THE UNIVERSITY
*of* EDINBURGH

# Text Technologies for Data Science

## INFR11145

# IR Evaluation


Instructor:
**Youssef Al Hariri**


18-Oct-2023

# Pre-lecture

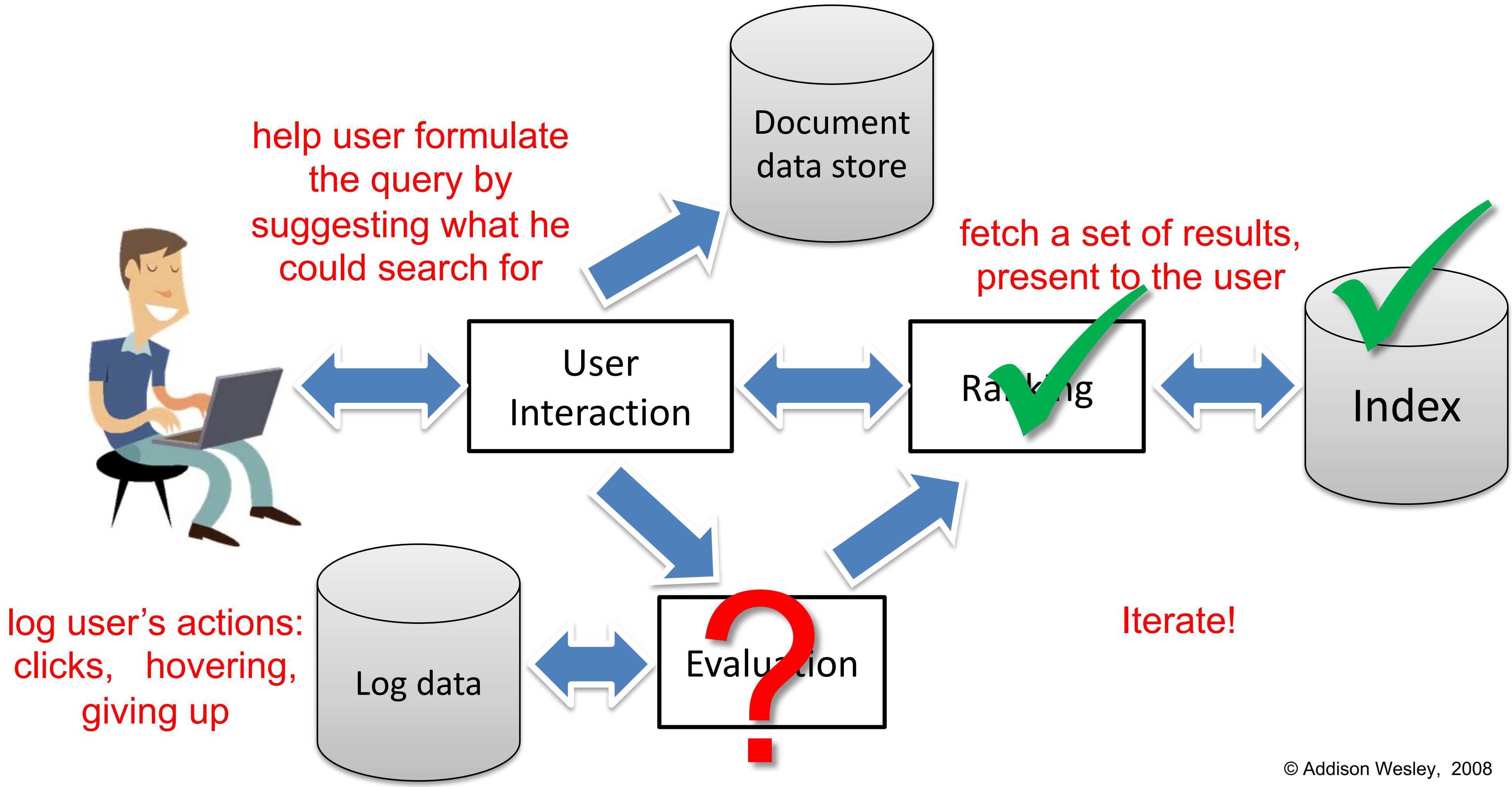
- How working on labs and CW going?
- Thanks for sharing lab results on Piazza
- Test collection for CW1 to be  released next week
- No new lab this week (support to continue for previous labs)
- Today: long L1 and short L2

# Lecture Objectives

- Learn about how to evaluate IR
  - Evaluation measures
  - P, R, F
  - MAP
  - nDCG

- Implement: (as part of CW2)
  - P, R
  - MAP
  - nDCG

# Search Process








THE UNIVERSITY of EDINBURGH

# IR as an Experimental Science!

- Formulate a research question: the hypothesis
- Design an experiment to answer the question
- Perform the experiment
  - Compare with a baseline "control"
- Does the experiment answer the question?
  - Are the results significant? Or is it just luck?
- Report the results!
- Iterate …
- e.g. stemming improves results? (university → univers)

# Lab 3 output

**Is that a good performance?**

| | | |
|---|---|---|
| 1, 65, 4.8040 | 2, 3549, 7.0396 | 3, 3354, 4.6113 |
| 1, 3533, 4.7264 | 2, 305, 6.8394 | 3, 3345, 4.5087 |
| 1, 3562, 3.5454 | 2, 288, 6.6742 | 3, 268, 3.6606 |
| 1, 3608, 3.4910 | 2, 223, 6.1252 | 3, 328, 3.4825 |
| 1, 141, 3.3262 | 2, 219, 4.8626 | 3, 21, 3.3984 |
| 1, 361, 3.3262 | 2, 3762, 4.8626 | 3, 304, 3.3722 |
| 1, 92, 3.2311 | 2, 3663, 4.5415 | 3, 313, 3.3436 |
| 1, 3829, 3.1818 | 2, 3766, 3.9924 | 3, 3790, 3.1796 |
| 1, 3420, 3.1273 | 2, 188, 3.8844 | 3, 55, 3.0462 |
| 1, 3734, 3.0561 | 2, 3360, 3.0988 | 3, 217, 2.8492 |
| 1, 3387, 2.9626 | 2, 3408, 3.0315 | 3, 361, 2.8348 |
| 1, 3599, 2.9626 | 2, 3390, 2.8498 | 3, 3789, 2.7158 |

# Configure your system

- **About the system**:
  - Stopping? Tokenise? Stemming? n-gram char?
  - Use synonyms improve retrieval performance?
- Corresponding experiment?
  - Run your search for a set of queries with each setup and find which one will achieve the best performance
- **About the user**:
  - Is letting users weight search terms a good idea?
- Corresponding experiment?
  - Build two different interfaces, one with term weighting functionality, and one without; run a user study

# Types of Evaluation Strategies

- **System-centered studies**:
  - Given documents, queries, and relevance judgments
  - Try several variations of the system
  - Measure which system returns the "best" hit list
  - Laboratory experiment

- **User-centered studies**
  - Given several users, and at least two retrieval systems
  - Have each user try the same task on both systems
  - Measure which system works the "best"

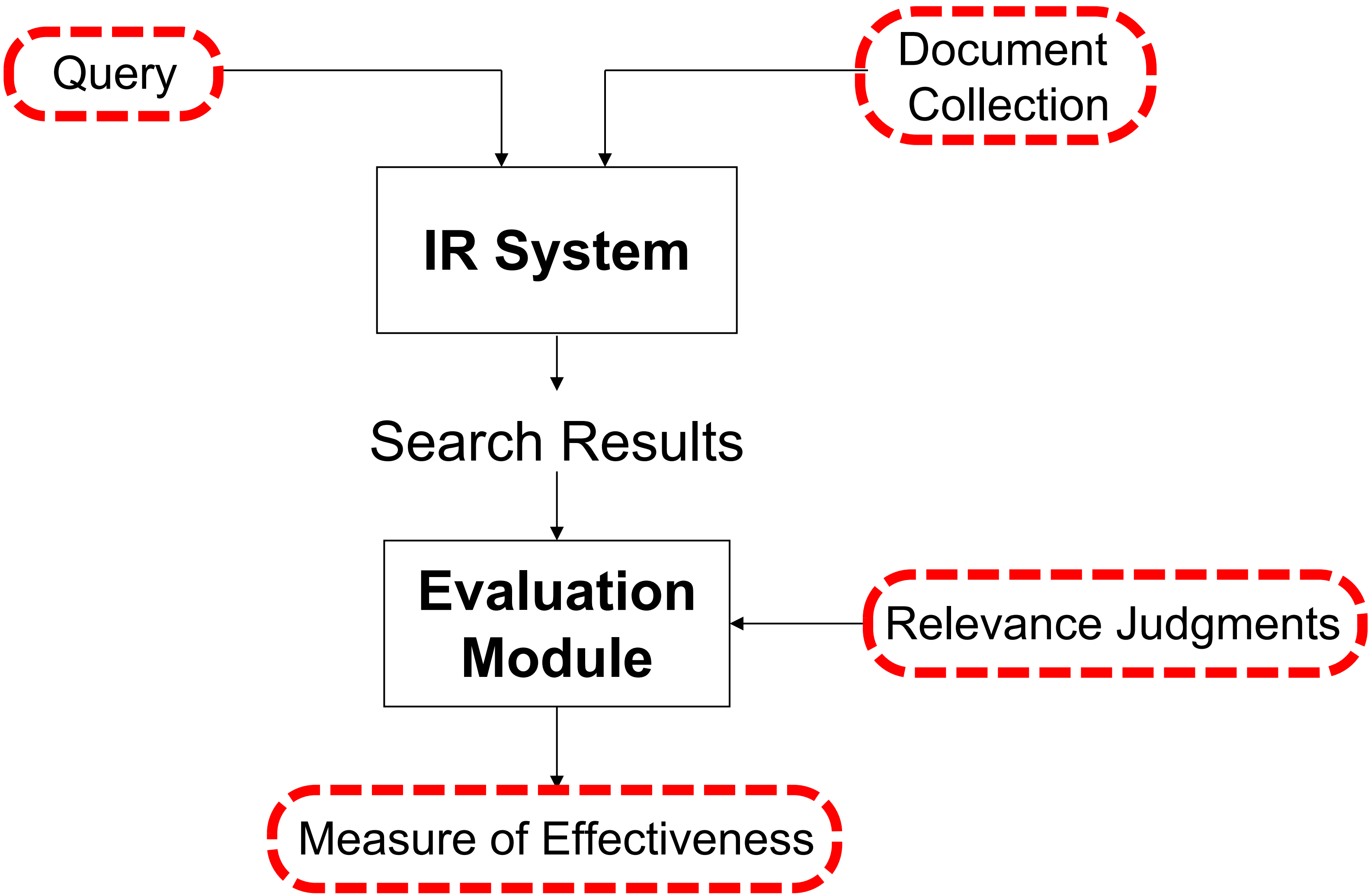
# Importance of Evaluation

- The ability to measure differences underlies experimental science
  - How well do our systems work?
  - Is A better than B?
  - Is it really?
  - Under what conditions?
- Evaluation drives what to research
  - Identify techniques that work and don’t work

# The 3-dimensions of Evaluation

- **Effectiveness**
  - How “good” are the documents that are returned?
  - System only,  human + system
- **Efficiency**
  - Retrieval time,  indexing time,  index size
- **Usability**
  - Learnability,  flexibility
  - Novice vs. expert users

# Cranfield Paradigm (Lab setting)



# Reusable IR Test Collection

- **Collection of Documents**
  - Should be "representative" to a given IR task
  - Things to consider: size, sources, genre, topics, …
- **Sample of information need**
  - Should be "randomized" and "representative"
  - Usually formalized **topic** statements (query + description)
- **Known relevance judgments**
  - Assessed by humans, for each topic-document pair
  - Binary/Graded
- **Evaluation measure**

# Good Effectiveness Measures

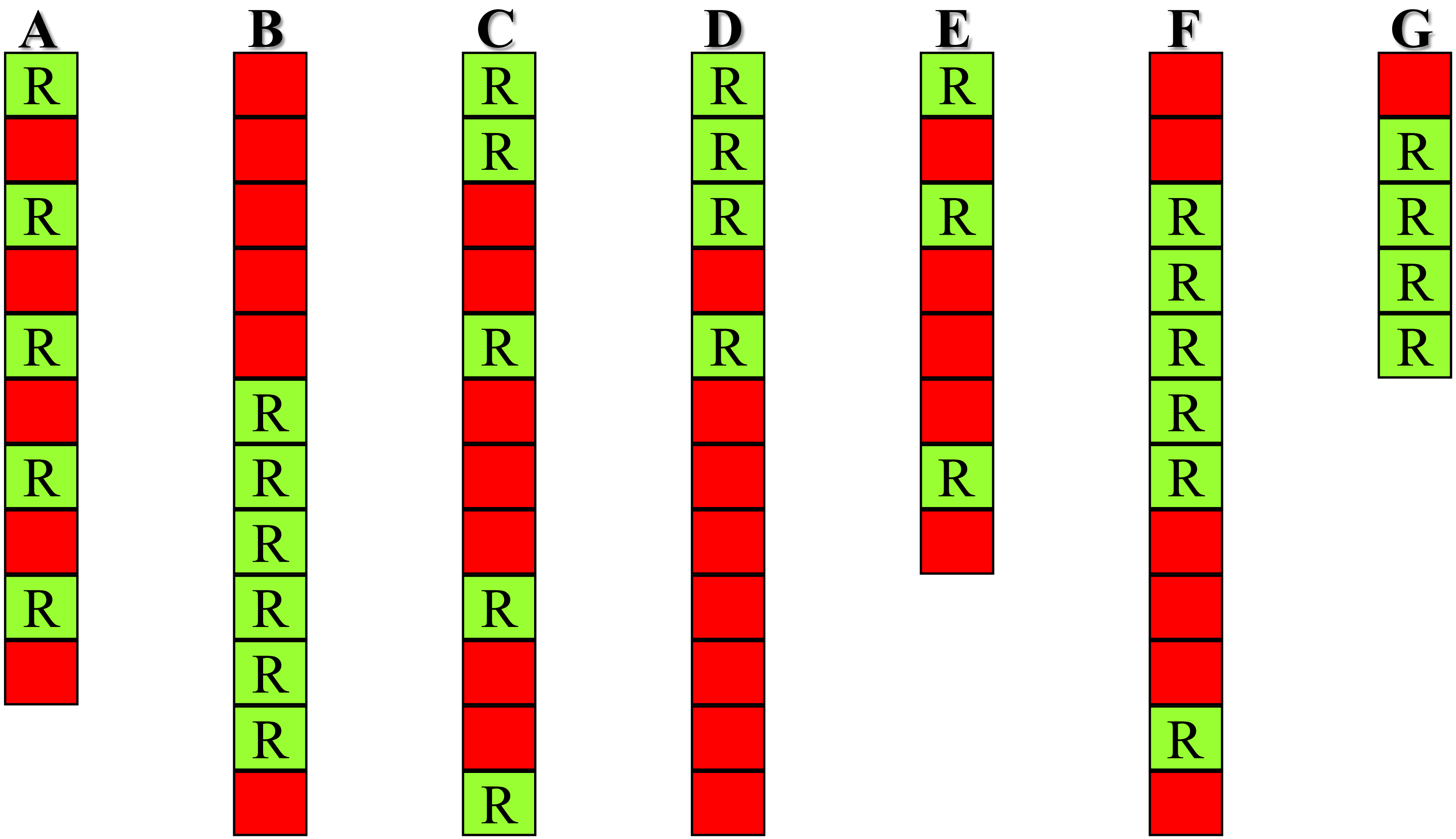
- Should capture some aspect of what the user wants
  - IR → Do the results satisfy user's information need?
- Should be easily replicated by other researchers
- Should be easily comparable
  - Optimally, expressed as a single number
    - Curves and multiple numbers are still accepted, but single numbers are much easier for comparison
- Should have predictive value for other situations
  - What happens with different queries on a different document collection?

# Set Based Measures

- Assuming IR system returns sets of retrieved results without ranking
- Suitable with Boolean Search
- No certain number of results per query

# Which looks the best IR system?

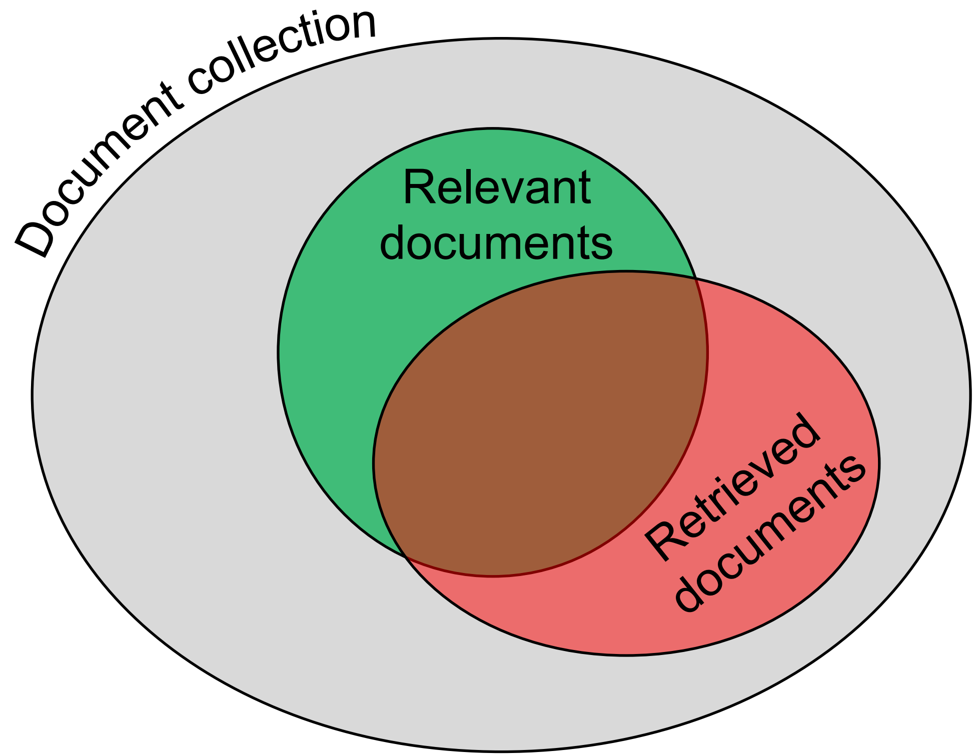
- For query **Q**, collection has **8 relevant documents**:

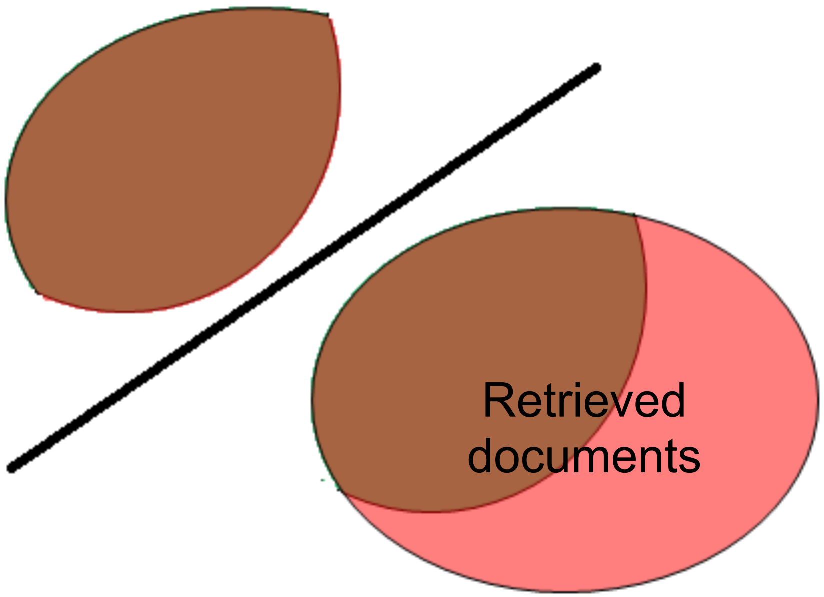


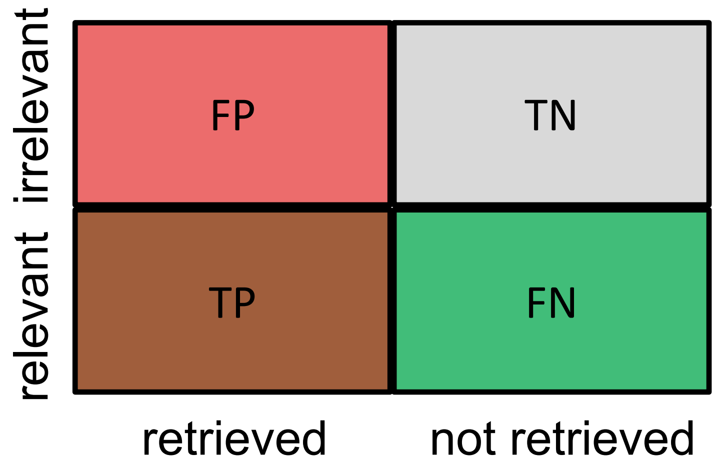
# Precision and Recall

- **Precision**:
  What fraction of these retrieved docs are relevant?

$$P = \frac{rel \cap ret}{retrieved} = \frac{TP}{TP + FP}$$



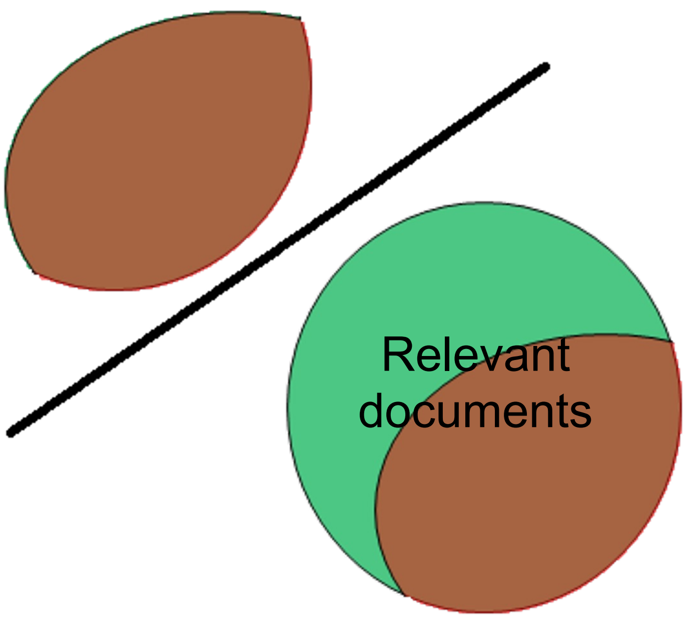



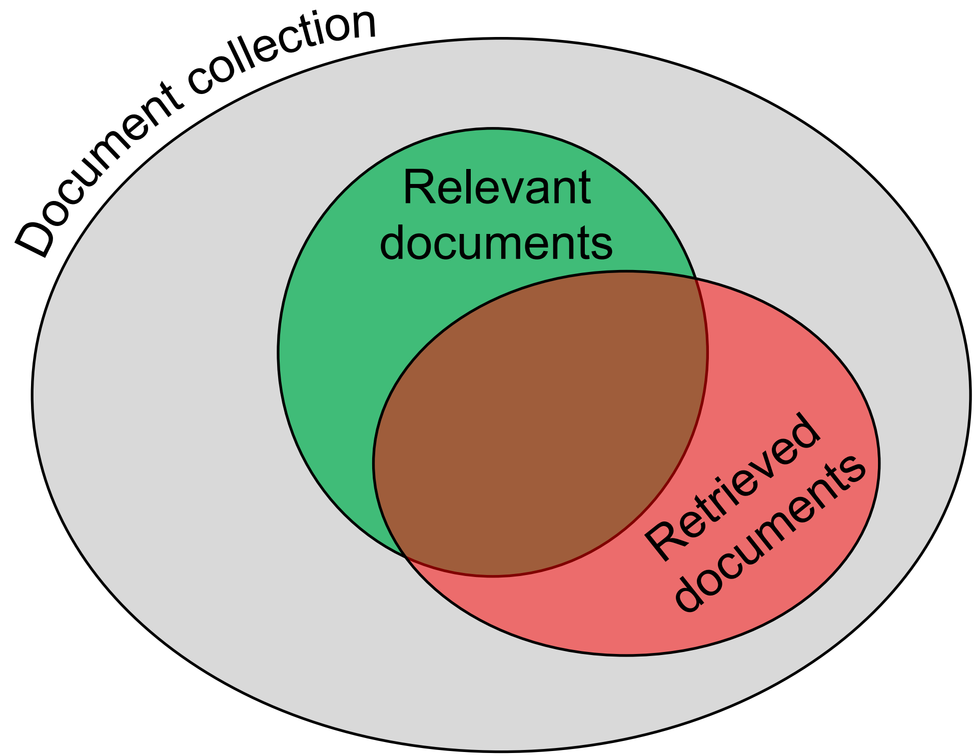



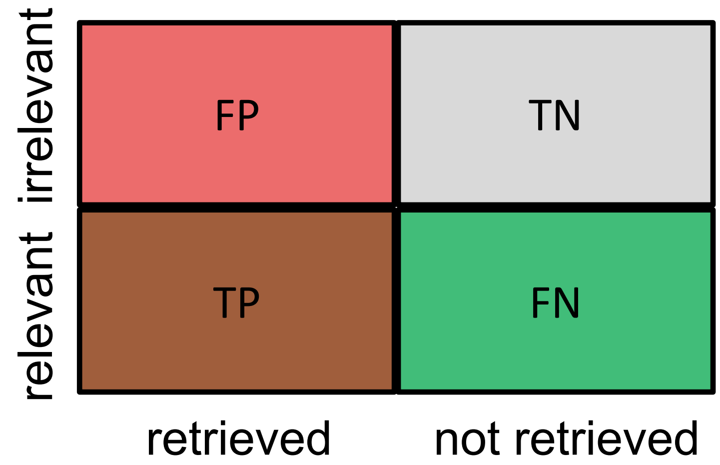
# Precision and Recall

- **Recall**:
  What fraction of the relevant docs were retrieved?

$$R = \frac{rel \cap ret}{relevant} = \frac{TP}{TP + FN}$$

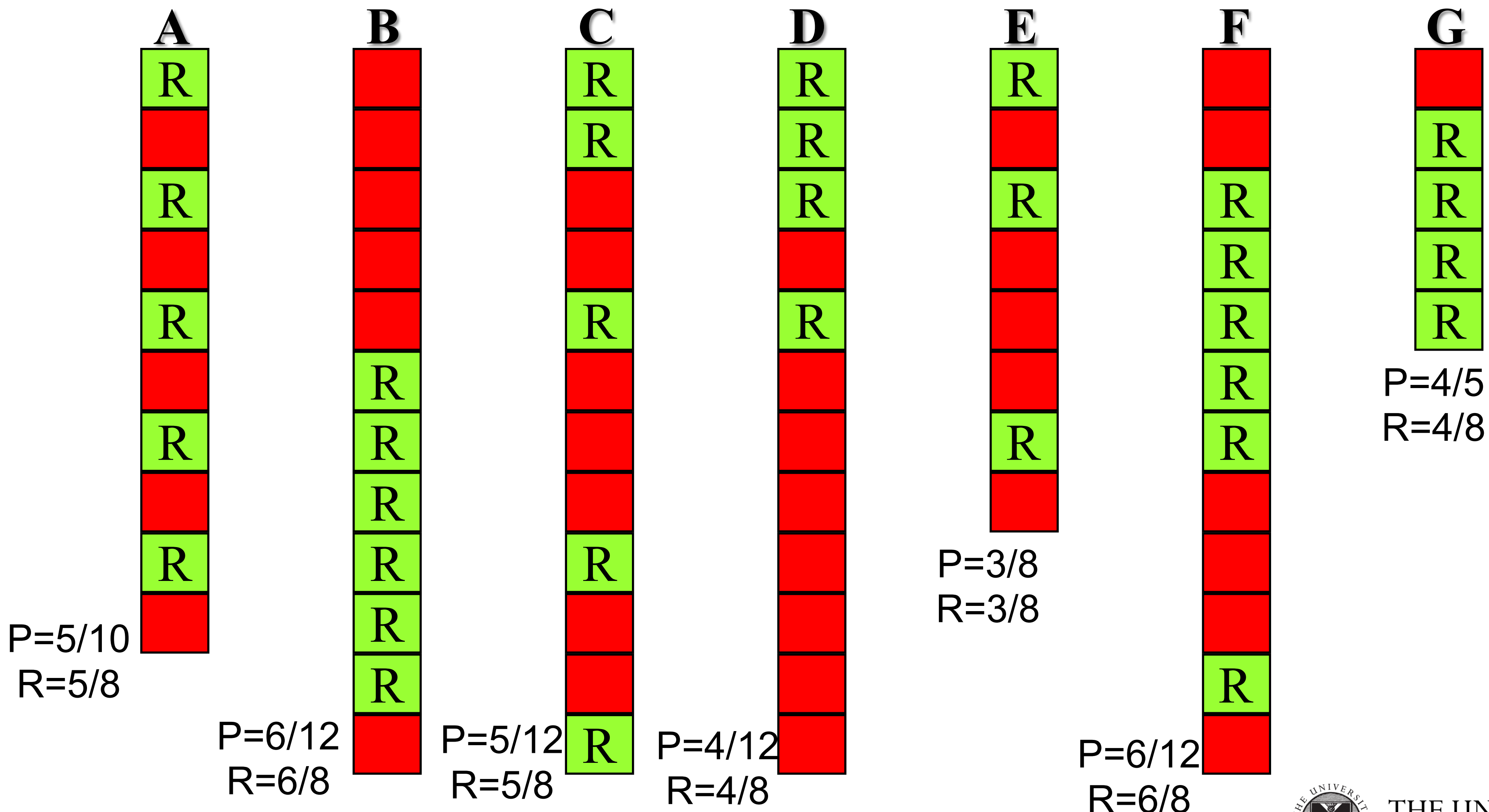


# Which looks the best IR system?

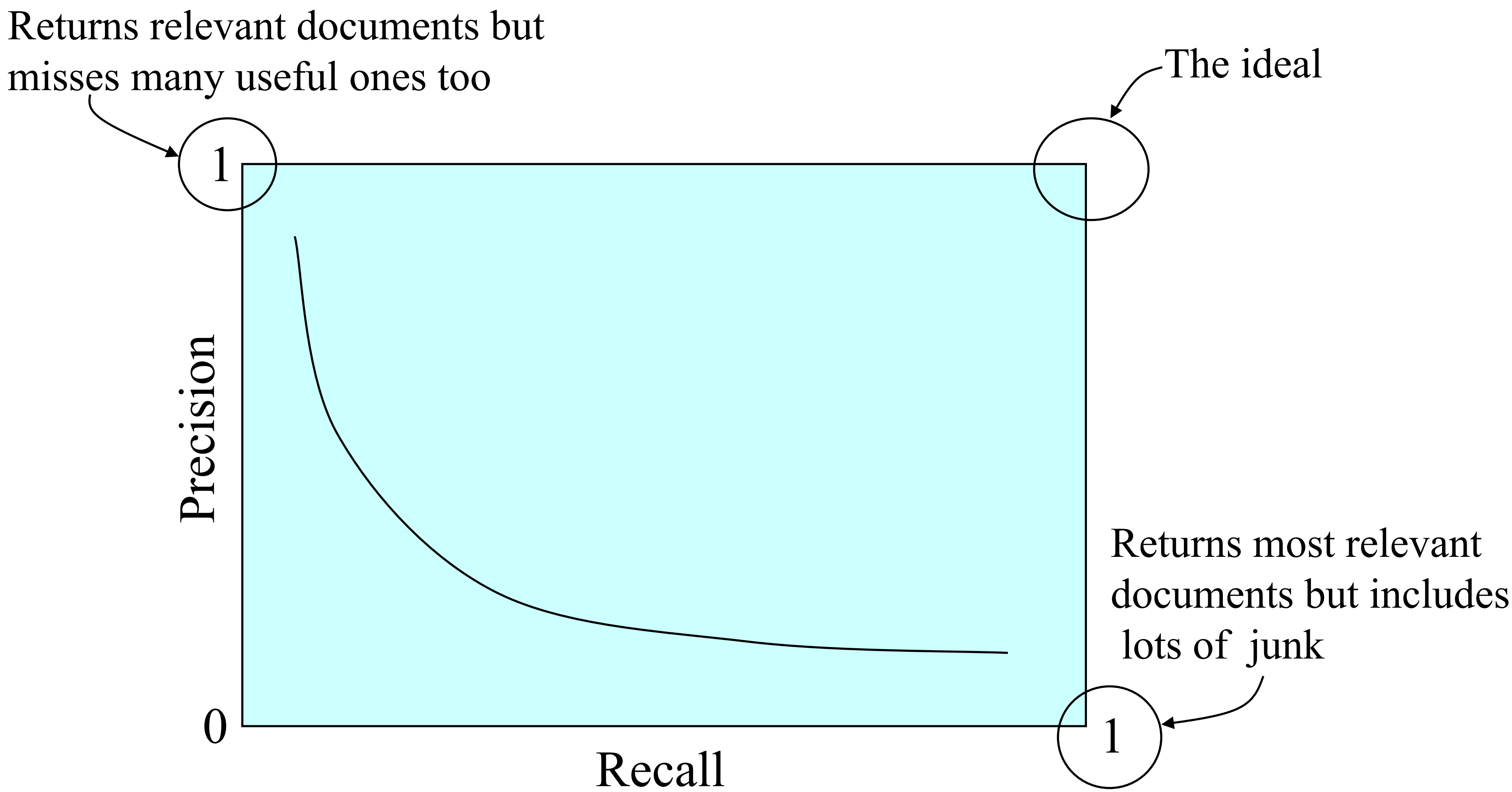
- For query **Q**, collection has **8 relevant documents**:



# Trade-off between P & R

- Precision: The ability to retrieve top-ranked docs that are mostly relevant.
- Recall: The ability of the search to find all of the relevant items in the corpus.
- Retrieve more docs:
  - Higher chance to find all relevant docs → R ↑↑
  - Higher chance to find more irrelevant docs → P ↓↓

# Trade-off between P & R



# What about Accuracy?

- **Accuracy**:
  What fraction of docs was classified correctly?

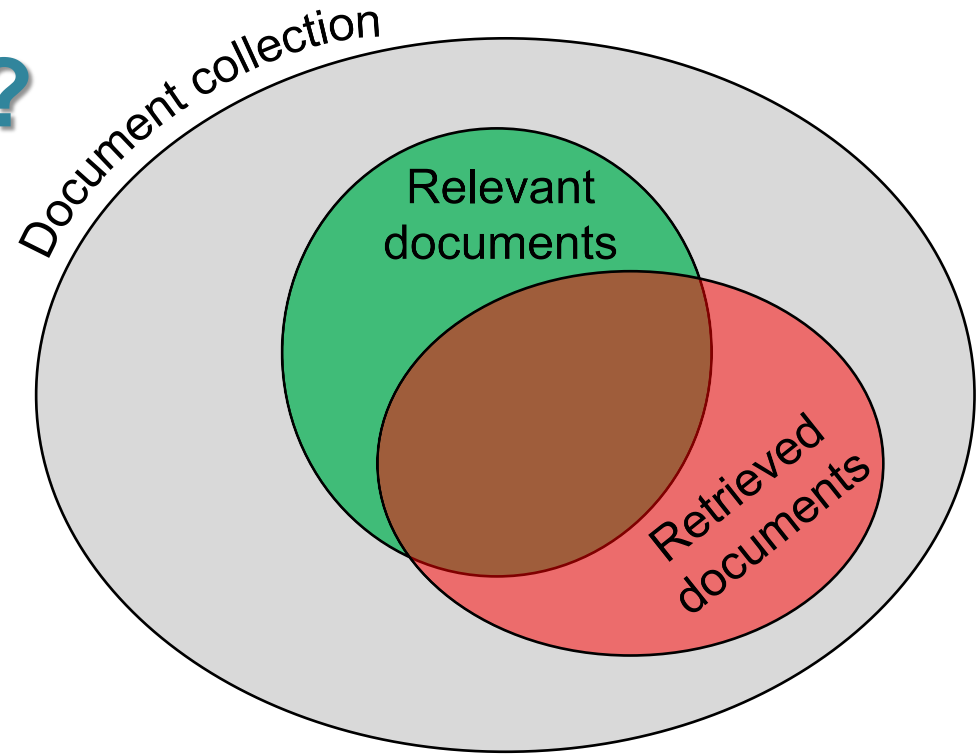
$$A = \frac{TP + TN}{TP + FP + TN + FN}$$

*irrelevant >>>>> relevant*

*(needle in a haystack)*

e.g.: $N_{docs}$ = 1M docs, *rel*=10, *ret*=10

$$TP = 5, \qquad FP = 5,$$
$$FN\ 5, \qquad TN = 1M - 15$$
$$\rightarrow A = 99.999\%$$




| | retrieved | not retrieved |
| --- | --- | --- |
| irrelevant | FP | TN |
| relevant | TP | FN |

# One Measure? F-measure

$$F1 = \frac{2 \cdot P \cdot R}{P + R}$$

$$F_\beta = \frac{(\beta^2 + 1)P \cdot R}{\beta^2 P + R}$$

- Harmonic mean of recall and precision
  - Emphasizes the importance of small values, whereas the arithmetic mean is affected more by outliers that are unusually large
- Beta ($\beta$) controls relative importance of P and R
  - $\beta$ = 1, precision and recall equally important → $F1$
  - $\beta$ = 5, recall five times more important than precision

# F-measure?

- For query **Q**, collection has **8 relevant documents**:

| System | Precision | Recall | F1 |
|---|---|---|---|
| **A** | 0.500 | 0.625 | 0.556 |
| **B** | 0.500 | 0.750 | 0.600 |
| **C** | 0.417 | 0.625 | 0.500 |
| **D** | 0.333 | 0.500 | 0.400 |
| **E** | 0.375 | 0.375 | 0.375 |
| **F** | 0.500 | 0.750 | 0.600 |
| **G** | 0.800 | 0.500 | 0.615 |

P=
R=5/8

R

P=6/12
R=6/8

P=5/12 R
R=5/8

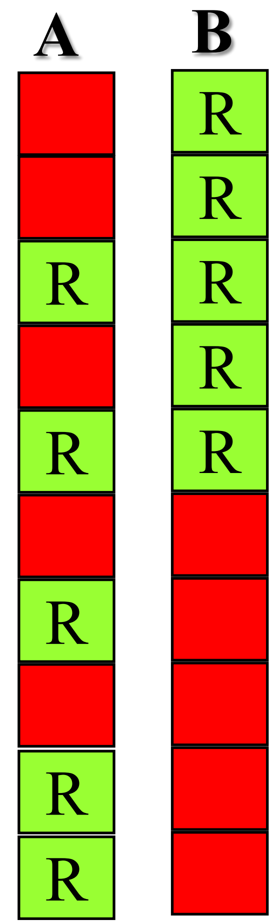
P=4/12
R=4/8

R

P=6/12
R=6/8

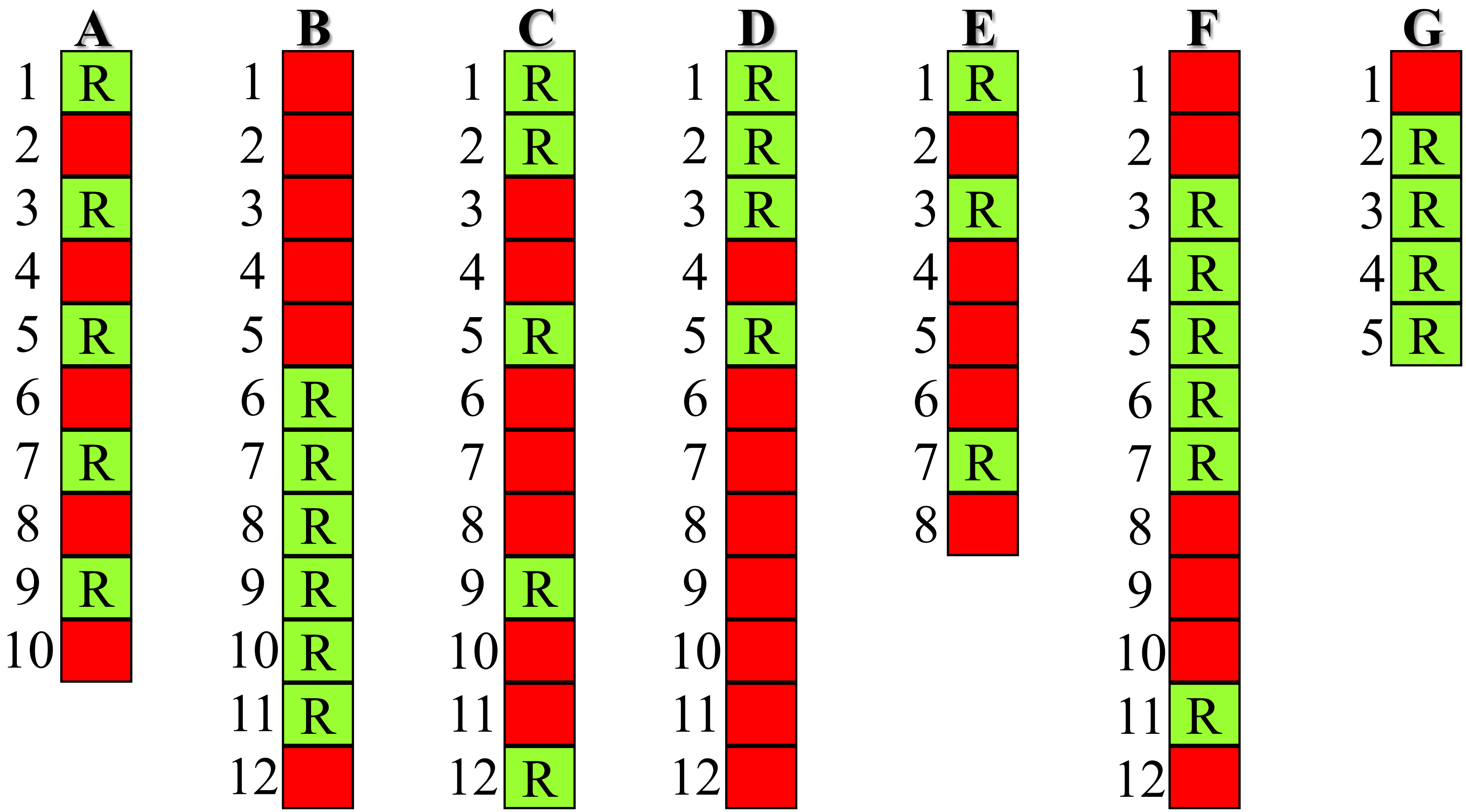
# Rank-based IR measures

- Consider systems A & B
  - Both retrieved 10 docs,  only 5 are relevant
  - P,  R & F are the same for both systems
    - Should their performances considered equal?
- Ranked IR requires taking "<u>ranks</u>" into consideration!
- How to do that?



# Which is the best ranked list?

- For query **Q**, collection has **8 relevant documents**:






THE UNIVERSITY *of* EDINBURGH

# Precision @ K

- *k* (a fixed number of documents)
- Have a cut-off on the ranked list at rank *k*,  then calculate precision!
- Perhaps appropriate for most of web search: most people only check the top *k* results
- But: averages badly,  Why?

# P@5

- For query **Q**, collection has **8 relevant documents**:

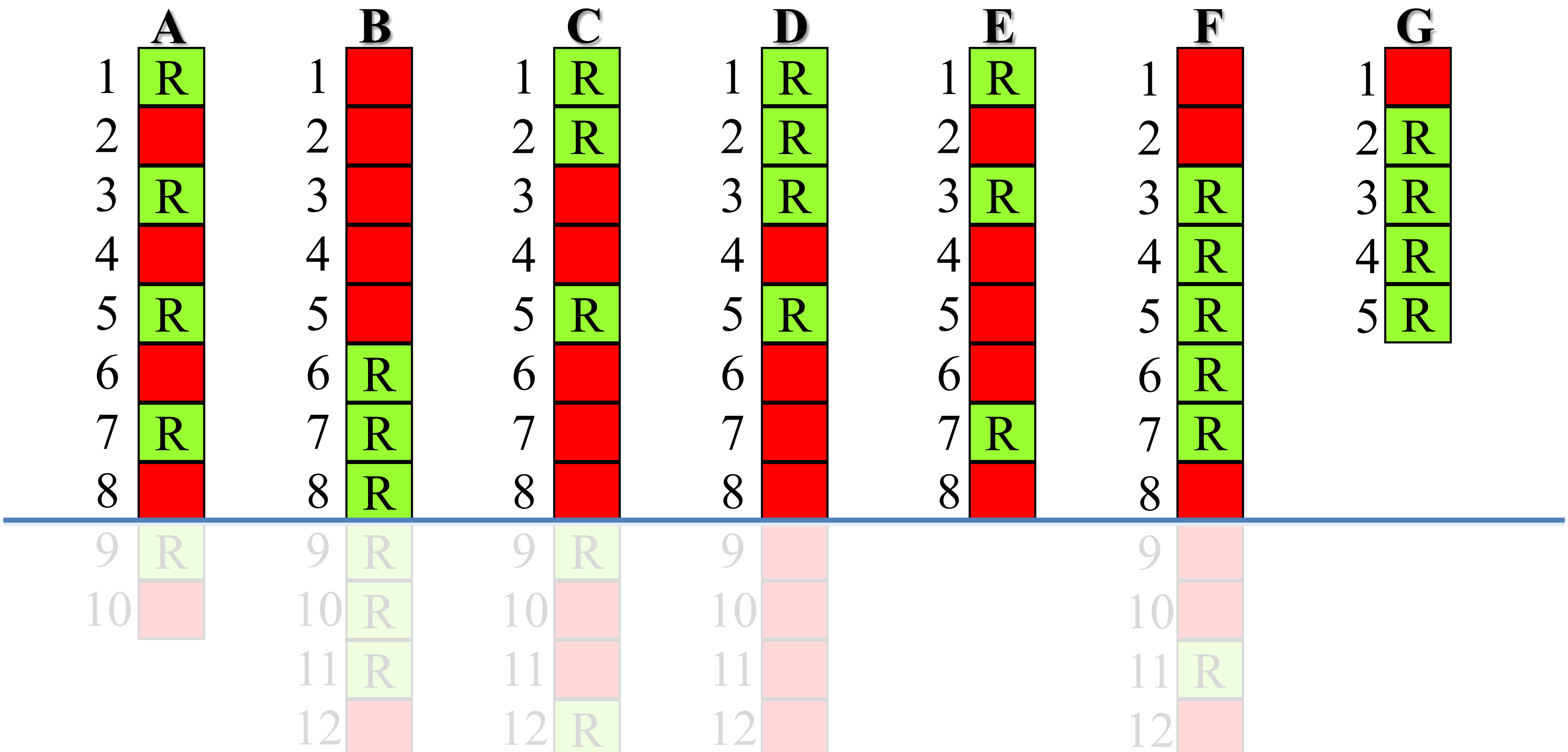
| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | R | | R | R | R | | |
| 2 | | | R | R | | | R |
| 3 | R | | | R | R | R | R |
| 4 | | | | | | R | R |
| 5 | R | | R | R | | R | R |
| 6 | | R | | | | R | |
| 7 | R | R | | | R | R | |
| 8 | | R | | | | | |
| 9 | R | R | R | | | | |
| 10 | | R | | | | | |
| 11 | | R | | | | R | |
| 12 | | | R | | | | |

# R-Precision

- For a query with known *r* relevant documents
  → R-precision is the precision at rank *r* (P@*r*)
- *r* is different from one query to another
- Concept:
  It examines the ideal case: getting all relevant documents in the top ranks
- Is it realistic?

# R-Precision

- For query **Q**, collection has **8 relevant documents**:



# User Satisfaction??

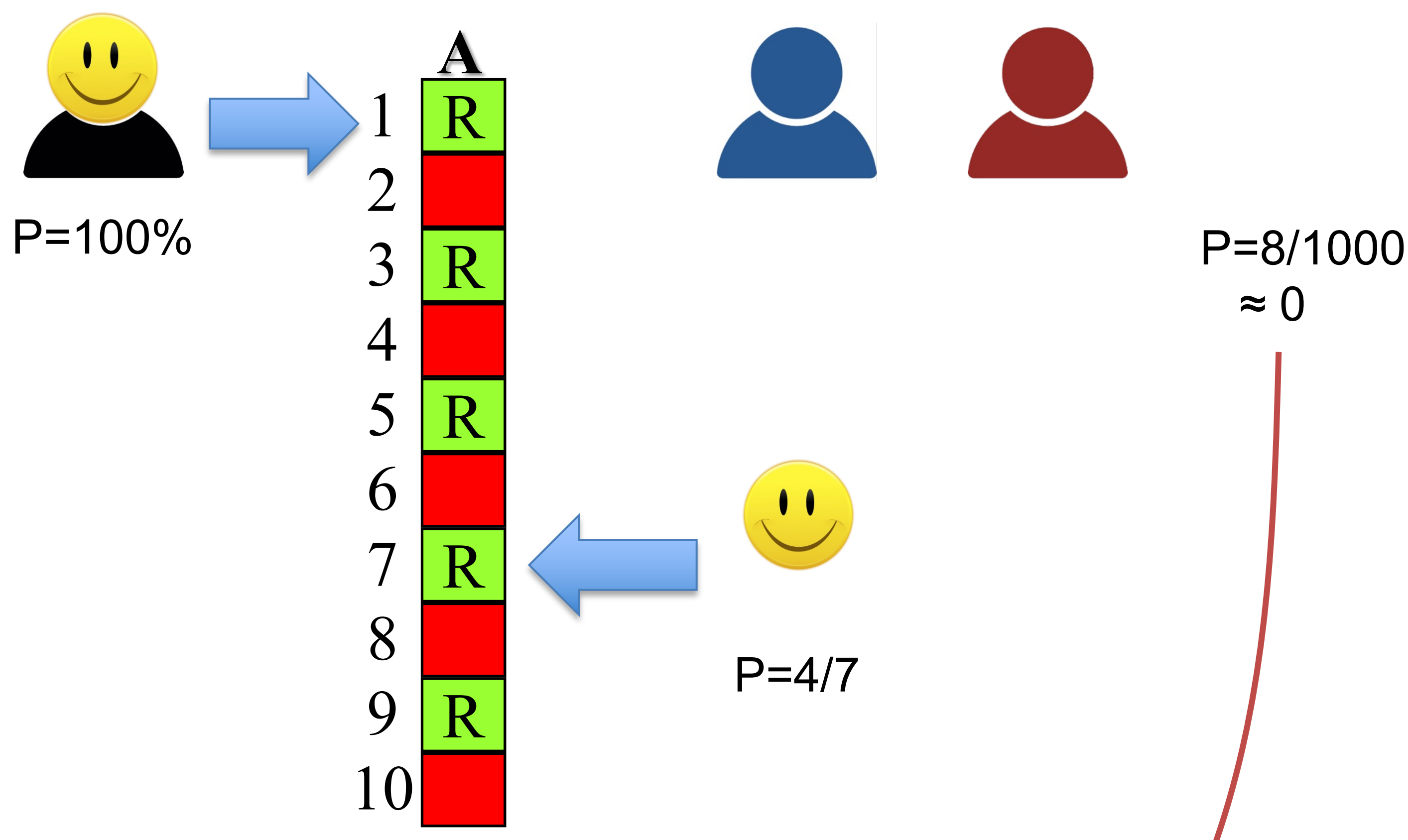
- It is assumed that users needs to find relevant docs at the highest possible ranks
  → Precision is a good measure
- But,  user would cut-off (stop inspecting results) at some point,  say rank x
  →P@x
- What is the optimal x?
  When you think a user can stop?

# When a user can stop?

- IR objective: "satisfy user information need"
- Assumption: a user will stop once his/her information need is satisfied
- How? user will keep looking for relevant docs in the ranked list, read them, then stop once he/she feels satisfied
- P@x →x can be any rank where a relevant document appeared (*assume uniform distribution*)

# When to stop?

- For query **Q**, collection has **8 relevant documents**:






THE UNIVERSITY *of* EDINBURGH

# When a user can stop?

- IR objective: "satisfy user information need"
- Assumption: a user will stop once his/her information need is satisfied
- How? user will keep looking for relevant docs in the ranked list, read them, then stop once he/she feels satisfied
- P@x → x can be any rank where a relevant document appeared (*assume uniform distribution*)
- What about calculating the averages over all x's?
  - every time you find relevant doc, calculate P@x, then take the average at the end

# Average Precision (AP)

**$Q_1$** (has 4 rel. docs)

| Rank | Doc | Precision |
|---|---|---|
| 1 | R | 1/1=1.00 |
| 2 | R | 2/2=1.00 |
| 3 | | |
| 4 | | |
| 5 | R | 3/5=0.60 |
| 6 | | |
| 7 | | |
| 8 | | |
| 9 | R | 4/9=0.44 |
| 10 | | |

AP = 3.04 / 4

= **0.76**

**$Q_2$** (has 3 rel. docs)

| Rank | Doc | Precision |
|---|---|---|
| 1 | | |
| 2 | | |
| 3 | R | 1/3=0.33 |
| 4 | | |
| 5 | | |
| 6 | | |
| 7 | R | 2/7=0.29 |
| 8 | | |
| ⋮ | | $\frac{3}{\infty}=0$ |

AP = 0.62 / 3

= **0.207**

**$Q_3$** (has 7 rel. docs)

| Rank | Doc | Precision |
|---|---|---|
| 1 | | |
| 2 | R | 1/2=0.50 |
| 3 | | |
| 4 | | |
| 5 | R | 2/5=0.40 |
| 6 | | |
| 7 | | |
| 8 | R | 3/8=0.375 |
| 9 | | |
| ⋮ | | |

AP = 1.275 / 7

= **0.182**

# Mean Average Precision (MAP)

| Rank | $Q_1$ (has 4 rel. docs) | | $Q_2$ (has 3 rel. docs) | | $Q_3$ (has 7 rel. docs) | |
|---|---|---|---|---|---|---|
| 1 | R | 1/1=1.00 | | | | |
| 2 | R | 2/2=1.00 | | | R | 1/2=0.50 |
| 3 | | | R | 1/3=0.33 | | |
| 4 | | | | | | |
| 5 | R | 3/5=0.60 | | | R | 2/5=0.40 |
| 6 | | | | | | |
| 7 | | | R | 2/7=0.29 | | |
| 8 | | | | | R | 3/8=0.375 |
| 9 | R | 4/9=0.44 | | | | |
| 10 | | | | | | |
| ⋮ | | | | $\frac{3}{\infty}=0$ | | |
| | AP = 0.76 | | AP = 0.207 | | AP = 0.182 | |

**MAP** = (0.76+0.207+0.182)/3 = **0.383**

# AP & MAP

$$AP = \frac{1}{r}\sum_{k=1}^{n} P(k) \times rel(k)$$

where, $r$: number of relevant docs for a given query
$n$: number of documents retrieved
$P(k)$ precision @ $k$
$rel(k)$: 1 if retrieved doc @ $k$ is relevant, 0 otherwise.

$$MAP = \frac{1}{Q}\sum_{q=1}^{Q} AP(q)$$

where, $Q$: number of queries in the test collection

# AP/MAP

$$AP = \frac{1}{r}\sum_{k=1}^{n} P(k) \times rel(k)$$

- A mix between precision and recall
- Highly focus on finding relevant document as early as possible
- When $r$=1 → MAP = MRR (mean reciprocal rank $\frac{1}{k}$)
- MAP is the most commonly used evaluation metric for most IR search tasks
- Uses binary relevance: rel = 0/1

# MAP

- For query **Q**, collection has **8 relevant documents**:

A 3 | B 6 | C 2 | D 1 | E 7 | F 4 | G 5

| Rank | A | G |
|---|---|---|
| 1 | R | |
| 2 | | R |
| 3 | R | R |
| 4 | | R |
| 5 | R | R |
| 6 | | |
| 7 | R | |
| 8 | | |
| 9 | R | |
| 10 | | |

| System | Precision | Recall | F1 |
|---|---|---|---|
| A | 0.500 | 0.625 | 0.556 |
| B | 0.500 | 0.750 | 0.600 |
| C | 0.417 | 0.625 | 0.500 |
| D | 0.333 | 0.500 | 0.400 |
| E | 0.375 | 0.375 | 0.375 |
| F | 0.500 | 0.750 | 0.600 |
| G | 0.800 | 0.500 | 0.615 |

12 (B) | 12 R (C) | 12 (D) | 12 (F)

# Binary vs. Graded Relevance

- Some docs are more relevant to a query than other relevant ones!
  - We need non-binary relevance
- Binary Relevance:
  - Relevant 1
  - Irrelevant 0
- Graded Relevance:
  - Perfect 4
  - Excellent 3
  - Good 2
  - Fair 1
  - Bad 0

# Binary vs. Graded Relevance

- Two assumptions:
  - Highly relevant documents are more useful than marginally relevant
  - 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 (DCG)
  - Uses graded relevance as a measure of the usefulness
  - The most popular for evaluating web search

# Discounted Cumulative Gain (DCG)

- Gain is accumulated starting at the top of the ranking and may be reduced (*discounted*) at lower ranks
- Users care more about high-ranked documents, so we discount results by $1/\log_2(rank)$
  - the discount at rank 4 is 1/2, and at rank 8 is 1/3
- $DCG_k$ is the total gain accumulated at a particular rank *k* (sum of DG up to rank *k*):

0, 1, 2, 3, … (graded)

$$DCG_k = rel_1 + \sum_{i=2}^{k} \frac{rel_i}{\log_2(i)}$$

# DCG

| k | G |
|---|---|
| 1 | 3 |
| 2 | 2 |
| 3 | 3 |
| 4 | 0 |
| 5 | 0 |
| 6 | 1 |
| 7 | 2 |
| 8 | 2 |
| 9 | 3 |
| 10 | 0 |

# Normalized DCG (nDCG)

- DCG numbers are averaged across a set of queries at specific rank values (DCG@*k*)
  - e.g., DCG at rank 5 is 6.89 and at rank 10 is 9.61
  - Can be any positive real number!
- DCG values are often normalized by comparing the DCG at each rank with the DCG value for the perfect ranking
  - makes averaging easier for queries with different numbers of relevant documents
- nDCG@*k* = DCG@*k* / iDCG@*k* (divide actual by ideal)
- nDCG ≤ 1 at any rank position
- To compare DCGs, normalize values so that a ideal ranking would have a normalized DCG of 1.0

# nDCG

÷

| k | G | DG | DCG@k | | iG |
|---|---|---|---|---|---|
| 1 | 3 | 3 | 3 | | 3 |
| 2 | 2 | 2 | 5 | | 3 |
| 3 | 3 | 1.89 | 6.89 | | 3 |
| 4 | 0 | 0 | 6.89 | | 2 |
| 5 | 0 | 0 | 6.89 | | 2 |
| 6 | 1 | 0.39 | 7.28 | | 2 |
| 7 | 2 | 0.71 | 7.99 | | 1 |
| 8 | 2 | 0.67 | 8.66 | | 0 |
| 9 | 3 | 0.95 | 9.61 | | 0 |
| 10 | 0 | 0 | 9.61 | | 0 |

# Summary:

- IR test collection:
  - Document collection
  - Query set
  - Relevant judgements
  - IR measures
- IR measures:
  - R, P, F → not commonly used
  - P@k, R-precision → used sometimes
  - MAP → the most used IR measure
  - nDCG → the most used measure for web search

# Resources

- Text book 1: Intro to IR,  Chapter 8
- Text book 2: IR in Practice,  Chapter 8