up

Correct evaluation depends on understanding the nature of the task you're evaluating. For instance, is it recall-oriented or precision-oriented?

Many other factors are also involved, and we'll discuss some of them in future videos.

Next, we'll look at the most commonly-used ways to measure the quality of a ranking.

# Batch Evaluation Measures

Evaluation, session 3

# Relevance Judgements

The goal of retrieval effectiveness evaluation is to rank IR systems.

In order to compare them, we use standard document collections with queries for which relevance judgements have already been collected.

In recall-oriented retrieval, the judgements are typically binary. Precision-oriented retrieval often uses graded relevance judgements.

- **Grade 0:** Non-relevant documents. These documents do not answer the information need.
- **Grade 1:** Somewhat relevant documents. These documents are on the right topic, but have incomplete information.
- **Grade 2:** Relevant documents. These documents do a reasonably good job of answering the query, but the information might be slightly incomplete.
- **Grade 3:** Highly relevant documents. These documents are an excellent reference on the

**Possible Relevance Grade Scheme**

# Relevance Judgement Ambiguity

Expert human judges often disagree on the correct relevance grade for a document.

- They may have different interpretations of the information need.
- They may have different understandings of the document.
- They may disagree on whether a document is “relevant” or “highly relevant.”

However, studies so far suggest that this has a negligible affect on the system ranking.

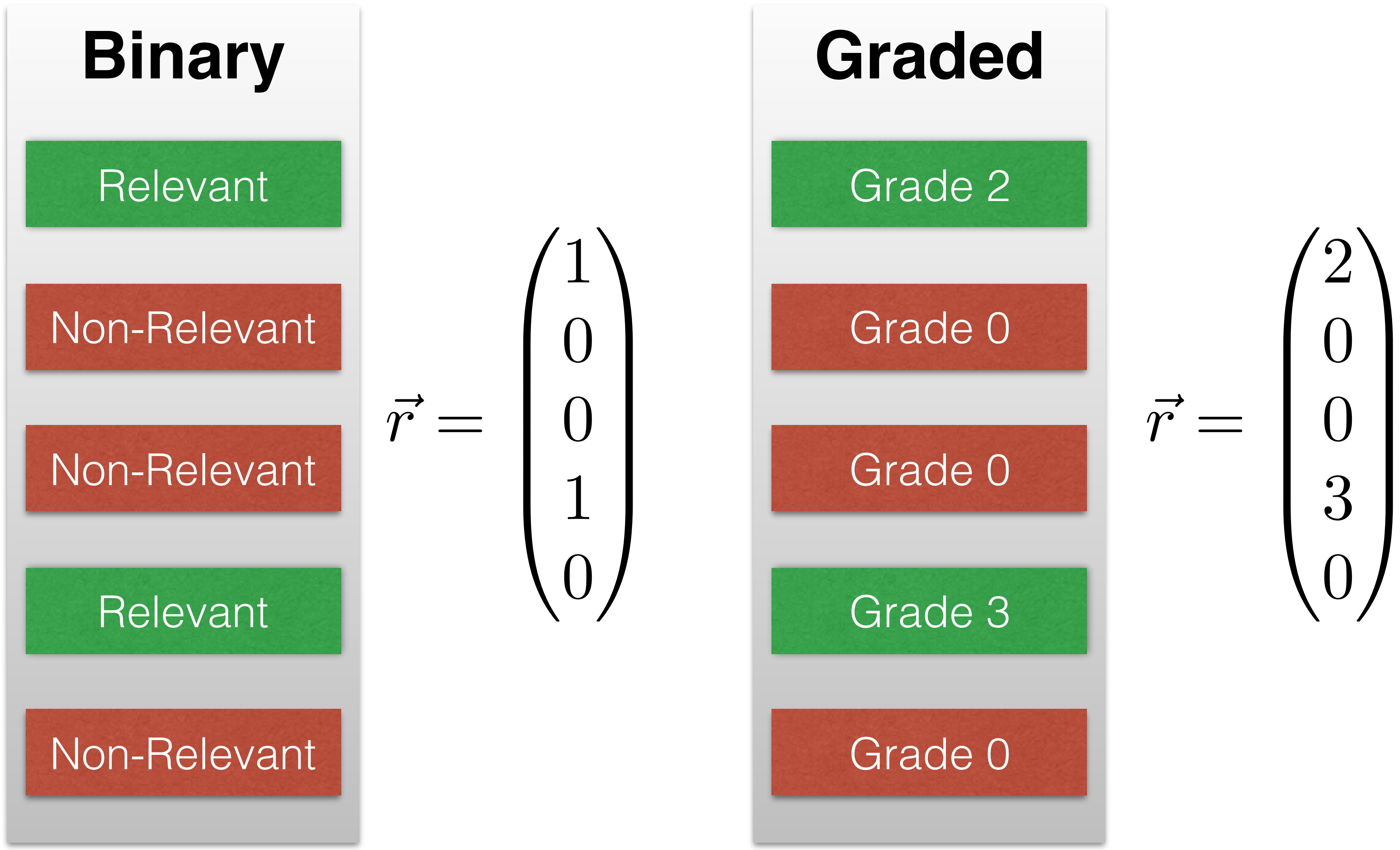
# Evaluating Rankings

Given a ranking and a relevance grade for each ranked document, we build a vector of relevance grades to use for evaluation.

Given a binary vector (and, for recall, the total number of relevant documents $R$):

$$recall(\vec{r}, R) = \frac{1}{R}\sum_{i} \vec{r}_i$$

$$precision(\vec{r}) = \frac{1}{|\vec{r}|}\sum_{i} \vec{r}_i$$



**Binary**

Relevant

Non-Relevant

Non-Relevant

Relevant

Non-Relevant

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

**Graded**

Grade 2

Grade 0

Grade 0

Grade 3

Grade 0

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

**Converting binary and graded rankings into vectors of grades**

# F-Measure

*F-Measure* combines both recall and precision, so systems that favor are penalized for whichever is lower.

The commonly-used *F1-Measure* is the harmonic mean of recall and precision.

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

$$\begin{aligned} F1(\vec{r}, R) &= F(\vec{r}, R, 1) \\ &= \frac{2 \cdot precision(\vec{r}) \cdot recall(\vec{r}, R)}{precision(\vec{r}) + recall(\vec{r}, R)} \end{aligned}$$

**Example**

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

$$\begin{aligned} precision(\vec{r}) &= 0.4 \\ recall(\vec{r}, 20) &= 0.1 \end{aligned}$$

$$\begin{aligned} F1(\vec{r}, 20) &= \frac{2 \cdot 0.4 \cdot 0.1}{0.4 + 0.1} \\ &= 0.16 \end{aligned}$$

# R-Precision

As you move down the ranked list, recall increases monotonically. Precision goes up and down, with a general downward trend.

*R-Precision* is the value of recall and precision at the rank where they are equal.

$$rprecision(\vec{r}, R) := precision@k(\vec{r}, R)$$

Note:

$$precision@k(\vec{r}, R) = recall@k(\vec{r}, R, k)$$

**Example**

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

$$\begin{aligned} precision@k(\vec{r}, k = 5) &= 0.4 \\ recall@k(\vec{r}, R = 5, k = 5) &= 0.4 \\ rprecision(\vec{r}, R = 5) &= 0.4 \end{aligned}$$

# Average Precision

*Average Precision* combines the precision at relevant documents, so it combines recall and precision in a different way.

It is the mean of the $precision@k$ scores for every rank containing a relevant document.

$$ap(\vec{r}, R) = \frac{1}{R} \sum_{k:\vec{r}_k=1} precision@k(\vec{r}, k)$$

**Example**

$$\vec{r} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \qquad prec@k = \begin{pmatrix} 1 \\ 1/2 \\ 1/3 \\ 1/2 \\ 2/5 \end{pmatrix}$$

$$\begin{aligned} ap((1,0,0,1,0)^T, 2) &= \frac{1}{R} \sum_{k:\vec{r}_k=1} precision@k(\vec{r}, k) \\ &= 0.5 \cdot (1 + 0.5) \\ &= 0.75 \end{aligned}$$

# Precision-Recall Curves

A precision-recall curve, or PR-curve, plots precision versus recall at increasing ranks.

The red line is an interpolated version of the plot. It plots recall versus the maximum precision for any higher rank.

AP is approximately the area under the interpolated PR curve. R-precision (rp) is the area under the piecewise linear approximation connecting (0,1) to (rp, rp) and (rp, rp) to (1, 0).

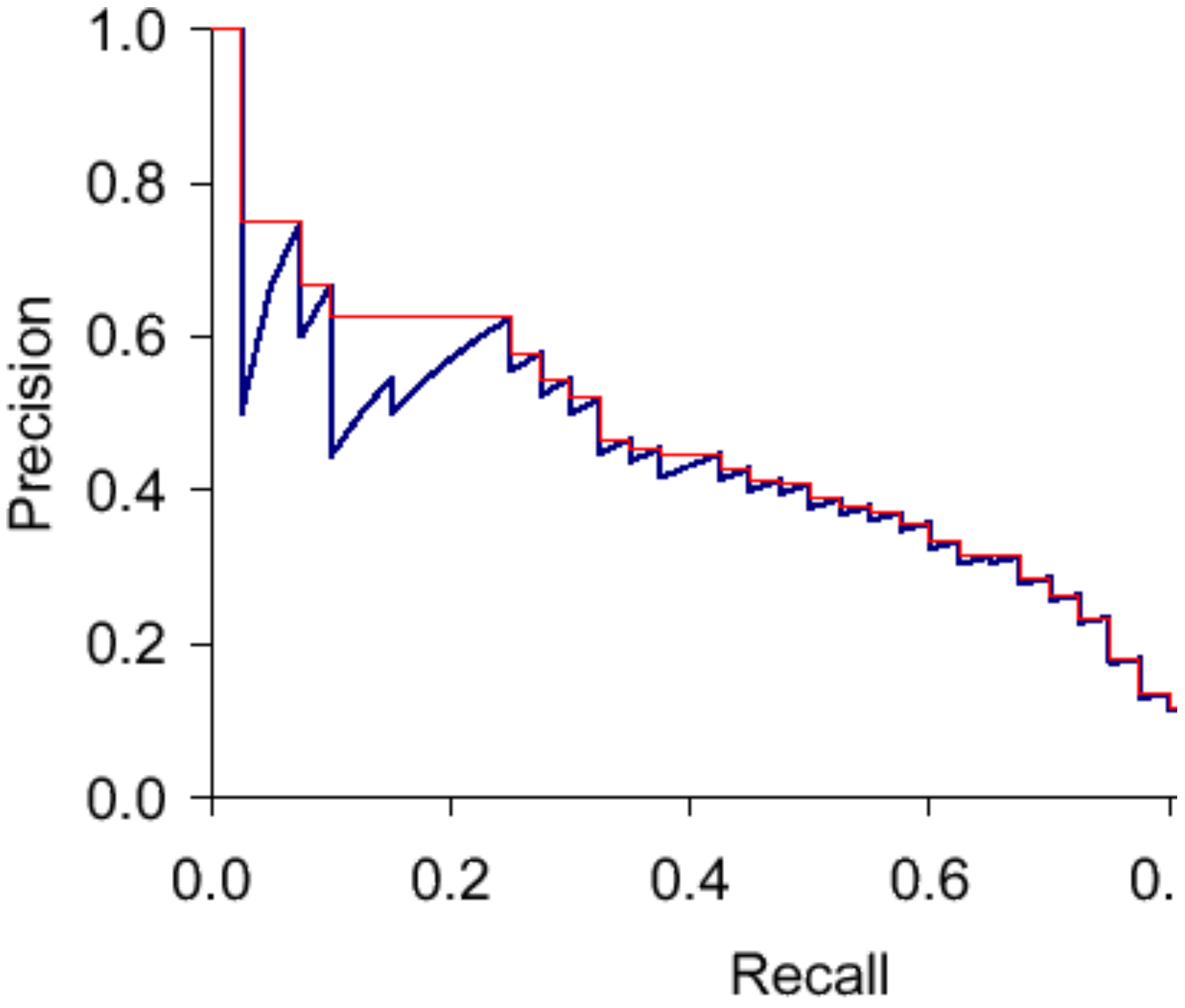


Image courtesy http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-ranked-retrieval-results-1.html

# Reciprocal Rank

*Reciprocal Rank* is the reciprocal of the rank of the first relevant document. It's equivalent to average precision when there is one relevant document.

$$rr(\vec{r}) = \frac{1}{\arg\min_k\{\vec{r}_k \neq 0\}}$$

It's commonly used for evaluating NAV queries, or high-precision queries.

**Example**

$$\vec{r} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \qquad \begin{aligned} rr(\vec{r}) &= \frac{1}{1} \\ &= 1 \end{aligned}$$

# Discounted Cumulative Gain

*DCG* is used for graded relevance judgments, but can't be compared across different queries.

$$dcg(\vec{r}, k) := r_1 + \sum_{i=2}^{k} \frac{r_k}{\lg i}$$

The normalized version, *nDCG*, fixes that by normalizing with the DCG of the sorted ("ideal") list.

$$ndcg(\vec{r}, k) := \frac{dcg(\vec{r}, k)}{dcg(\textsf{sort-desc}(\vec{r}), k)}$$

**Example**

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

$$\begin{aligned} dcg(\vec{r}, 5) &= 2 + \frac{0}{\lg 2} + \frac{0}{\lg 3} + \frac{3}{\lg 4} + \frac{0}{\lg 5} \\ &= 2 + 3/2 \\ &= 3.5 \end{aligned}$$

$$\begin{aligned} ndcg(\vec{r}, 5) &= \frac{dcg(\vec{r}, 5)}{dcg((3, 2, 0, 0, 0)^T, 5)} \\ &= 3.5 / \left(3 + \frac{2}{\lg 2} + \frac{0}{\lg 3} + \frac{0}{\lg 4} + \frac{0}{\lg 5}\right) \\ &= 0.7 \end{aligned}$$

# Wrapping Up

The measures seen here are the most common, but there are many more to choose from. How do you pick?

- F-measure forces you to optimize for both precision and recall, and lets you choose their relative importance.
- RP and AP are recall-oriented, and approximate the area under the PR curve.
- RR and NDCG are precision-oriented. RR is stricter, but NDCG considers more documents in the list.

Next, we'll try to shed some light on what these measures imply about how users interact with a ranked list.

# Modeling Relevance Gain

Evaluation, session 4

# Expected Relevance Gain

All of the measures we've seen so far can be expressed in a different way, based on a user model.

The user model gives the probability of the user reading each document in the ranking.

With these probabilities, we can calculate the expected amount of relevance the user would gain from the ranking.

$$\begin{aligned}
\text{Let } P(i) &:= \text{prob. user reads doc } i \\
R(\vec{r}) &:= \text{fraction of docs user reads} \\
&\quad\;\; \text{from } \vec{r} \text{ which are relevant} \\
\text{Then } gain(\vec{r}) &:= \mathbb{E}_P[R(\vec{r})] \\
&= \sum_{i=1}^{|\vec{r}|} P(i) \cdot r_i
\end{aligned}$$

# Precision@k

$$P_{prec@k}(i) := \begin{cases} 1/k & \text{if } i \leq k \\ 0 & \text{otherwise} \end{cases}$$

For *precision@k*, we model the user as having equal probability of reading each of the top $k$ documents and zero probability of reading anything else.

Is this a reasonable user model?

$$\mathbb{E}_{P_{prec@k}}[R(\vec{r})] = \sum_{i=1}^{|\vec{r}|} P_{prec@k}(i) \cdot r_i$$

$$= \sum_{i=1}^{k} \frac{1}{k} r_i$$

$$= \frac{1}{k} \sum_{i=1}^{k} r_i$$

# Scaled DCG

DCG and nDCG don't normalize easily for this framework, so instead we introduce a related measure: *Scaled DCG*, or sdcg.

This user model is top-weighted: the probability of observing a document is higher for top-ranked documents.

$$P_{sdcg@k}(i) := \begin{cases} 1/Z \cdot 1/\lg(i+1) & \text{if } i \leq k \\ 0 & \text{otherwise} \end{cases}$$

$$Z := \sum_{i=1}^{k} 1/\lg(i+1)$$

$$sdcg@k(\vec{r}) := \sum_{i=1}^{\infty} r_i P_{sdcg@k}(i)$$

$$= \frac{1}{Z} \sum_{i=1}^{k} \frac{r_i}{\lg(i+1)}$$

# Probability of Continuing

So far, we have reconsidered the measures based on the probability of the user observing a document.

It’s sometimes useful to instead consider the probability of the user continuing past a given document. If they read doc *i*, will they read *i+1*?

$$C_M(i) := \frac{P_M(i+1)}{P_M(i)}$$

$$C_{prec@k}(i) := \begin{cases} 1 & \text{if } i < k \\ 0 & \text{otherwise} \end{cases}$$

$$C_{sdcg@k}(i) := \begin{cases} \frac{\lg(i+1)}{\lg(i+2)} & \text{if } i < k \\ 0 & \text{otherwise} \end{cases}$$

# Rank-biased Precision

Rank-biased precision is the measure we get if we imagine that the user has some fixed probability, $p$, of continuing.

This hypothetical user flips a $p$-biased coin at each document to decide when to give up.

On average, this user will read $1 / (1 - p)$ documents before giving up.

$$P_{rbp}(i) := (1 - p)p^{i-1}$$

$$C_{rbp}(i) := p$$

# Inverse Squares

This form of *Inverse Squares* (by Moffat et al 2012) is built on the intuition that the probability of continuing depends on the number of documents the user expects to need to satisfy her information need.

Its parameter $T$ is the anticipated number of documents.

- For nav queries, $T \cong 1$
- For info queries, $T \gg 1$

Let $S_m := \dfrac{\pi^2}{6} - \displaystyle\sum_{i=1}^{m} \dfrac{1}{i^2}$

Then:

$$P_{insq}(T, i) := \frac{1}{S_{2T-1}} \cdot \frac{1}{(i + 2T - 1)^2}$$

$$C_{insq}(T, i) = \frac{(i + 2T - 1)^2}{(i + 2T)^2}$$

# Average Precision

A final way to model user behavior is based on the probability that document *i* is the last document read.

$$L_M(i) := \frac{P_M(i) - P_M(i+1)}{P_M(1)}$$

This gives an interpretation for Average Precision: the expected relevance gained from the user choosing a relevant document *i* uniformly at random, and reading all documents from *1* to *i*.

$$L_{ap}(i) := \begin{cases} r_i/R & \text{if } R > 0 \\ 0 & \text{otherwise} \end{cases}$$

Imagine that exactly one of the relevant documents will satisfy the user, but we don’t know which one.

# Wrapping Up

Evaluation metrics should be carefully chosen to be well-suited to the users and task you’re trying to measure.

Understanding the user model underlying a given metric can help shed light on what you’re really measuring.

Next, we’ll look at the construction and use of test collections.

# Test Collections

Evaluation, session 5

# The Cranfield Paradigm

Librarians have been “finding the information from a collection that is relevant to a user’s information need, as expressed by a query” since long before computers came around.

The rigorous study of Information Retrieval was kicked off by a librarian: Cyril Cleverdon, in the late 1950’s and early 1960’s.

His work at the library of the College of Aeronautics at Cranfield, UK, was the basis of modern IR test collections and evaluation.

**Cyril Cleverdon (1914-1997)**

# Cranfield Results

**Key Findings:**

- Methods for choosing documents, queries, and relevance assessors, and evaluating search tasks on the resulting collection
- The usefulness of indexing individual terms from the document's contents (as opposed to expert-selected topical or category terms)
- The inverse relationship of precision and recall

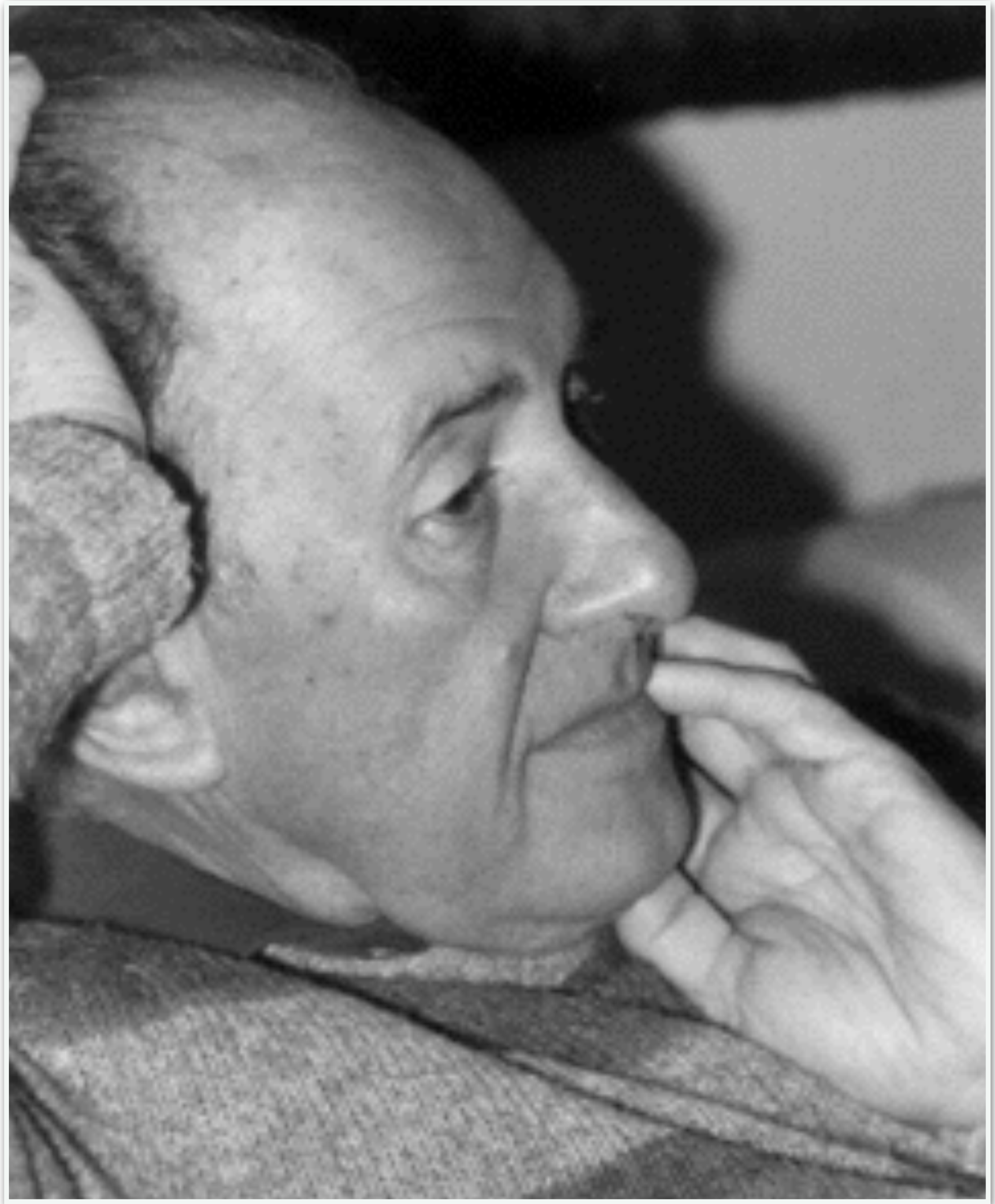
> "Quite the most astonishing and seemingly inexplicable conclusion that arises from the project is that the single term index languages are superior to any other type. …This conclusion is so controversial and so unexpected that it is bound to throw considerable doubt on the methods which have been used to obtain these results, and our first reaction was to doubt the evidence. A complete recheck has failed to reveal any discrepancies, and unless one is prepared to say that the whole test conception is so much at fault that the results are completely distorted, then there is no other course except to attempt to explain the results which seem to offend against every canon on which we were trained as librarians."
>
> – Cyril Cleverdon, 1966

# The SMART System

Gerard Salton was another pioneer of test collection construction. He ran the SMART system at Harvard from 1961-1965, and then at Cornell until his death in 1995.

His team built many test collections for various IR experiments, and produced an enormous number of developments.

Some examples: the vector space model, term weighting, relevance feedback, clustering, term discrimination value, term dependency, text understanding and structuring, passage retrieval…



**Gerard Salton (1927-1995)**

# TREC

The Text REtrieval Conference (TREC), run by NIST in Maryland, has been a major driver of IR research and source of test collections from 1992 onward. Its test collection paradigm is based on the Cranfield model.

The TREC conference ushered in a new era of large-scale IR evaluation, and several language-specific conferences have been created internationally to follow its approach.

**Donna Harman**

**Ellen Vorhees**

# TREC-Style Conferences

| Conference | Focus | Started | URL |
|---|---|---|---|
| TREC | English (mainly) | 1992 | http://trec.nist.gov |
| NTCIR | Asian Languages | 1999 | http://research.nii.ac.jp/ntcir |
| CLEF | European Languages | 2000 | http://www.clef-initiative.eu |
| INEX | Structured Documents | 2002 | https://inex.mmci.uni-saarland.de |
| ROMIP | Russian | 2003 | http://romip.ru/en |
| FIRE | Indian and South Asian Languages | 2008 | http://www.isical.ac.in/~clia |

# Building Collections

1. Identify the research task(s) you want to evaluate with your collection and the distinctive characteristics of the users who carry out this task. Define relevance for your task, and choose appropriate evaluation measures.

2. Obtain and prepare documents suitable for the task. Make sure your document collection is suitably large, and that it contains appropriate levels and types of noise.

3. Create queries for the users and task. Ideally, harvest them from an existing query log, or hire users who carry out this task to create them. The topics must be suitable for the task and documents.

4. Hire relevance assessors to assess the documents for the topics. This is typically done through pooling (addressed next).

5. Validate the resulting collection: look for biases in queries or documents, determine the level of consistency and completeness of the relevance judgements, etc.

# Pooling

In most collections, it’s not feasible to assess the relevance of each document for each query. This is a major challenge that limits the potential collection size. It’s typically addressed through *pooling*.

1. The documents and queries are created.

2. Several IR systems are run on the queries, and each system’s top-ranking 1000 documents are collected into a pool.

3. The resulting pool of documents is assessed in random order, typically by multiple judges.

4. When judges disagree, they meet and discuss the document until they reach consensus.

# Wrapping Up

The right choice of test collection depends on the task you are carrying out. Be careful that the collection's properties are suitable for your needs.

Watch out for:

- The particular definition of relevance used in the collection.
- Are the documents representative of the collection you'll ultimately use?
- Are the users the collection targets representative of your end users?

Next, we'll see how the various test measures compare in various test collections.