# Evaluation

Dorota Glowacka

glowacka@cs.helsinki.fi

# What can we evaluate in IR?

- **coverage** of the collection: extent to which the system includes relevant material
  - this is (was) important in web retrieval (since it was the case that individual search - Altavista, Lycos, etc) engine covers maybe up to 16% of the web space.
- **efficiency** in terms of speed, memory usage, etc.
- **time lag (efficiency):** average interval between the time a request is made and the answer is given
- **presentation** of the output, has to do with interface and visualisation issues.
- **effort** involved by user in obtaining answers to a request
- **recall** of the system: proportion of relevant documents retrieved
- **precision** of the system: proportion of the retrieved documents that are actually relevant

# IR Evaluation: Difficulties

- IR system
  - in: a query
  - out: relevant documents
- Evaluation of IR systems
  - Goal: predict future from past experience
- Reasons why IR evaluation is hard:
  - Large variation in human information needs and queries
  - The precise contributions of each component are hard to entangle:
    - Collection coverage
    - Document indexing
    - Query formulation
    - Matching algorithm

# Cranfield Test Methodology

- Specify a retrieval task
- Create a collection of sample documents
- Create a set of topics/queries appropriate for the retrieval task
- Create a set of relevance judgments (i.e., judgments about which document is relevant to which query)
- Define a set of measures
- Apply a method to (or run a system on) the collection to obtain performance figures

# What counts as an acceptable datset collection?

- In 60s and 70s, very small test collections, arbitrarily different, one per project
  - in 60s: 35 queries on 82 documents
  - in 1990: still only 35 queries on 2000 documents
- not always kept test and training apart as so many environment factors were tested
- TREC-3: 742,000 documents
- Large test collections are needed:
  - to capture user variation
  - to support claims of statistical significance in results
  - to demonstrate that performance levels and differences hold as document file sizes grow
  - commercial credibility
- Practical difficulties in obtaining data; non-balanced nature of the collection

# Today’s Test Collections

A test collection consists of:

- Document set:
  - Large, in order to reflect diversity of subject matter, literary style, noise such as spelling errors
- Queries/Topics:
  - short description of information need
  - TREC “topics”: longer description detailing relevance criteria
  - “frozen” --> reusable
- Relevance judgements:
  - binary
  - done by same person who created the query

# Relevance Judgement

- Relevance is inherently subjective, so we need humans to do them
- Problem: relevance is situational:
  - Information needs are unique to a particular person at a particular time
  - judgements will differ across judges and for the same judge at different times
  - need extensive sampling to counteract natural variation: large populations of users and information needs
- Guidelines given to assessors, in order to define relevance as a reasonably objective property of the document–query pair
  - not fulfillment of information need, not novel information
  - relevance is defined to be irrespective of information contained in other documents (redundancy)
- These guidelines ensure that each relevance decision can be taken independently

# TREC

- Text REtrieval Conference
- Run by NIST (US National Institute of Standards and Technology)
- Began in 1992 as part of the TIPSTER text program
- Marks a new phase in retrieval evaluation
  - common task and data set
  - many participants
  - continuity
- Large test collection: text, queries, relevance judgements

# Sample TREC query

<num> Number: 508
<title> hair loss is a symptom of what diseases

<desc> Description:
Find diseases for which hair loss is a symptom.

<narr> Narrative:
A document is relevant if it positively connects the loss of head hair in humans with a specific disease. In this context, ”thinning hair” and ”hair loss” are synonymous. Loss of body and/or facial hair is irrelevant, as is hair loss caused by drug therapy.

# TREC: relevance agreement

- Queries devised and judged by information specialist (same person)
- Relevance judgements done only for up to 1000 documents/query
- Annotators don’t agree on relevance judgements
- Nevertheless the relative ordering of systems is stable:

  “The comparative effectiveness of different retrieval methods is stable in the face of changes to the relevance judgements” (Vorhees, 2000)

# Pooling

- Pooling (Sparck Jones and van Rijsbergen, 1975)
- Pool is constructed by putting together top *N* retrieval results from a set of *n* systems (TREC: *N* = 100)
- Humans judge every document in this pool
- Documents outside the pool are automatically considered to be irrelevant
- There is overlap in returned documents: pool is smaller than theoretical maximum of *N x n* systems (around 1/3 the maximum size)
- Pooling works best if the approaches used are very different
- Large increase in pool quality by manual runs which are recall oriented, in order to supplement pools

# Validity of relevance assessment

- Relevance assessments are only usable if they are consisten
- If they are not consistent, then there is no ground truth and the experiments are not repeatable
- How can we measure this consistency or agreement among judges?
- Kappa measure for inter-assessor (dis)agreement
  - Agreement measure among assessors
  - Designed for categorical judgments
  - Corrects for chance agreement
    - P(A) – proportion that judges agree
    - P(E) – what agreement would be by chance

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

- Kappa = 0 for chance agreement, Kappa = 1 for total agreement

# Kappa measure: example

judge 2 relevance

judge 1 relevance

| | Yes | No | Total |
|---|---|---|---|
| Yes | 300 | 20 | 320 |
| No | 10 | 70 | 80 |
| Total | 310 | 90 | 400 |

$$P(A) = \frac{300 + 70}{400} = 0.925$$

$$P(E) = \left(\frac{80 + 90}{400 + 400}\right)^2 + \left(\frac{320 + 310}{400 + 400}\right)^2 = 0.2125^2 + 0.7878^2 = 0.665$$

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)} = \frac{0.925 - 0.665}{1 - 0.665} = 0.776$$

# Standard relevance benchmarks: other

- GOV2
  - Another TREC/NIST collection
  - 25 million web pages
  - Used to be the largest collection that is easily available
- NTCIR
  - East Asian languages and cross-language information retrieval
- Cross Language Evaluation Forum (CLEF)
  - This evaluation series focuses on European languages and cross-language information retrieval

# System Oriented Evaluation

- **Test collection** methodology:
  - Benchmark (data set) upon which effectiveness is measured and compared
  - Data that tell us for a given query what are the relevant documents.
- Measuring **effectiveness** has been the most predominant in IR evaluation:
  - **recall** of the system: proportion of relevant documents retrieved
  - **precision** of the system: proportion of the retrieved documents that are actually relevant
- Looking at these two aspects is part of what is called **system-oriented evaluation.**

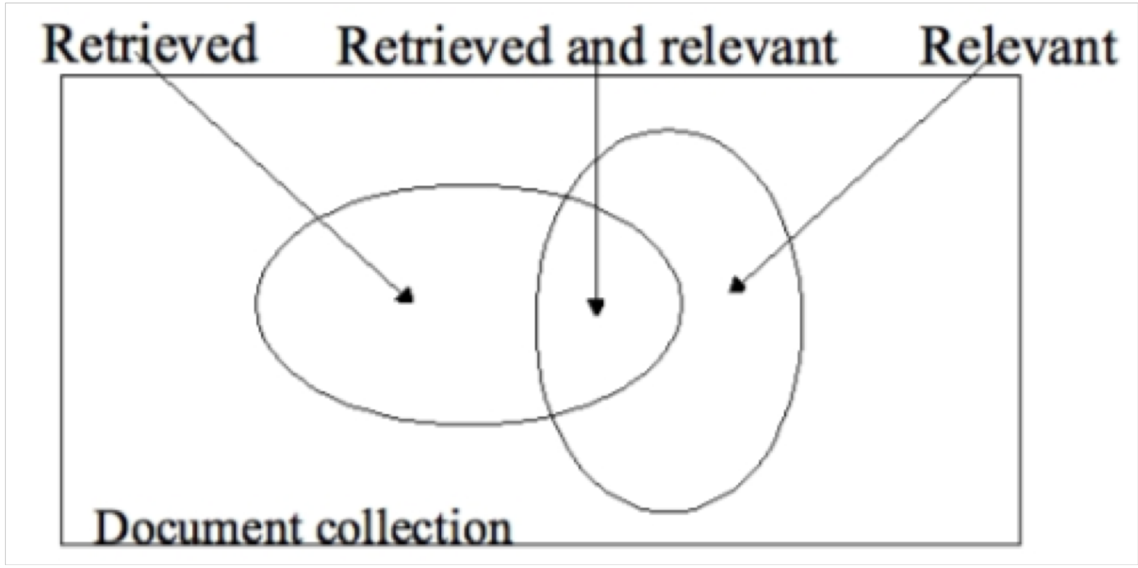
# Effectiveness

- We recall that the goal of an IR system is to retrieve as many relevant documents as possible and as few non-relevant documents as possible.

- Evaluating the above consists of a comparative evaluation of technical performance of IR system(s):
    - In traditional IR, technical performance means the effectiveness of the IR system: the ability of the IR system to retrieve relevant documents and suppress non-relevant documents
    - Effectiveness is measured by the combination of recall and precision.

# Recall/Precision

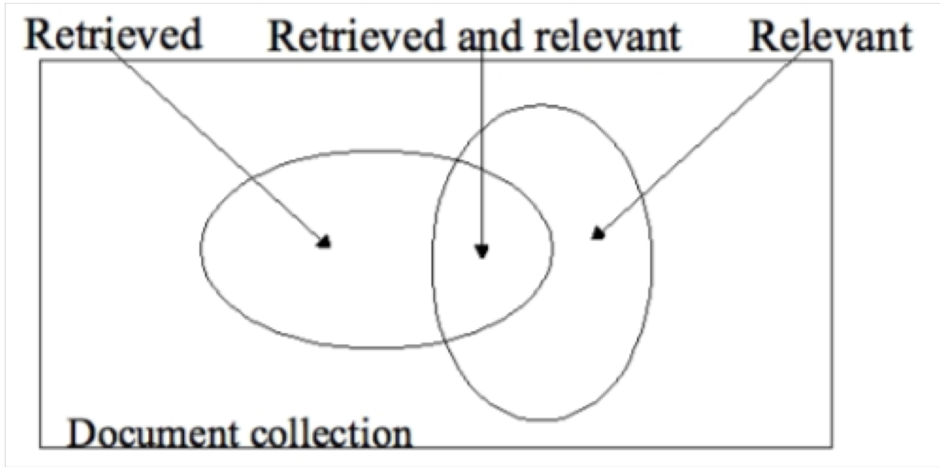
For a given query, the document collection can be divided into three sets: the set of retrieved document, the set of relevant documents, and the rest of the documents.




Note: knowing which documents are relevant comes from the test collection

# Recall/Precision

In the ideal case, the set of retrieved documents is equal to the set of relevant documents. However, in most cases, the two sets will be different. This difference is formally measured with **precision** and **recall**.




$$Precision = \frac{\text{number of relevant documents retrieved}}{\text{number of documents retrieved}}$$

$$Recall = \frac{\text{number of relevant documents retrieved}}{\text{number of relevant documents}}$$

# A combined measure: *F*

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

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

| | relevant | not relevant |
|---|---|---|
| retrieved | 20 | 40 |
| not retrieved | 60 | 1,000,000 |
| | 80 | 1,000,040 |

P = 20/(20 + 40) = 1/3
R = 20/(20 + 60) = 1/4

$$F_1 = 2\frac{1}{\frac{1}{\frac{1}{3}} + \frac{1}{\frac{1}{4}}} = 2/7$$

# Exercise

- Compute precision, recall and F1 for this set of results:

| | relevant | not relevant |
|---|---|---|
| retrieved | 18 | 2 |
| not retrieved | 82 | 1,000,000,000 |

# E measure (parametrized F measure)

- Variant of F measure that allows weighting of 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 $\beta$ controls the trade-off:
  - $\beta = 1$: equally weight precision and recall (E = F)
  - $\beta > 1$: weight recall more
  - $\beta < 1$: weight precision more

# Precision vs. Recall

- Inverse relationship between precision and recall forces general systems to go for compromise between them
- But some tasks particularly need good precision whereas others need good recall:

| Precision-critical task | Recall-critical task |
| --- | --- |
| Little time available | Time matters less |
| A small set of relevant documents answers the information need | One cannot afford to miss a single document |
| Potentially many documents might fill the information need (redundantly) | Need to see each relevant document |
| Example: web search for factual information | Example: patent search |

# Recall/Precision

$$Precision = \frac{\text{number of relevant documents retrieved}}{\text{number of documents retrieved}}$$

$$Recall = \frac{\text{number of relevant documents retrieved}}{\text{number of relevant documents}}$$

The above two measures do not take into account where the relevant documents are retrieved, this is, at which rank (crucial since the output of most IR systems is a ranked list of documents).

This is very important because an effective IR system should not only retrieve as many relevant documents as possible and as few non-relevant documents as possible, but also it should retrieve relevant documents **before** the non-relevant ones.

# Recall/Precision

- Let us assume that for a given query, the following documents are relevant (10 relevant documents):
{d3, d5, d9, d25, d39, d44, d56, d71, d89, d123}
- Now suppose that the following documents are retrieved for that query:

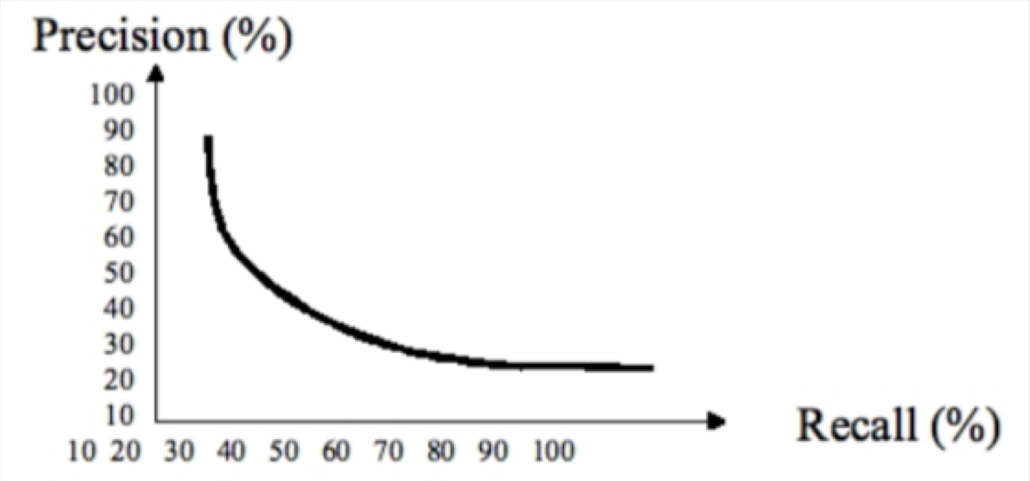
| rank | doc | precision | recall | rank | doc | precision | recall |
|---|---|---|---|---|---|---|---|
| 1 | **d123** | 1/1 | 1/10 | 8 | d129 | | |
| 2 | d84 | | | 9 | d187 | | |
| 3 | **d56** | 2/3 | 2/10 | 10 | **d25** | 4/10 | 4/10 |
| 4 | d6 | | | 11 | d48 | | |
| 5 | d8 | | | 12 | d250 | | |
| 6 | **d9** | 3/6 | 3/10 | 13 | d113 | | |
| 7 | d511 | | | 14 | **d3** | 5/14 | 5/10 |

- For each relevant document (in red bold), we calculate the precision value and the recall value. For example, for d56, we have 3 retrieved documents, and 2 among them are relevant, so the precision is 2/3. We have 2 of the relevant documents so far retrieved (the total number of relevant documents being 10), so recall is 2/10.

# Recall/Precision

- For each query, we obtain pairs of recall and precision values
  - In our example, we would obtain (1/10, 1/1) (2/10, 2/3) (3/10, 3/6) (4/10, 4/10) (5/10, 5/14) . . . which are usually expressed in % (10%, 100%) (20%, 66.66%) (30%, 50%) (40%, 40%) (50%, 35.71%) . . .
  - This can be read for instance: at 20% recall, we have 66.66% precision; at 50% recall, we have 35.71% precision

The pairs of values are plotted into a graph, which has the following curve:



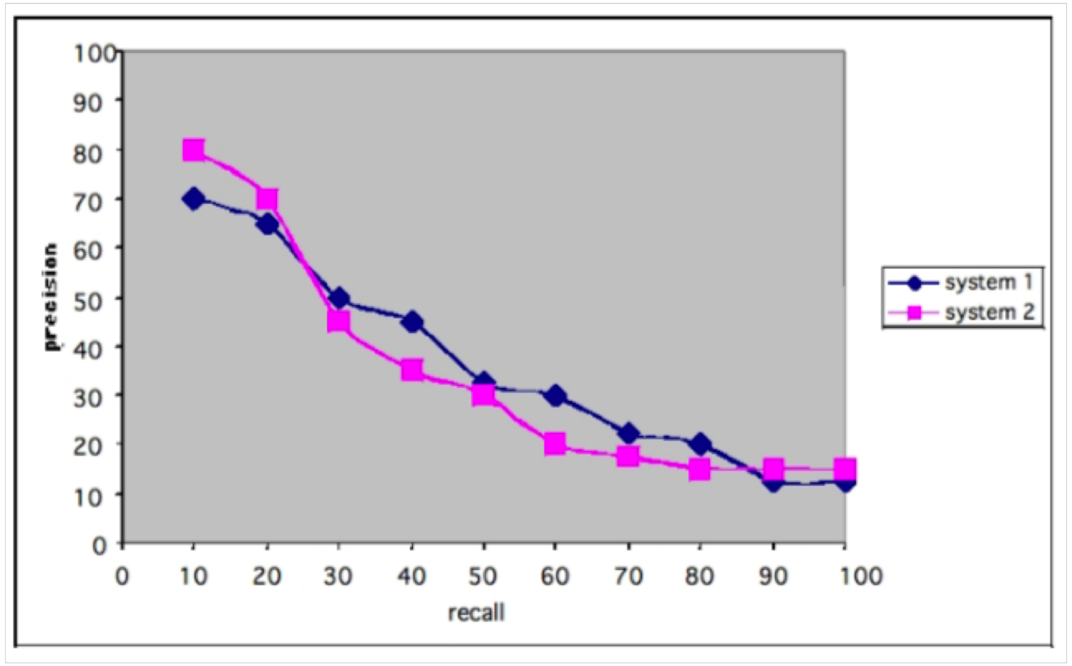
# The Complete Methodology

For each IR system / IR system version:

- For each query in the test collection
  - We first run the query against the system to obtain a ranked list of retrieved documents
  - We use the ranking and relevance judgements to calculate recall/precision pairs
- Then we average recall / precision values across all queries, to obtain an overall measure of the effectiveness.

# Comparison of Systems

We can compare IR systems / system versions. For example, here we see that at low recall, system 2 is better than system 1, but this changes from recall value 30%, etc. It is common to calculate an average precision value across all recall levels, so that to have a single value to compare, so called **Mean Average Precision** (MAP).



# Averaging

| Recall in % | Precision in % | | |
|---|---|---|---|
| | Query 1 | Query 2 | Average |
| **10** | 80 | 60 | **70** |
| **20** | 80 | 50 | **65** |
| **30** | 60 | 40 | **50** |
| **40** | 60 | 30 | **45** |
| **50** | 40 | 25 | **32.5** |
| **60** | 40 | 20 | **30** |
| **70** | 30 | 15 | **22.5** |
| **80** | 30 | 10 | **20** |
| **90** | 20 | 5 | **11.5** |
| **100** | 20 | 5 | **11.5** |

The same information can be displayed in a plot.

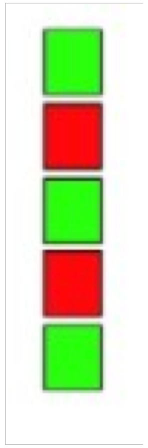
# Rank-Based Measures

- Binary relevance
  - Precision@K (P@K)
  - Mean Average Precision (MAP)
  - Mean Reciprocal Rank (MRR)

- Multiple levels of relevance
  - Normalized Dsicounted Cumulative Gain

# Precision@K

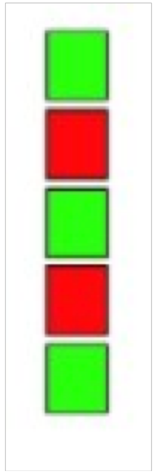
- Set a rank threshold K
- Compute % of relevant documents in top K
- Ignore documents ranked lower than K
- Example:
  - Prec@3 of 2/3
  - Prec@4 of 2/4
  - Prec@5 of 3/5



- In a similar fashion, we have Recall@K

# Mean Average Precision (MAP)

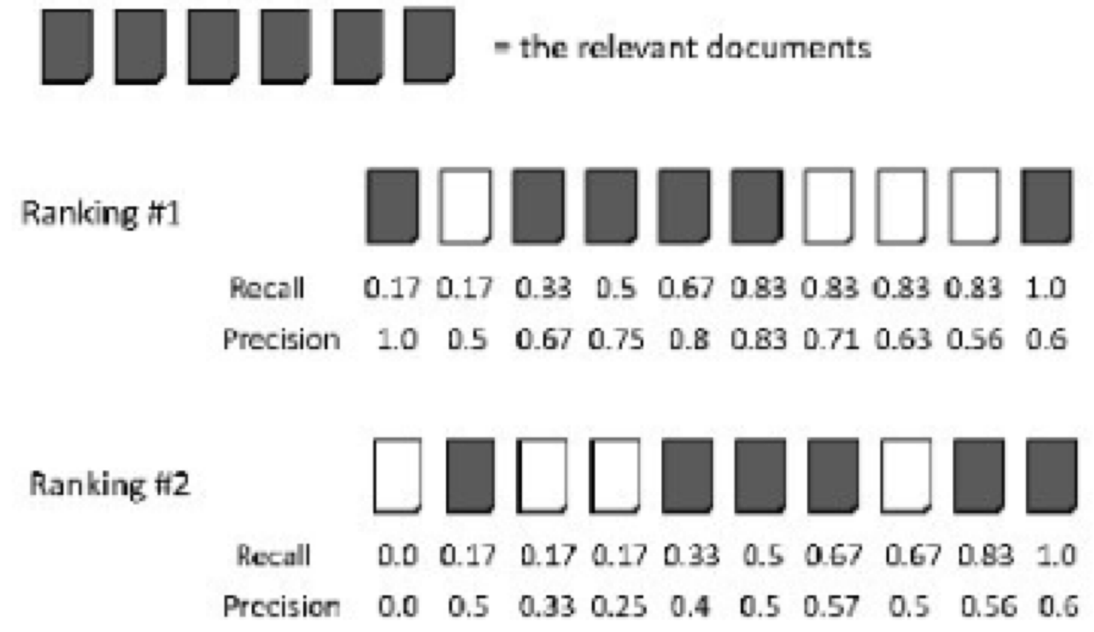
- Consider rank position of each **relevant** document, i.e. $K_1, K_2, \ldots, K_R$
- Compute Precision@K for each $K = K_1, K_2, \ldots, K_R$
- Average precision = average of Precision@K



Has average precision of 1/3 x (1/1 + 2/3 + 3/5) = 0.76

- MAP is Average Precision across multiple queries/rankings/systems

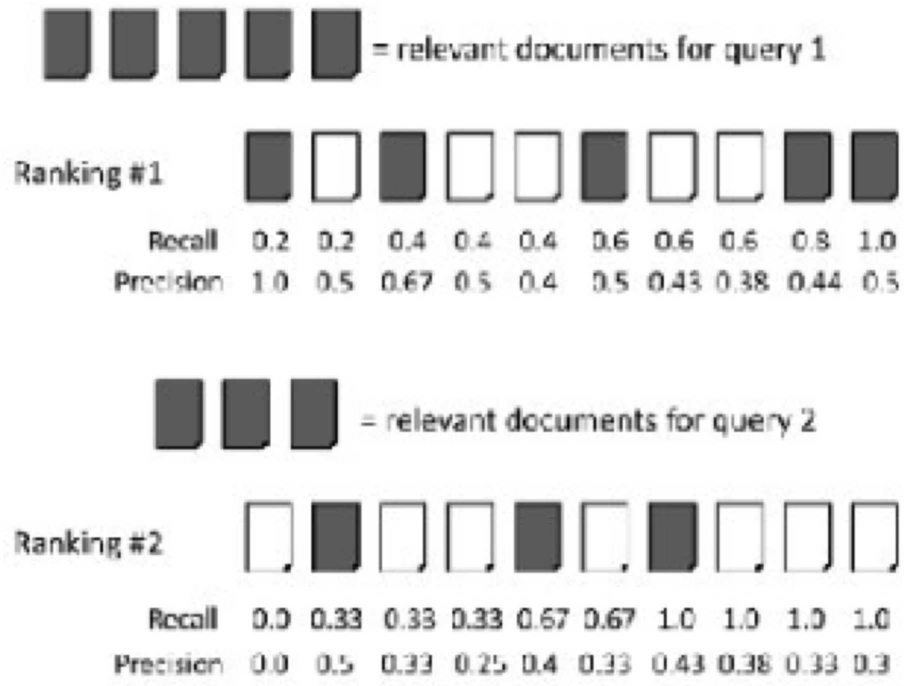
# Average Precision




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

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

# Mean Average Precision (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 document to be zero.
- MAP is macro-averaging: each query counts equally
- One of the most commonly used measures in research papers
- MAP assumes user is interested in finding many relevant documents for each query
- MAP requires many relevance judgements in text collections

# Discounted Cumulative Gain

- Popular measure to evaluate 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 becuase it is less likely to be examined.
- Uses ***graded relevance*** as a measure of usefulness, or ***gain***, from examining the 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 (rank): with base 2, the discount at rank 4 is ¼ and at rank 8 it is 1/3.

# Discounted Cumulative Gain

What if the relevance judgements are on a scale of [0, r], where r > 2?

- Cumulative Gain (CG) at rank *p*:
  - Let the ratings of the n documents be $r_1, r_2, \ldots, r_p$ (in ranked order)
  - CG = $r_1 + r_2 + \ldots + r_p$
- Discounted Cumulative Gain (DCG) at rank *p*:
  - DCG = $r_1 + r_2/\log_2 2 + r_3/\log_2 3 + \ldots + r_p/\log_2 p$ (we may use any log base)
- DCG is the total gain accumulated at rank *p*:

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

# 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

# Normalized Discounted Cumulative Gain (NDCG)

- Normalize DCG at rank *p* by the DCG value at rank p 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 d1, d2, d3, d4

| i | Ground Truth | | Ranking $\text{Function}_1$ | | Ranking $\text{Function}_2$ | |
|---|---|---|---|---|---|---|
| | 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 | |

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

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

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

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

# NDCG (at 4): Example

- Graded ranking/ordering:

| | | | |
|---|---|---|---|
| 4 | 2 | 0 | 1 |

- DCG = 4 + 2/log(2) + 0/log(3) + 1/log(4)
  - = 6.5
- IDCG = 4 + 2/log(2) + 1/log(3) + 0/log(4)
  - = 6.63
- NDCG = DCG/IDCG = 6.5/6.63 = .98

# Limitations of NDCG

- NDCG does not penalize for bad documents in the result list, e.g. if a query returns two results with scores 1, 1, 1 and 1, 1, 1, 0, then both would be considered equally good.
- NDCG does not penalize for missing documents in the result list. For example, if a query returns two results with scores 1,1,1 and 1,1,1,1,1, both would be considered equally good, assuming ideal DCG is computed to rank 3 for the former and rank 5 for the latter.
- NDCG may not be suitable to measure performance of queries that may often have several equally good results, especially when looking only at the first few results as it is done in practice. For example, for queries such as "restaurants" nDCG@1 would account for only the first result and hence if one result set contains only 1 restaurant from the nearby area while the other contains 5, both would end up having the same score even though the latter is more comprehensive.

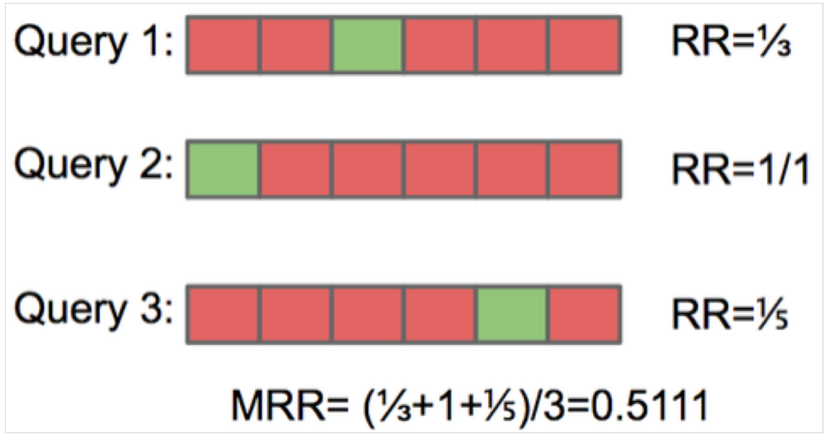
# What if there is only one relevant document?

- The user is interested in only one specific document/item.
- The assumption is that the user will keep going down the results list until he finds the one relevant document.
- If the document is found at rank *p*, the quality of the search is measured by the reciprocal of the rank, i.e. *1/p*
- This measures the user's effort
- Scenarios:
  - Known-item search
  - Navigational queries
  - Factual queries, e.g. *What is the capital of Australia?*

# Mean Reciprocal Rank (MRR)

- MRR evaluates systems that produce a list of ranked items for queries
- The reciprocal rank is the multiplicative inverse of the rank of the first correct item
- For calculating MRR, the items don’t need to be rated.
- MRR doesn’t apply if there are multiple correct responses (hits) in the resulting list

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}$$



# Large search engine evaluation

- Recall is difficult to measure on the web
- Search engines often use precision at top k (Precision@K)
- … or measures that prioritise getting rank 1 right than getting rank 10 right (NDCG)
- Search engines also use non-relevance based measures:
  - Clickthrough on first result
    - Not very reliable if you look at a single user but quite reliable in the aggregate
  - Analysing search logs
  - Studies of user behaviour in the lab
  - A/B testing

# A/B testing

Two-sample hypothesis testing

- Two versions of a system (A and B) are compared, which are identical except for one variation that might affect a user's behaviour, e.g. two different font types

- Randomized experiment
  - Separate the population into equal size groups, e.g. 10% random users for system A and 10% random users for system B
  - Null hypothesis: no difference between system A and B

# Behaviour-based measures

- **Abandonment rate**: fraction of queries for which no results were clicked on
- **Reformulation rate**: fraction of queries that were followed by another query during the same search session
- **Queries per session**: mean number of queries issued by a user during a search session
- **Clicks per query**: mean number of results clicked for each query
- **Time to first click**: mean time from query being issued until first click on any result
- **Time to last click**: mean time being issued until last click on any result

# Behaviour-based metrics

When search results become **worse**:

| Metric | Change as ranking gets worse |
|---|---|
| *Abandonment rate* | Increase (more bad result sets) |
| *Reformulation rate* | Increase (more need to reformulate) |
| *Queries per session* | Increase (more need to reformulate) |
| *Clicks per query* | Decrease (fewer relevant results) |
| *Max recip. rank* | Decrease (top results are worse) |
| *Mean recip. rank* | Decrease (more need for many clicks) |
| *Time to first click* | Increase (good results are lower) |
| *Time to last click* | Decrease (fewer relevant results) |

# A/B testing: constructing comparison systems

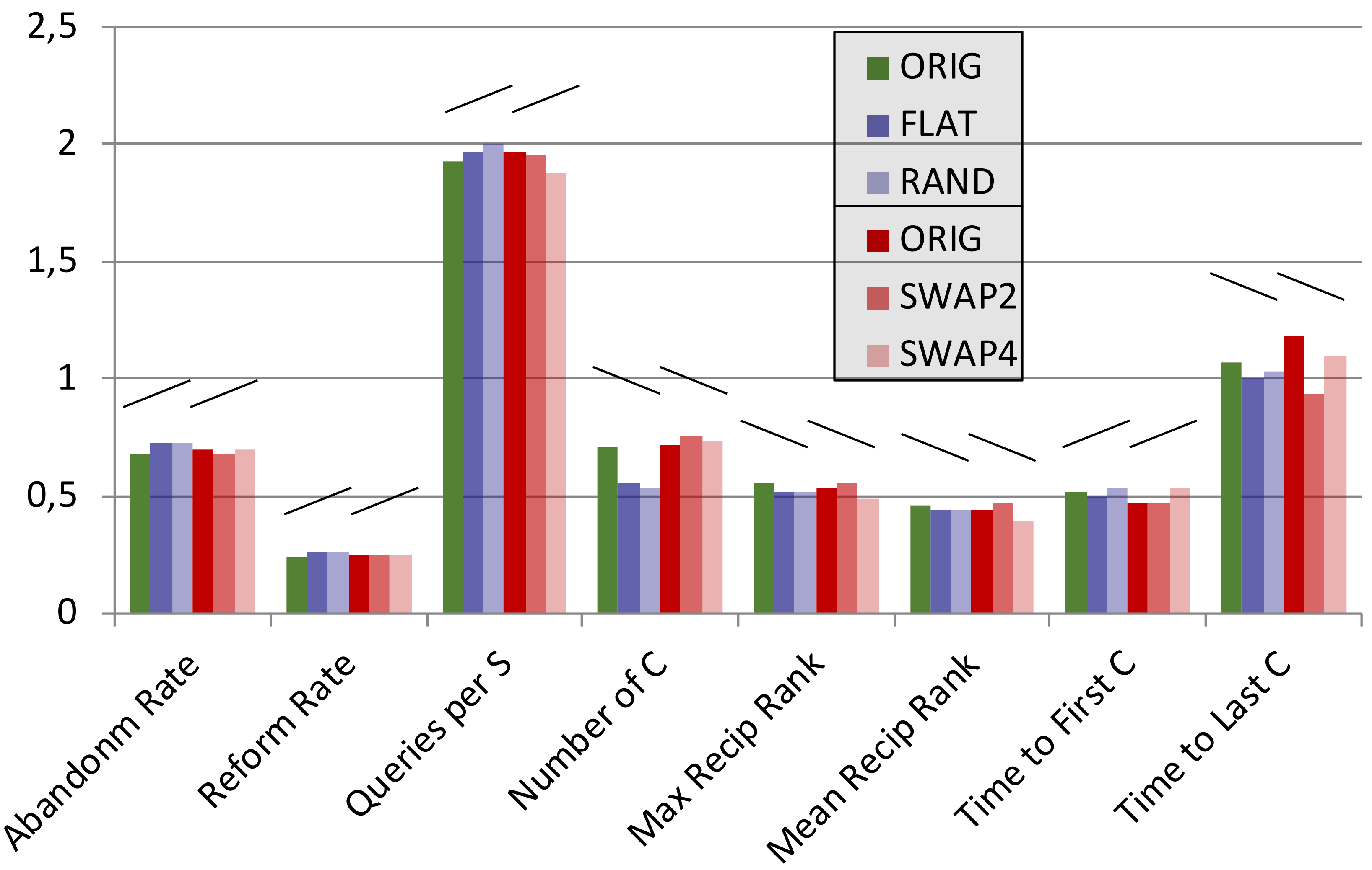
- Orig > Flat > Rand
  - Orig: original ranking algorithm from arXiv.org
  - Flat: no field weights
  - Rand: random shuffle of top 10 Flat's results

- Orig > Swap2 > Swap4
  - Swap2: Orig with 2 pairs swapped
  - Swap4: Orig with 4 pairs swapped

Do all pairwise tests
Evaluation on 3500 x 6 queries

[Radlinski et al. 2008]

# Evaluation of Absolute Metrics on ArXiv.org




[Radlinski et al. 2008]

# Results for A/B test

1/6 users of arXiv.org are routed to each of the testing systems in one month period

| | $\mathcal{H}_1$ | ORIG≻FLAT≻RAND | | |
|---|---|---|---|---|
| | | ORIG | FLAT | RAND |
| Abandonment Rate (Mean) | < | $0.680 \pm 0.021$ | $0.725 \pm 0.020$ | $0.726 \pm 0.020$ |
| Reformulation Rate (Mean) | < | $0.247 \pm 0.021$ | $0.257 \pm 0.021$ | $0.260 \pm 0.021$ |
| Queries per Session (Mean) | < | $1.925 \pm 0.098$ | $1.963 \pm 0.100$ | $2.000 \pm 0.115$ |
| Clicks per Query (Mean) | > | $0.713 \pm 0.091$ | $0.556 \pm 0.081$ | $0.533 \pm 0.077$ |
| Max Reciprocal Rank (Mean) | > | $0.554 \pm 0.029$ | $0.520 \pm 0.029$ | $0.518 \pm 0.030$ |
| Mean Reciprocal Rank (Mean) | > | $0.458 \pm 0.027$ | $0.442 \pm 0.027$ | $0.439 \pm 0.028$ |
| Time (s) to First Click (Median) | < | $31.0 \pm 3.3$ | $30.0 \pm 3.3$ | $32.0 \pm 4.0$ |
| Time (s) to Last Click (Median) | > | $64.0 \pm 19.0$ | $60.0 \pm 14.0$ | $62.0 \pm 9.0$ |

# Results for A/B test

1/6 users of arXiv.org are routed to each of the testing systems in one month period

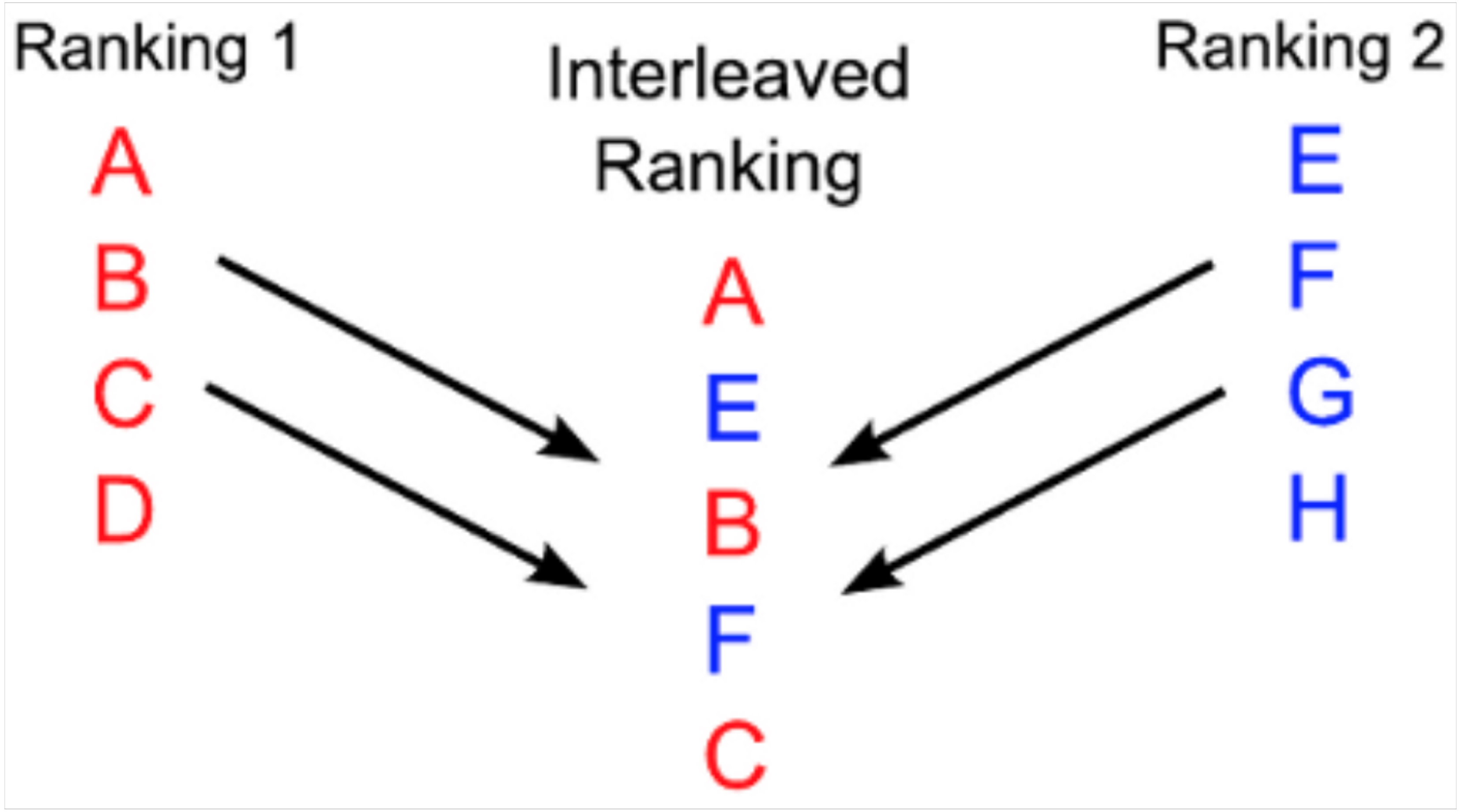
| | $\mathcal{H}_1$ | ORIG≻SWAP2≻SWAP4 | | |
|---|---|---|---|---|
| | | ORIG | SWAP2 | SWAP4 |
| Abandonment Rate (Mean) | < | $0.704 \pm 0.021$ | $0.680 \pm 0.021$ | $0.698 \pm 0.021$ |
| Reformulation Rate (Mean) | < | $0.248 \pm 0.021$ | $0.250 \pm 0.021$ | $0.248 \pm 0.021$ |
| Queries per Session (Mean) | < | $1.971 \pm 0.110$ | $1.957 \pm 0.099$ | $1.884 \pm 0.091$ |
| Clicks per Query (Mean) | > | $0.720 \pm 0.098$ | $0.760 \pm 0.127$ | $0.734 \pm 0.125$ |
| Max Reciprocal Rank (Mean) | > | $0.538 \pm 0.029$ | $0.559 \pm 0.028$ | $0.488 \pm 0.029$ |
| Mean Reciprocal Rank (Mean) | > | $0.444 \pm 0.027$ | $0.467 \pm 0.027$ | $0.394 \pm 0.026$ |
| Time (s) to First Click (Median) | < | $28.0 \pm 2.2$ | $28.0 \pm 3.0$ | $32.0 \pm 3.5$ |
| Time (s) to Last Click (Median) | > | $71.0 \pm 19.0$ | $56.0 \pm 10.0$ | $66.0 \pm 15.0$ |

Overall result: most differences not significant and none of the absolute metrics reliably reflect expected order

# Interleaved Ranking

Directly asking the user which of the ranking methods is better

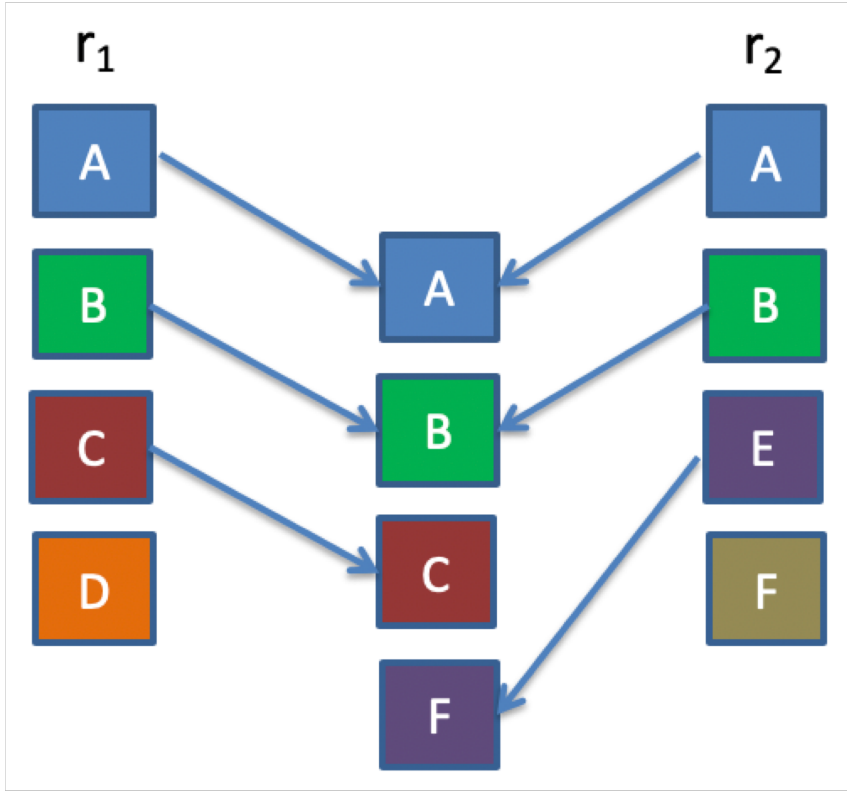
Randomized experiments:
- Interleave results from rankings A and B
- Give interleaved results to the same population and ask for their preference
- We can interpret clicks as users’ preference judgements



# Interleaved Ranking

Scoring interleaved ranking:

- Clicks credited to ”owner” of the result, i.e. ranking 1 or ranking 2
- Ranking with more credits wins
- Rankings share top K results when they have identical results at each rank 1 … K



# Intearleave for IR evaluation

## Team-draft interleaving

**Input**: Rankings $A = (a_1, a_2, \dots)$ and $B = (b_1, b_2, \dots)$
**Init**: $I \leftarrow (); TeamA \leftarrow \emptyset; TeamB \leftarrow \emptyset;$
**while** $(\exists i : A[i] \notin I) \wedge (\exists j : B[j] \notin I)$ **do**
  **if** $(|TeamA| < |TeamB|) \vee$
    $((|TeamA| = |TeamB|) \wedge (RandBit() = 1))$ **then**
    $k \leftarrow \min_i \{i : A[i] \notin I\}$ ..... *top result in A not yet in I*
    $I \leftarrow I + A[k];$ ........................ *append it to I*
    $TeamA \leftarrow TeamA \cup \{A[k]\}$ ...... *clicks credited to A*
  **else**
    $k \leftarrow \min_i \{i : B[i] \notin I\}$ ..... *top result in B not yet in I*
    $I \leftarrow I + B[k]$ ........................ *append it to I*
    $TeamB \leftarrow TeamB \cup \{B[k]\}$ ...... *clicks credited to B*
  **end if**
**end while**
**Output**: Interleaved ranking $I$, $TeamA$, $TeamB$

# Results for interleaved test (arXiv experiment)

- 1/6 users of arXiv.org are routed to each of the testing system in one month period; test which group receives more clicks

| Comparison Pair A ≻ B | Query Based | | | User Based | | |
|---|---|---|---|---|---|---|
| | A wins | B wins | # queries | A wins | B wins | # users |
| ORIG ≻ FLAT | **47.7%** | **37.3%** | 1272 | **49.6%** | **36.0%** | 667 |
| FLAT ≻ RAND | **46.7%** | **39.7%** | 1376 | **46.3%** | **36.8%** | 646 |
| ORIG ≻ RAND | **55.6%** | **29.8%** | 1095 | **58.7%** | **28.6%** | 622 |
| ORIG ≻ SWAP2 | 44.4% | 40.3% | 1170 | **44.7%** | **37.4%** | 693 |
| SWAP2 ≻ SWAP4 | 44.2% | 40.3% | 1202 | 45.1% | 39.8% | 703 |
| ORIG ≻ SWAP4 | **47.7%** | **37.8%** | 1332 | **47.2%** | **35.0%** | 697 |

- Interleaved test is more accurate and sensitive than A/B testing (9 out of 12 experiments follow our expectation)
- Only click count is suffcient

# Benefits & Drawbacks of Interleaving

- Benefits
  - A more direct way to elicit user preferences
  - A more direct way to perform retrieval evaluation
  - Deals with issues of position bias and calibration
- Drawbacks
  - Reusability: Can only elicit pairwise preferences for specific pairs of ranking functions
  - Benchmark: No absolute number for benchmarking
  - Interpretation: Unable to interpret much at the document-level, or about user behavior