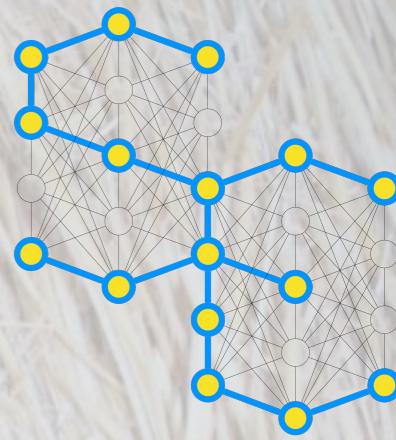


**800**  
ANNI  
1222-2022



**UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA**



# Evaluation Measures



## Search Engines

Master Degree in Computer Engineering

Master Degree in Data Science

Academic Year 2023/2024



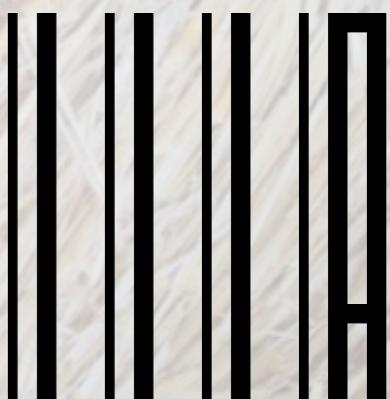
DIPARTIMENTO  
DI INGEGNERIA  
DELL'INFORMAZIONE

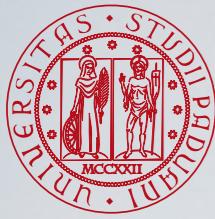
**Nicola Ferro**

Intelligent Interactive Information Access (IIIA) Hub

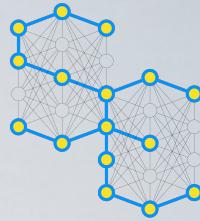
Department of Information Engineering

University of Padua

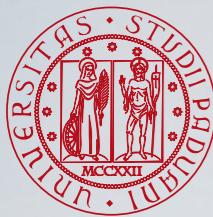




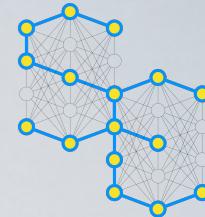
# Outline



- Set-based Evaluation Measures
- Rank-based Evaluation Measures



# Evaluation Measures

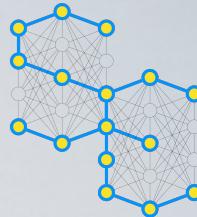


“Measure what is measurable  
and make measurable what is  
not”

Galileo Galilei (1564-1642)

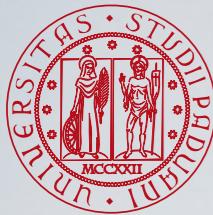


# A Taxonomy of Evaluation Measures



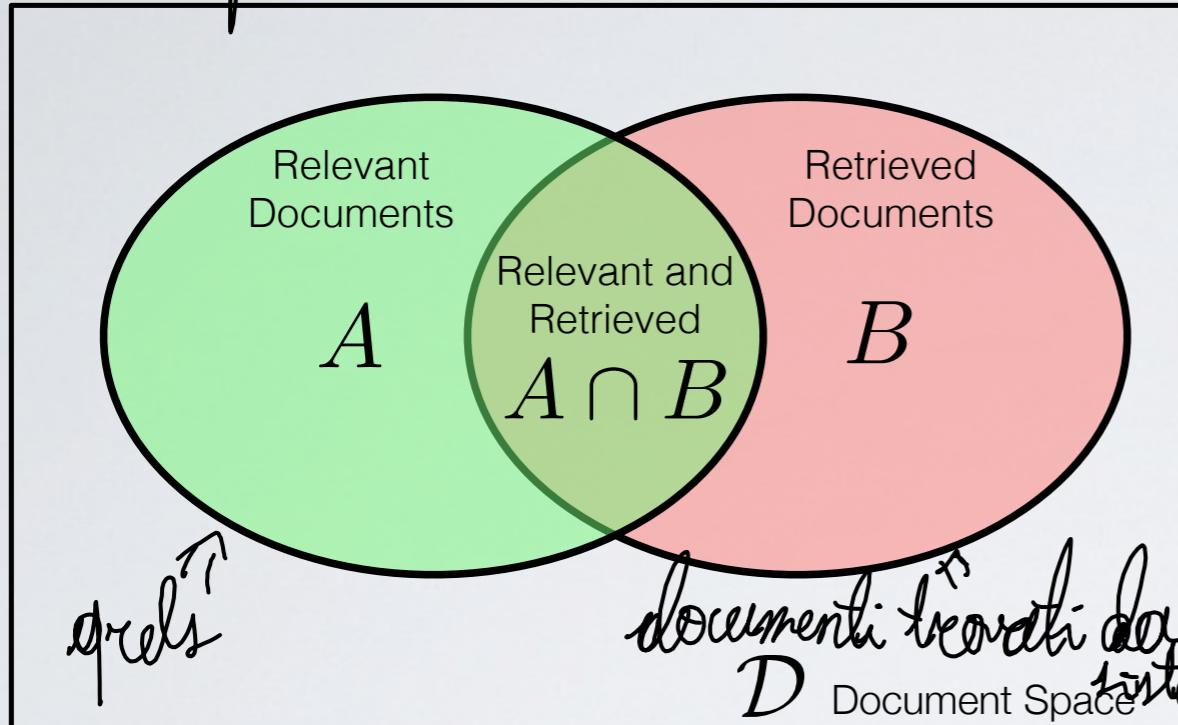
|                        | Set-Based Retrieval<br><i>→ niente ranking <math>\Rightarrow</math> usato in primi sistemi</i> | Rank-Based Retrieval<br><i>primi <math>k</math> elementi, calcolo misure</i>   |
|------------------------|--|--|
| Binary Relevance       | Precision (P)<br>Recall (R)<br>F-measure (F)   | Precision at Document Cut-off (P@k)<br>Recall at Document Cut-off (R@k)<br>R-Precision (Rprec)<br>Average Precision (AP)<br>Rank-Biased Precision (RBP)<br>... |
| Multi-graded Relevance | Not widely agreed generalizations of Precision and Recall                                      | Discounted Cumulated Gain (DCG)<br>...   |

# Set-based Evaluation Measures



# Set-based Measures: Precision, Recall and F-measure

qui, parliamo di un solo topic



$$P = \frac{|A \cap B|}{|B|}$$

$$R = \frac{|A \cap B|}{|A|}$$

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

- **Precision** is the proportion of retrieved documents that are actually relevant
- **Recall** is the proportion of relevant documents actually retrieved *→ bontà di uno da campione di doc valutati*
- Together, Precision and Recall measure **retrieval effectiveness**, meant as the ability of a system to retrieve relevant documents while at the same time holding back non-relevant ones
  - maximizing Precision and Recall corresponds to optimal retrieval in the sense of the **Probability Ranking Principle**, i.e. ordering documents by their decreasing probability of being relevant, and creates a tight link between retrieval models and evaluation
- **F-measure** is the harmonic mean of Precision and Recall, summarising them into a single score

van Rijsbergen, C. J. (1974). Foundations of Evaluation. *Journal of Documentation*, 30(4):365–373.

van Rijsbergen, C. J. (1981). Retrieval effectiveness. In Spärck Jones, K., editor, *Information Retrieval Experiment*, pages 32–43. Butterworths, London, United Kingdom.

# Set-based Measures: Example

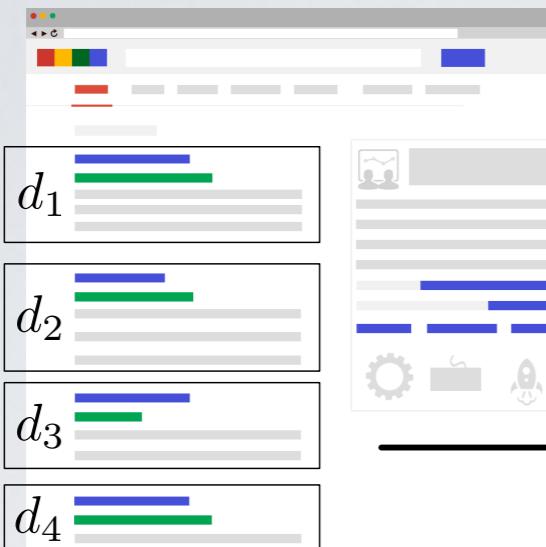
Topic

Run

Assessed Run

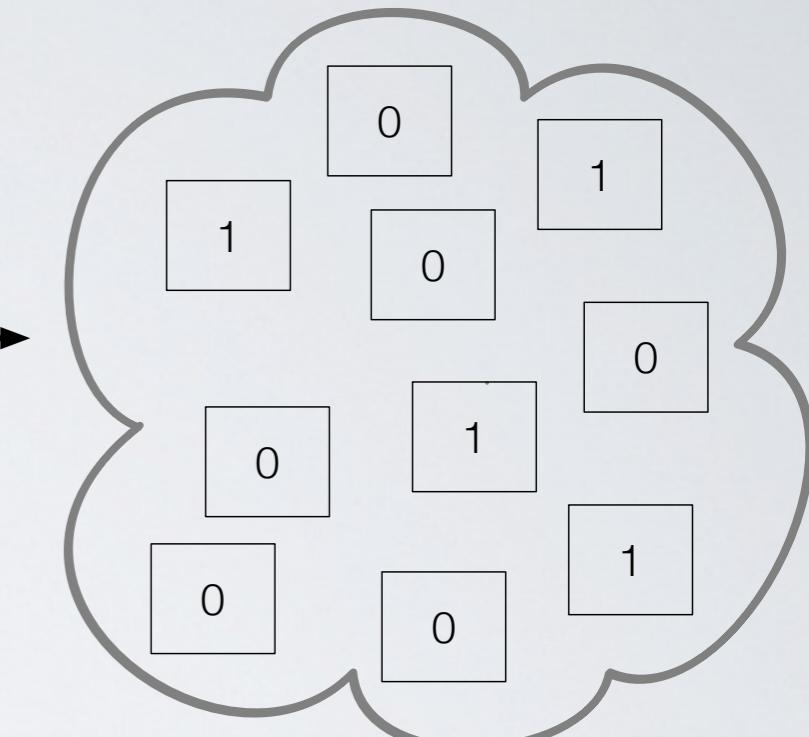
Binary Weighted  
Assessed Run

Set-based  
View



|    |                    |
|----|--------------------|
| 1  | Highly Relevant    |
| 2  | Not Relevant       |
| 3  | Partially Relevant |
| 4  | Fairly Relevant    |
| 5  | Not Relevant       |
| 6  | Not Relevant       |
| 7  | Not Relevant       |
| 8  | Fairly Relevant    |
| 9  | Not Relevant       |
| 10 | Not Relevant       |

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



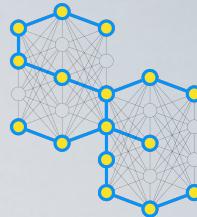
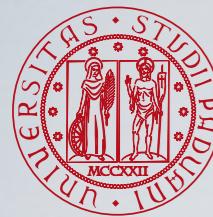
$$P = \frac{4}{10} = 0.40$$

$$R = \frac{4}{8} = 0.50$$

$$F = 2 \cdot \frac{\frac{4}{10} \cdot \frac{4}{8}}{\frac{4}{10} + \frac{4}{8}} = \frac{4}{9} = 0.44$$

Assume

- $|A| = 8$  relevant documents in total
- Lenient mapping to binary relevance degrees



# What Type of Mean?

perché  $F$ -mean è  
media armonica?



$$s = \frac{d}{t}$$



- Suppose Herbie goes forward a distance  $d$  at speed  $s_1$ , say 60 km/h, and goes backward the same distance  $d$  at speed  $s_2$ , say 20 km/h. What is Herbie's average speed during the travel?

$$\frac{s_1 + s_2}{2} = 40 \text{ km/h}$$

$$2 \frac{s_1 \cdot s_2}{s_1 + s_2} = 30 \text{ km/h}$$

Ferger, W. F. (1931). The Nature and Use of the Harmonic Mean. *Journal of the American Statistical Association*, 26(173):36–40.

# What Type of Mean?



$$s = \frac{d}{t}$$



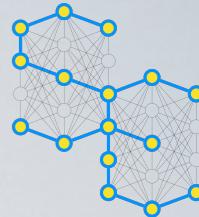
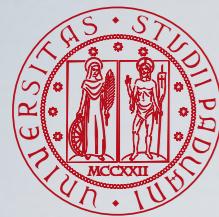
- Suppose Herbie goes forward a distance  $d$  at speed  $s_1$ , say 60 km/h, and goes backward the same distance  $d$  at speed  $s_2$ , say 20 km/h. What is Herbie's average speed during the travel?

$$\frac{s_1 + s_2}{2} = 40 \text{ km/h}$$

$$2 \frac{s_1 \cdot s_2}{s_1 + s_2} = 30 \text{ km/h}$$

$$\bar{s} = \frac{\text{total distance}}{\text{total time}} = \frac{d + d}{\frac{d}{s_1} + \frac{d}{s_2}} = 2 \frac{s_1 \cdot s_2}{s_1 + s_2}$$

Ferger, W. F. (1931). The Nature and Use of the Harmonic Mean. *Journal of the American Statistical Association*, 26(173):36–40.



# What Type of Mean?



Mean of two numbers,  $s_1$  and  $s_2$ , which are ratios and where the numerator  $d$  is fixed

$$s = \frac{d}{t}$$

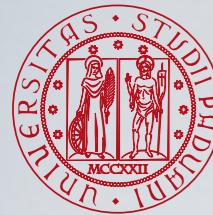


Suppose Herbie goes forward a distance  $d$  at speed  $s_1$ , say 60 km/h, and goes backward the same distance  $d$  at speed  $s_2$ , say 20 km/h. What is Herbie's average speed during the travel?

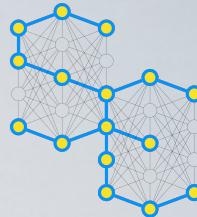
$$2 \frac{s_1 \cdot s_2}{s_1 + s_2} = 30 \text{ km/h}$$

$$\bar{s} = \frac{\text{total distance}}{\text{total time}} = \frac{d + d}{\frac{d}{s_1} + \frac{d}{s_2}} = 2 \frac{s_1 \cdot s_2}{s_1 + s_2}$$

Ferger, W. F. (1931). The Nature and Use of the Harmonic Mean. *Journal of the American Statistical Association*, 26(173):36–40.



# What Type of Mean?



$$s = \frac{d}{t}$$



- Suppose Herbie goes for a time  $t$ , say 10 minutes, at speed  $s_1$ , say 60 km/h, and for the same amount of time  $t$  at speed  $s_2$ , say 20 km/h. What is Herbie's average speed during the travel?

$$\frac{s_1 + s_2}{2} = 40 \text{ km/h}$$

$$2 \frac{s_1 \cdot s_2}{s_1 + s_2} = 30 \text{ km/h}$$

# What Type of Mean?



$$s = \frac{d}{t}$$

- Suppose Herbie goes for a time  $t$ , say 10 minutes, at speed  $s_1$ , say 60 km/h, and for the same amount of time  $t$  at speed  $s_2$ , say 20 km/h. What is Herbie's average speed during the travel?

$$\frac{s_1 + s_2}{2} = 40 \text{ km/h}$$

$$2 \frac{s_1 \cdot s_2}{s_1 + s_2} = 30 \text{ km/h}$$

$$\bar{s} = \frac{\text{total distance}}{\text{total time}} = \frac{s_1 \cdot t + s_2 \cdot t}{t + t} = \frac{s_1 + s_2}{2}$$

# What Type of Mean?

Quando usava media aritmetica:  
 Mean of two numbers,  $s_1$  and  $s_2$ ,  
 which are ratios and where the  
 denominator  $t$  is fixed

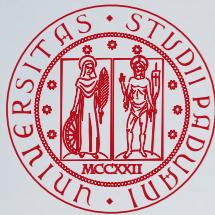


$$s = \frac{d}{t}$$

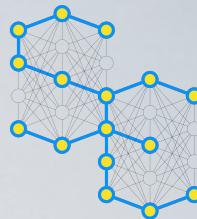
Suppose Herbie goes for a time  $t$ , say 10 minutes, at speed  $s_1$ , say 60 km/h, and for the same amount of time  $t$  at speed  $s_2$ , say 20 km/h. What is Herbie's average speed during the travel?

$$2 \frac{s_1 \cdot s_2}{s_1 + s_2} = 30 \text{ km/h}$$

$$\bar{s} = \frac{\text{total distance}}{\text{total time}} = \frac{s_1 \cdot t + s_2 \cdot t}{t + t} = \frac{s_1 + s_2}{2}$$



# Back to Precision, Recall, and F-measure



$$P = \frac{|A \cap B|}{|B|}$$

$$R = \frac{|A \cap B|}{|A|}$$

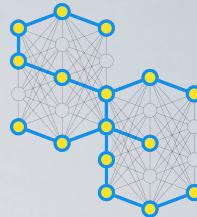


- F-measure is a mean between two ratios whose numerator is fixed, thus the harmonic mean looks more appropriate

$$F = 2 \frac{P \cdot R}{P + R} = 2 \frac{\frac{|A \cap B|}{|B|} \cdot \frac{|A \cap B|}{|A|}}{\frac{|A \cap B|}{|B|} + \frac{|A \cap B|}{|A|}} = \frac{|A \cap B|}{\frac{|A| + |B|}{2}}$$

- F-measure scores what went good  $|A \cap B|$ , i.e. the relevant retrieved documents, against the average of what might have been  $|A|$ , i.e. the relevant documents, and what actually happened  $|B|$ , i.e. the retrieved documents

# Rank-based Evaluation Measures



- Precision at Document Cut-off:

$$P(k) = \frac{1}{k} \sum_{n=1}^k r_n$$

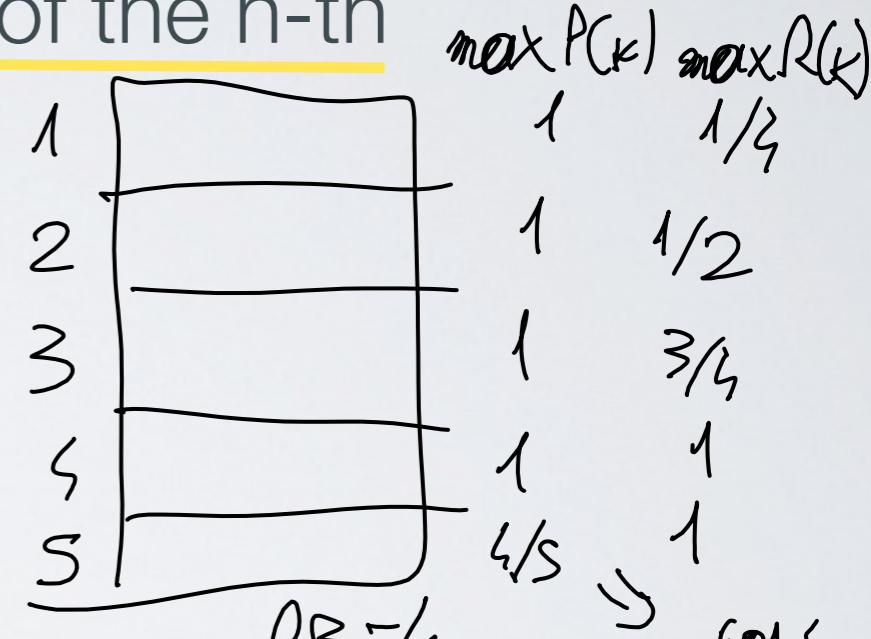
where  $r_k \in \{0, 1\}$  is the relevance degree of the n-th document

*Typo: somma dei k doc.*

- Recall at Document Cut-off:

$$R(k) = \frac{1}{RB} \sum_{n=1}^k r_n$$

where  $RB = |A|$  is the recall base, i.e. the total number of relevant documents



- Rprec is Precision computed at the recall base

*precisione esatta*

$$Rprec = P(RB)$$

*retrieval perfetto  
scegli contesto per  
delle cose misere*

# Rank-based Measures: Example of Precision and Recall

Universitas Studii Regentum

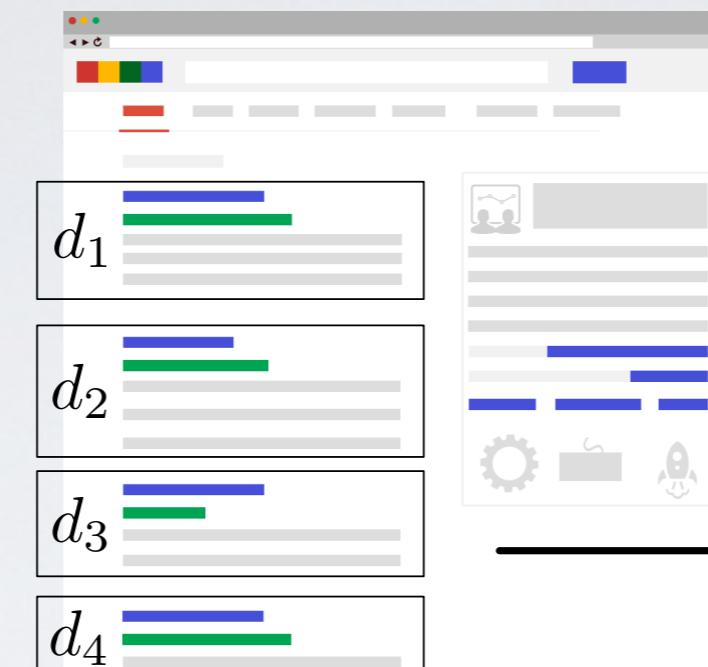
## Topic

Assume

- $RB = 8$  relevant documents in total
- Lenient mapping to binary relevance degrees



## Run



## Assessed Run

|    |                    |
|----|--------------------|
| 1  | Highly Relevant    |
| 2  | Not Relevant       |
| 3  | Partially Relevant |
| 4  | Fairly Relevant    |
| 5  | Not Relevant       |
| 6  | Not Relevant       |
| 7  | Not Relevant       |
| 8  | Fairly Relevant    |
| 9  | Not Relevant       |
| 10 | Not Relevant       |

## Binary Weighted Assessed Run

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

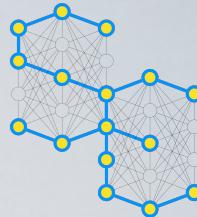
$$P(5) = \frac{3}{5} = 0.600$$

$$R(5) = \frac{3}{8} = 0.375$$

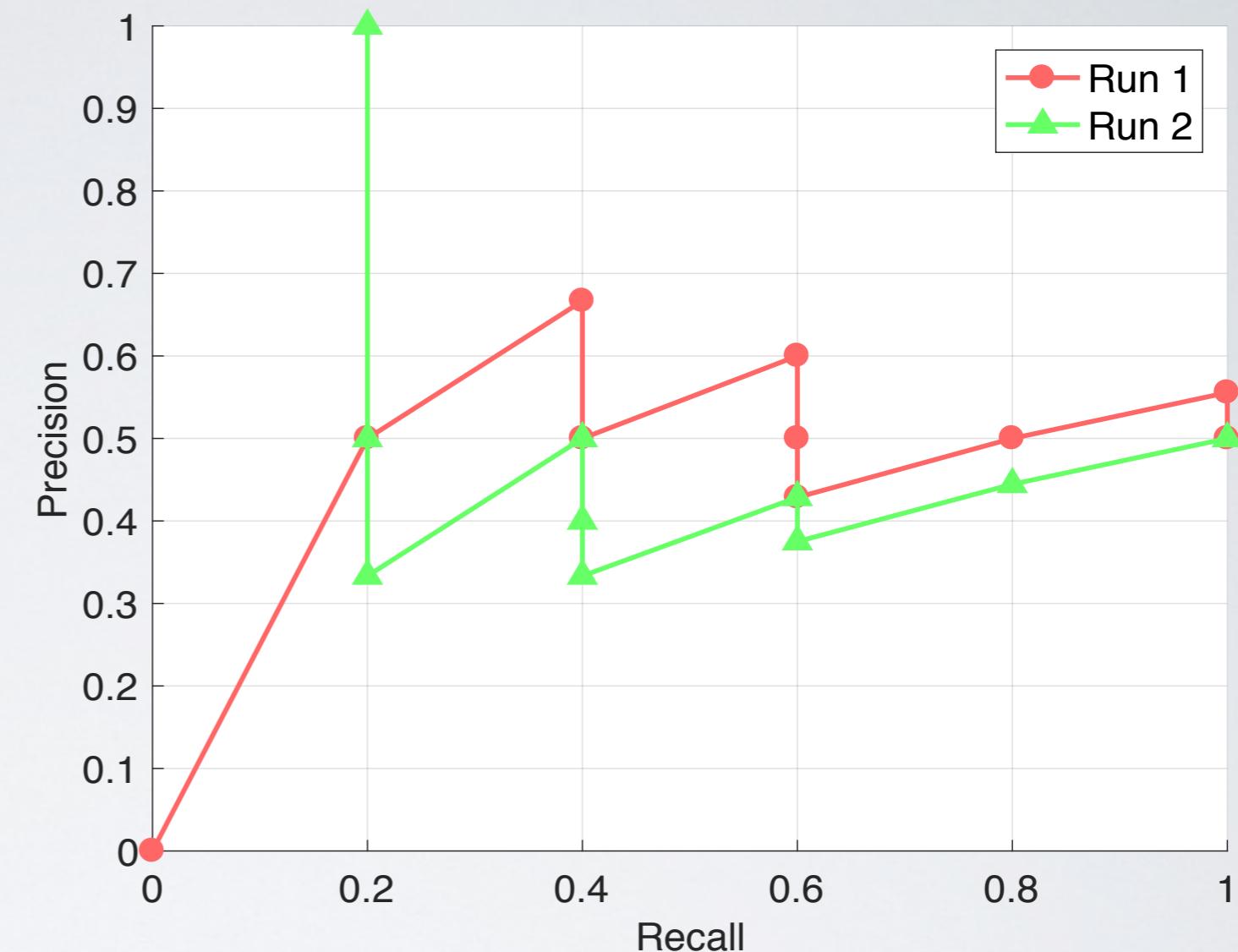
$$R_{prec} = P(8) = \frac{4}{8} = 0.500$$



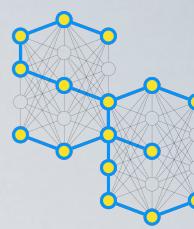
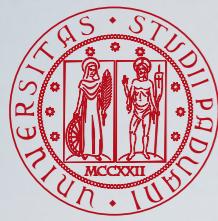
# Precision-Recall Curve



| Run1 |   | Run2     |          |
|------|---|----------|----------|
| 1    | 0 | P = 0.00 | P = 1.00 |
| 2    | 1 | R = 0.00 | R = 0.20 |
| 3    | 1 | P = 0.50 | P = 0.50 |
| 4    | 0 | R = 0.20 | R = 0.20 |
| 5    | 1 | P = 0.66 | P = 0.33 |
| 6    | 0 | R = 0.40 | R = 0.20 |
| 7    | 0 | P = 0.50 | P = 0.50 |
| 8    | 1 | R = 0.40 | R = 0.40 |
| 9    | 1 | P = 0.60 | P = 0.40 |
| 10   | 0 | R = 0.60 | R = 0.40 |
| 11   | 0 | P = 0.50 | P = 0.33 |
| 12   | 0 | R = 0.60 | R = 0.40 |
| 13   | 0 | P = 0.42 | P = 0.42 |
| 14   | 1 | R = 0.60 | R = 0.60 |
| 15   | 1 | P = 0.50 | P = 0.37 |
| 16   | 0 | R = 0.80 | R = 0.60 |
| 17   | 1 | P = 0.55 | P = 0.44 |
| 18   | 1 | R = 1.00 | R = 0.80 |
| 19   | 0 | P = 0.50 | P = 0.50 |
| 20   | 0 | R = 1.00 | R = 1.00 |



- Assume  $RB = 5$  relevant documents in total
- The Precision-Recall curve has a typical saw-tooth shape
  - We may have multiple Precision values for the same Recall value
  - It is difficult to compare runs because they may not have the same Recall values

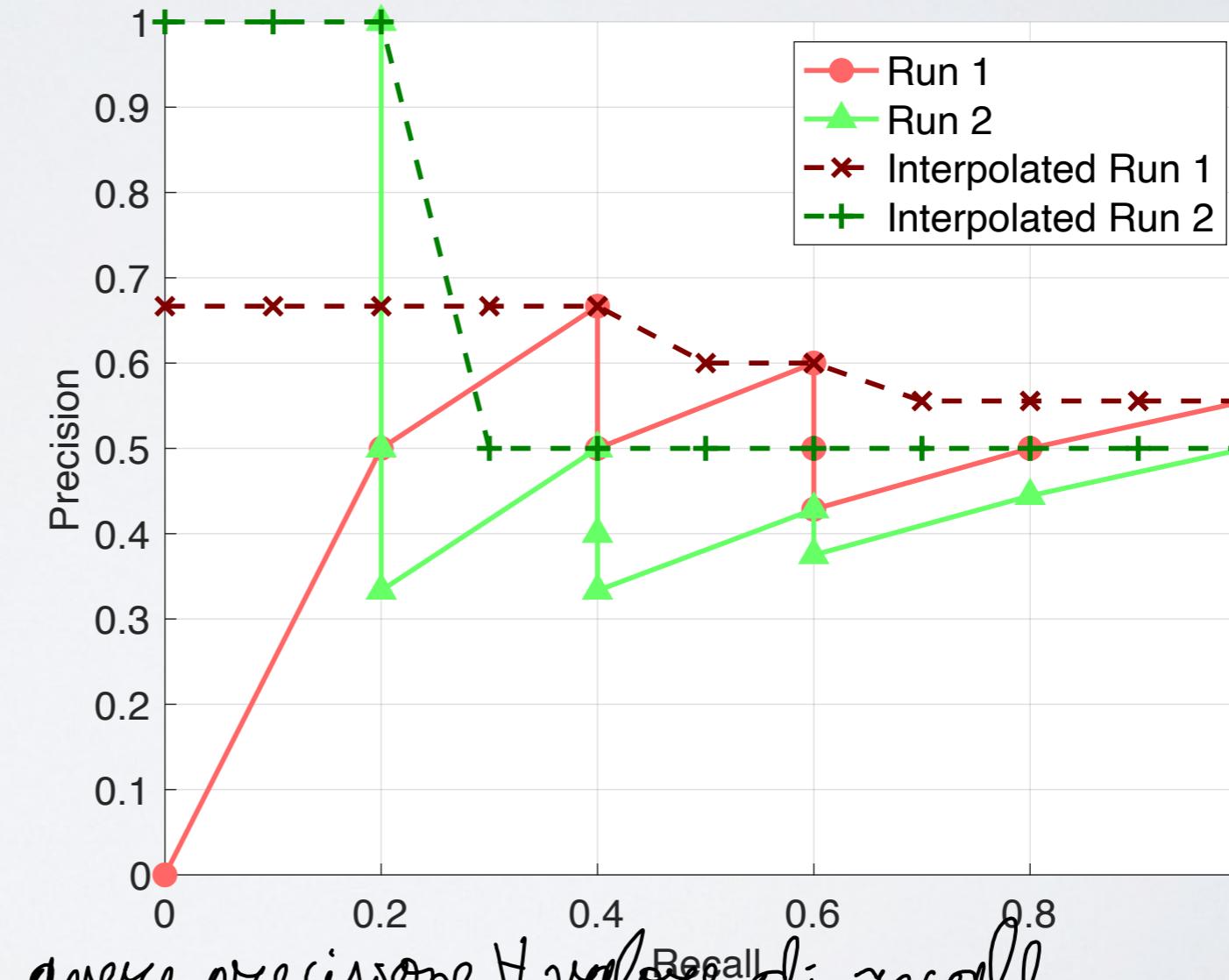


# Interpolated Precision-Recall Curve

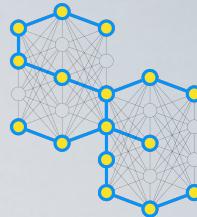
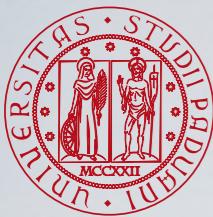
|    | Run1                              | Run2                              |
|----|-----------------------------------|-----------------------------------|
| 0  | iP = 0.66<br>P = 0.00<br>R = 0.00 | 1                                 |
| 1  | iP = 0.66<br>P = 0.50<br>R = 0.20 | 0                                 |
| 2  | iP = 0.66<br>P = 0.66<br>R = 0.40 | 0                                 |
| 3  | 0                                 | iP = 1.00<br>P = 1.00<br>R = 0.20 |
| 4  | iP = 0.66<br>P = 0.50<br>R = 0.40 | 1                                 |
| 5  | iP = 0.60<br>P = 0.60<br>R = 0.60 | 0                                 |
| 6  | iP = 0.60<br>P = 0.50<br>R = 0.60 | 0                                 |
| 7  | 0                                 | iP = 0.60<br>P = 0.42<br>R = 0.60 |
| 8  | iP = 0.55<br>P = 0.50<br>R = 0.80 | 1                                 |
| 9  | iP = 0.55<br>P = 0.55<br>R = 1.00 | 0                                 |
| 10 | iP = 0.55<br>P = 0.50<br>R = 1.00 | 1                                 |

To interpolate Precision at standard Recall value  $R_j$  we use the maximum Precision obtained for any actual Recall value  $R$  greater than or equal to  $R_j$

$$iP@R_j = \max_{R \geq R_j} P@R$$



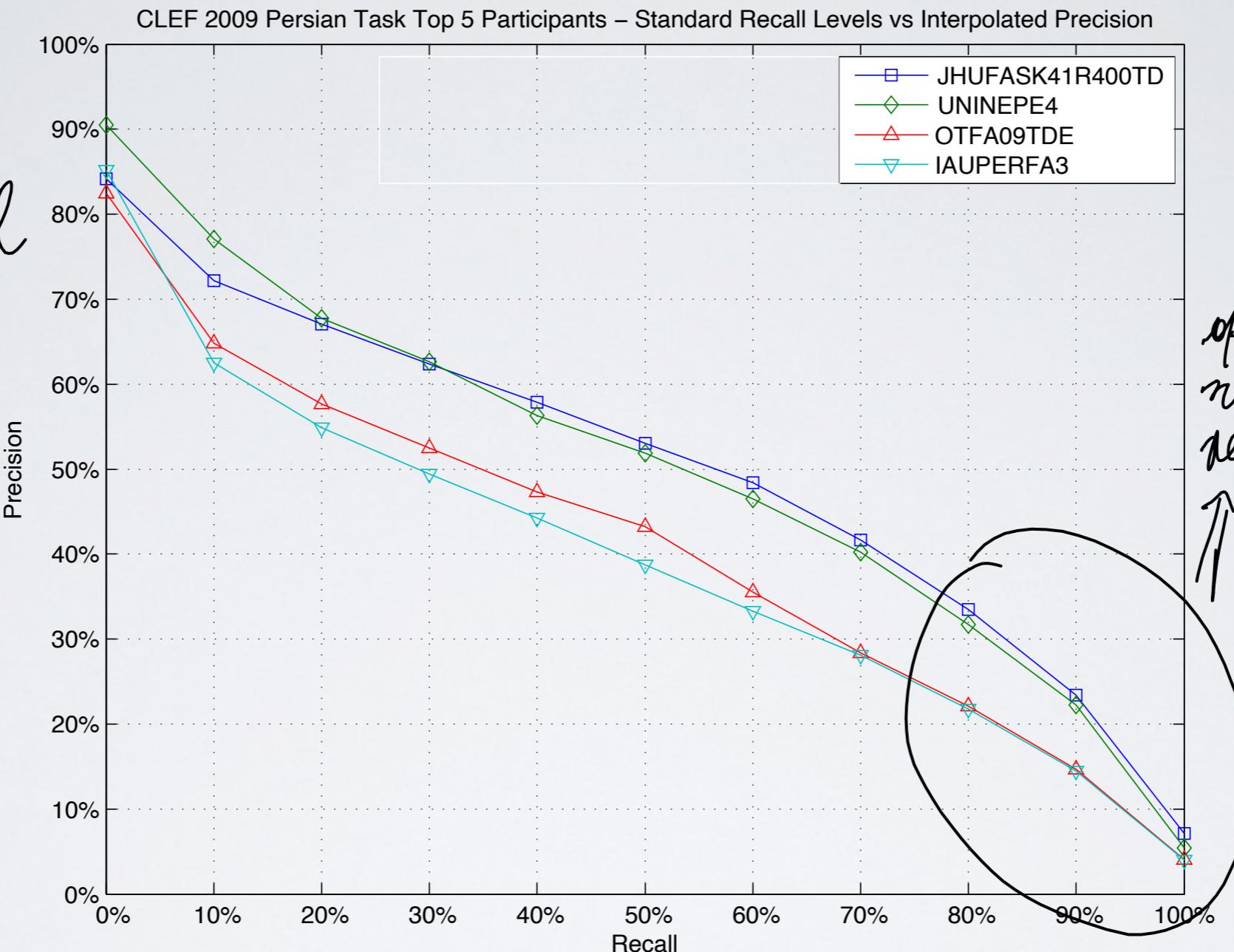
per avere precisione il valore di recall



# 11 points Interpolated Precision-Recall Curve

messaggi tra valori di Recall non sono essere molto diversi ↓ tenti come giochi doc diversi tra valori

questi si leggono non variano bene per recall-based tasks ↑

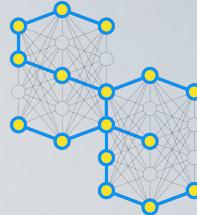
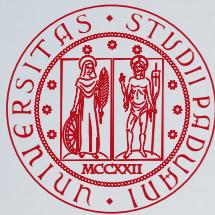


- Standard Interpolated Precision-Recall curves exhibit a typical inverse relationship among Precision and Recall, indicating a trade-off between these two goals of effectiveness

Cleverdon, C. W. (1972). On the inverse relationship of recall and precision. *Journal of Documentation*, 28(3):195–201.

Buckland, M. and Gey, F. (1994). The relationship between Recall and Precision. *Journal of the American Society for Information Science and Technology (JASIST)*, 45(1):12–19.

Eggle, L. (2008). The measures precision, recall, fallout and miss as a function of the number of retrieved documents and their mutual interrelations. *Information Processing & Management*, 44(2):856–876.



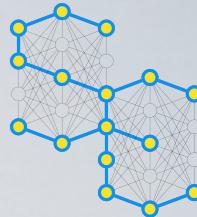
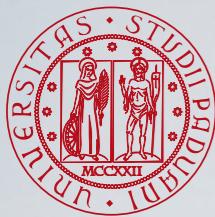
# Using trec\_eval to Compute Set-based Measures

- To compute set-based evaluation measures (Precision, Recall, F-measure) run

```
trec_eval -q -m set pool.txt run.txt
```

- q prints topic-by-topic results
- m selects which measures to compute,  
use set for set-based evaluation measures

```
[ferro~/trec_eval.9.0$ ./trec_eval -q -m set ../data/CLEF2009-Persian/pool/  
AH-PERSIAN-CLEF2009.txt ../data/CLEF2009-Persian/runs/UNINEPE1.txt  
num_ret 601-AH 1000  
num_rel 601-AH 89  
num_rel_ret 601-AH 67  
utility 601-AH -866.0000  
set_P 601-AH 0.0670  
set_relative_P 601-AH 0.7528  
set_recall 601-AH 0.7528  
set_map 601-AH 0.0504  
set_F 601-AH 0.1230  
num_ret 602-AH 1000  
num_rel 602-AH 93  
num_rel_ret 602-AH 92
```



# Using trec\_eval to Compute Set-based Measures

- To compute set-based evaluation measures (Precision, Recall, F-measure) run

```
trec_eval -q -m set pool.txt run.txt
```

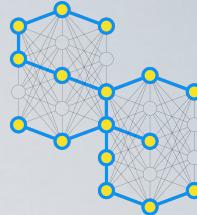
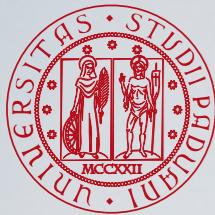
- q prints topic-by-topic results
- m selects which measures to compute,  
use set for set-based evaluation measures

```
[ferro~/trec_eval.9.0$ ./trec_eval -q -m set ../data/CLEF2009-Persian/pool/  
AH-PERSIAN-CLEF2009.txt ../data/CLEF2009-Persian/runs/UNINEPE1.txt  
num_ret 601-AH 1000  
num_rel 601-AH 89  
num_rel_ret 601-AH 67  
utility 601-AH -866.0000  
set_P 601-AH 0.067  
set_relative_P 601-AH 0.752  
set_recall 601-AH 0.752  
set_map 601-AH 0.050  
set_F 601-AH 0.123  
num_ret 602-AH 1000  
num_rel 602-AH 93  
num_rel_ret 602-AH 92
```

$\text{num\_ret}$  is the total number of retrieved documents, i.e.  $|B|$  - typically 1,000 documents

$\text{num\_rel}$  is the total number of relevant documents, i.e.  $|A|$

$\text{num\_rel\_ret}$  is the total number of relevant retrieved documents, i.e.  $|A \cap B|$



# Using trec\_eval to Compute Set-based Measures

- To compute set-based evaluation measures (Precision, Recall, F-measure) run

**trec\_eval -q -m set pool.txt run.txt**

- q prints topic-by-topic results
- m selects which measures to compute,  
use set for set-based evaluation measures

**set\_P** is set-based Precision

**set\_recall** is set-based Recall

```
[ferro~/trec_eval.9.0$ ./trec_eval -q -m set ../../data/CLEF2009-Persian/pool/1  
AH-PERSIAN-CLEF2009.txt ../../data/CLEF2009-Persian/runs/run1  
num_ret 601-AH 1000  
num_rel 601-AH 89  
num_rel_ret 601-AH 67  
utility 601-AH -866.0000  
set_P 601-AH 0.0670  
set_relative_P 601-AH 0.7528  
set_recall 601-AH 0.7528  
set_map 601-AH 0.0504  
set_F 601-AH 0.1230  
num_ret 602-AH 1000  
num_rel 602-AH 93  
num_rel_ret 602-AH 92
```

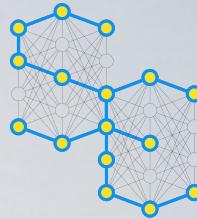
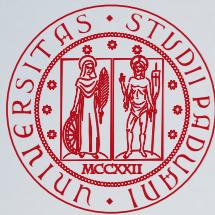
# Using trec\_eval to Compute Rank-based Measures

```
[ferro@trec_eval.9.0$ ./trec_eval -q -m all_trec ../data/CLEF2009-Persian/
pool/AH-PERSIAN-CLEF2009.txt ../data/CLEF2009-Persian/runs/UNINEPE1.txt
num_ret          601-AH 1000
num_rel          601-AH 89
num_rel_ret      601-AH 67
map              601-AH 0.1856
Rprec            601-AH 0.2360
bpref            601-AH 0.1788
recip_rank       601-AH 1.0000
iprec_at_recall_0.00 601-AH 1.0000
iprec_at_recall_0.10 601-AH 0.4737
iprec_at_recall_0.20 601-AH 0.2500
iprec_at_recall_0.30 601-AH 0.2248
iprec_at_recall_0.40 601-AH 0.2139
iprec_at_recall_0.50 601-AH 0.1852
iprec_at_recall_0.60 601-AH 0.1333
iprec_at_recall_0.70 601-AH 0.0947
iprec_at_recall_0.80 601-AH 0.0000
iprec_at_recall_0.90 601-AH 0.0000
iprec_at_recall_1.00 601-AH 0.0000
P_5              601-AH 0.4000
P_10             601-AH 0.3000
P_15             601-AH 0.4000
P_20             601-AH 0.4500
P_30             601-AH 0.3667
P_100            601-AH 0.2300
P_200            601-AH 0.1950
P_500            601-AH 0.1200
P_1000            601-AH 0.0670
relstring        601-AH '1100001000'
recall_5          601-AH 0.0225
recall_10         601-AH 0.0337
recall_15         601-AH 0.0674
recall_20         601-AH 0.1011
recall_30         601-AH 0.1236
recall_100        601-AH 0.2584
recall_200        601-AH 0.4382
recall_500        601-AH 0.6742
recall_1000        601-AH 0.7528
infAP            601-AH 0.1856
Rprec_mult_0.20  601-AH 0.4444
Rprec_mult_0.40  601-AH 0.3056
```

Handwritten annotations on the terminal window:

- A large bracket on the left side groups the first 10 rows of the output.
- Handwritten text above the bracket: "per plottarli: attenzione al distanziale e ai cut-offs".
- An arrow points from the handwritten text to the "relstring" row.
- Handwritten text near the bottom of the list: "meglio non usare linee continue".
- An arrow points from the handwritten text to the "recall\_5" row.

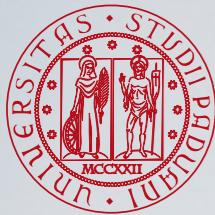
- To compute all the evaluation measures run **trec\_eval -q -m all\_trec pool.txt run.txt**
- num\_ret**, **num\_rel**, and **num\_rel\_ret** have the usual meaning
- Rprec** is Precision at the recall base
- iprec\_\*** is interpolated Precision
- P\_\*** is Precision at Document Cut-off
- R\_\*** is Recall at Document Cut-off



# Compute it Yourself!

```
% A topic of a run.  
% Each position is a retrieved document; first element of the vector is  
% the top of the rank.  
% Assume binary relevance: 0 for not relevant; 1 for relevant.  
run = [0 1 1 0 1 0 0 1 1 0];  
  
% Recall Base  
RB = 5;  
  
% Recall at each rank position  
recall = cumsum(run) ./ RB;  
  
% Precision at each rank position  
precision = cumsum(run) ./ (1:length(run));  
  
% Rprec  
rprec = NaN; → valore default dipende da misure da calcolare  
if (RB <= length(run))  
    rprec = precision(RB);  
end  
  
% Standard Recall levels  
recallLevel = 0:0.1:1;  
  
% 11-points interpolated precision  
iPrecision = zeros(1, length(recallLevel));  
  
for i = 1:length(recallLevel)  
  
    % find the recall values R that are greater than or equal to the  
    % current standard recall level R_i  
    idx = recall >= recallLevel(i);  
  
    % if there is any such recall value, take the maximum precision  
    % otherwise interpolated precision is zero at that standard recall  
    % level, i.e. skip any assignment  
    if (any(idx))  
        iPrecision(i) = max(precision(idx));  
    end  
end
```

*↓*  
eng. non conviene -1 se calcolo media (non dà errore)  
*↓*  
O può comunque dare problemi

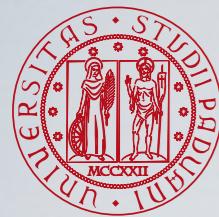


# Compute it Yourself!



```
% A topic of a run.  
% Each position is a retrieved document; first element of the vector is  
% the top of the rank.  
% Assume binary relevance: 0 for not relevant; 1 for relevant.  
run = [0 1 1 0 1 0 0 1 1 0];  
  
% Recall Base  
RB = 5;  
  
% Recall at each rank position  
recall = cumsum(run) ./ RB;  
  
% Precision at each rank position  
precision = cumsum(run) ./ (1:length(run));  
  
% Rprec  
rprec = NaN;  
if (RB <= length(run))  
    rprec = precision(RB);  
end  
  
% Standard Recall levels  
recallLevel = 0:0.1:1;  
  
% 11-points interpolated precision  
iPrecision = zeros(1, length(recallLevel));  
  
for i = 1:length(recallLevel)  
  
    % find the recall values R that are greater than or equal to the  
    % current standard recall level R_i  
    idx = recall >= recallLevel(i);  
  
    % if there is any such recall value, take the maximum precision  
    % otherwise interpolated precision is zero at that standard recall  
    % level, i.e. skip any assignment  
    if (any(idx))  
        iPrecision(i) = max(precision(idx));  
    end  
end
```

What happens in corner cases?

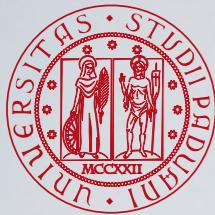


# Compute it Yourself!

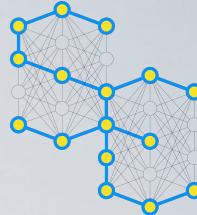


```
% A topic of a run.  
% Each position is a retrieved document; first element of the vector is  
% the top of the rank.  
% Assume binary relevance: 0 for not relevant; 1 for relevant.  
run = [0 1 1 0 1 0 0 1 1 0];  
  
% Recall Base  
RB = 5;  
  
% Recall at each rank position  
recall = cumsum(run) ./ RB;  
  
% Precision at each rank position  
precision = cumsum(run) ./ (1:length(run));  
  
% Rprec  
rprec = NaN;  
if (RB <= length(run))  
    rprec = precision(RB);  
end  
  
% Standard Recall levels  
recallLevel = 0:0.1:1;  
  
% 11-points interpolated precision  
iPrecision = zeros(1, length(recallLevel));  
  
for i = 1:length(recallLevel)  
  
    % find the recall values R that are greater than or equal to the  
    % current standard recall level R_i  
    idx = recall >= recallLevel(i);  
  
    % if there is any such recall value, take the maximum precision  
    % otherwise interpolated precision is zero at that standard recall  
    % level, i.e. skip any assignment  
    if (any(idx))  
        iPrecision(i) = max(precision(idx));  
    end  
end
```

What if I wish set-based Precision and Recall?

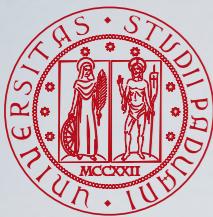


# Compute it Yourself!

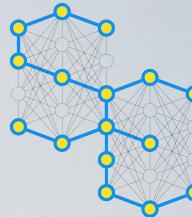


```
% A topic of a run.  
% Each position is a retrieved document; first element of the vector is  
% the top of the rank.  
% Assume binary relevance: 0 for not relevant; 1 for relevant.  
run = [0 1 1 0 1 0 0 1 1 0];  
  
% Recall Base  
RB = 5;  
  
% Recall at each rank position  
recall = cumsum(run) ./ RB;  
  
% Precision at each rank position  
precision = cumsum(run) ./ (1:length(run));  
  
% Rprec  
rprec = NaN;  
if (RB <= length(run))  
    rprec = precision(RB);  
end  
  
% Standard Recall levels  
recallLevel = 0:0.1:1;  
  
% 11-points interpolated precision  
iPrecision = zeros(1, length(recallLevel));  
  
for i = 1:length(recallLevel)  
  
    % find the recall values R that are greater than or equal to the  
    % current standard recall level R_i  
    idx = recall >= recallLevel(i);  
  
    % if there is any such recall value, take the maximum precision  
    % otherwise interpolated precision is zero at that standard recall  
    % level, i.e. skip any assignment  
    if (any(idx))  
        iPrecision(i) = max(precision(idx));  
    end  
end
```

What happens if  
there are no relevant  
documents?



# Rank-based Measures: Average Precision



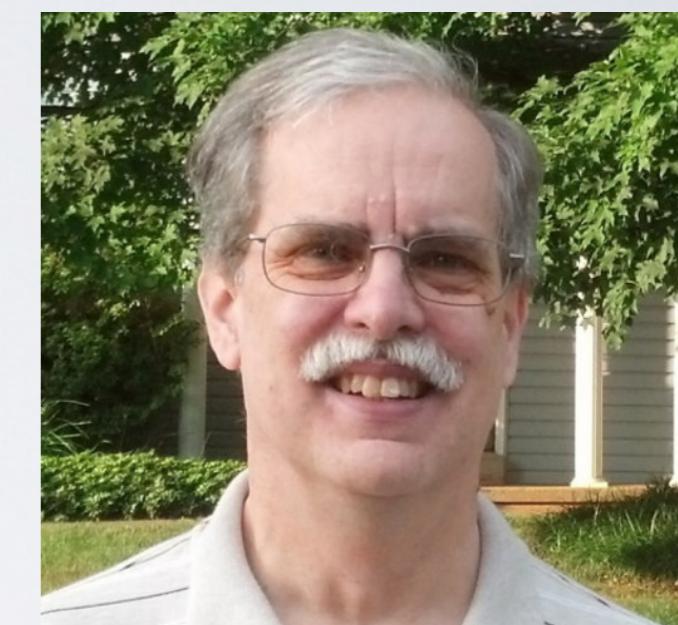
"average":  
specie di  
media di  
valori  
precisione

$$AP = \frac{1}{RB} \sum_{k \in \mathcal{R}} P(k) = \frac{1}{RB} \sum_{n=1}^N \underbrace{\left( \frac{1}{n} \sum_{m=1}^n r_m \right)}_{P(n)} r_n =$$

$$= \frac{rr}{RB} \cdot \underbrace{\frac{1}{rr} \sum_{k \in \mathcal{R}} P(k)}_{\text{arithmetic mean of } P(k)}$$

where

- $\mathcal{R}$  is the set of the rank positions of the relevant retrieved documents
- $rr = |\mathcal{R}|$  is the total number of relevant retrieved documents
- $N$  is the total number of retrieved documents, i.e. the length of the run
- The **Mean Average Precision (MAP)** is the mean of AP over a set of topics
  - Differently from the other measures, this mean has its own name since it is the most widely used single number to summarise the whole performance of a system



Chris Buckley

Buckley, C. and Voorhees, E. M. (2005). Retrieval System Evaluation. In Harman, D. K. and Voorhees, E. M., editors, *TREC. Experiment and Evaluation in Information Retrieval*, pages 53–78. MIT Press, Cambridge (MA), USA.

# Rank-based Measures: Example of Average Precision

UNIVERSITAS STUDII  
DUENTINI  
MCCXXII

## Topic

Assume

- $RB = 8$  relevant documents in total
- Lenient mapping to binary relevance degrees



## Run



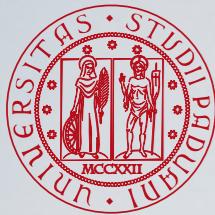
## Assessed Run

|    |                    |
|----|--------------------|
|    | Highly Relevant    |
| 1  | Not Relevant       |
| 2  | Partially Relevant |
| 3  | Fairly Relevant    |
| 4  | Not Relevant       |
| 5  | Not Relevant       |
| 6  | Not Relevant       |
| 7  | Not Relevant       |
| 8  | Fairly Relevant    |
| 9  | Not Relevant       |
| 10 | Not Relevant       |

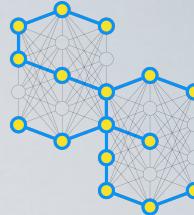
## Binary Weighted Assessed Run

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

$$\begin{aligned}
 AP &= \frac{1}{RB} \left( P(1) + P(3) + P(4) + P(8) \right) \\
 &= \frac{1}{8} \left( 1 + \frac{2}{3} + \frac{3}{4} + \frac{4}{8} \right) = \frac{35}{96} = 0.36
 \end{aligned}$$



# Computing AP Using trec\_eval

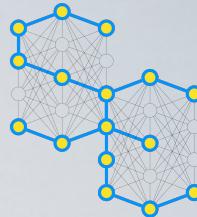
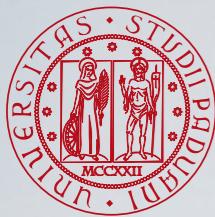


```
[ferro~trec_eval.9.0$ ./trec_eval -q -m all_trec ../data/CLEF2009-Persian/pool/AH-PERSIAN-CLEF2009.txt ../data/CLEF2009-Persian/runs/UNINEPE1.txt
num_ret          601-AH 1000
num_rel          601-AH 89
num_rel_ret      601-AH 67
map              601-AH 0.1856
Rprec            601-AH 0.2360
bpref             601-AH 0.1788
recip_rank       601-AH 1.0000
iprec_at_recall_0.00 601-AH 1.0000
iprec_at_recall_0.10 601-AH 0.4737
iprec_at_recall_0.20 601-AH 0.2500
iprec_at_recall_0.30 601-AH 0.2216
iprec_at_recall_0.40 601-AH 0.2139
iprec_at_recall_0.50 601-AH 0.1802
iprec_at_recall_0.60 601-AH 0.1333
```

Beware of trec eval calling map what actually is AP on a topic.  
However, when all is reported, i.e. when trec eval is averaging measures over topics, what it still calls map is the true MAP, i.e. the mean of AP over topics

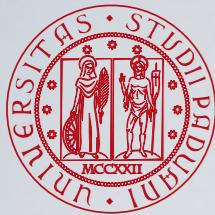
- To compute all the evaluation measures run

**trec\_eval -q -m all\_trec pool.txt run.txt**

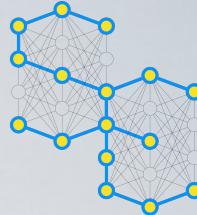


# Compute it Yourself!

```
% A topic of a run.  
% Each position is a retrieved document; first element of the vector is  
% the top of the rank.  
% Assume binary relevance: 0 for not relevant; 1 for relevant.  
run = [0 1 1 0 1 0 0 1 1 0];  
  
% Recall Base  
RB = 5;  
  
% Find the rank positions of the relevant retrieved documents  
ranks = find(run);  
  
% compute average precision, if any relevant document has been retrieved  
if ~isempty(ranks)  
    ap = sum(cumsum(run(ranks)) ./ ranks) ./ RB;  
else  
    ap = 0;  
end
```

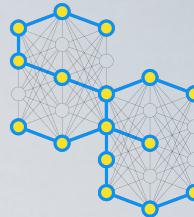
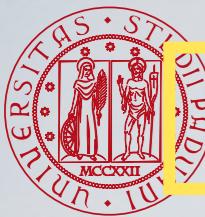


# Compute it Yourself!

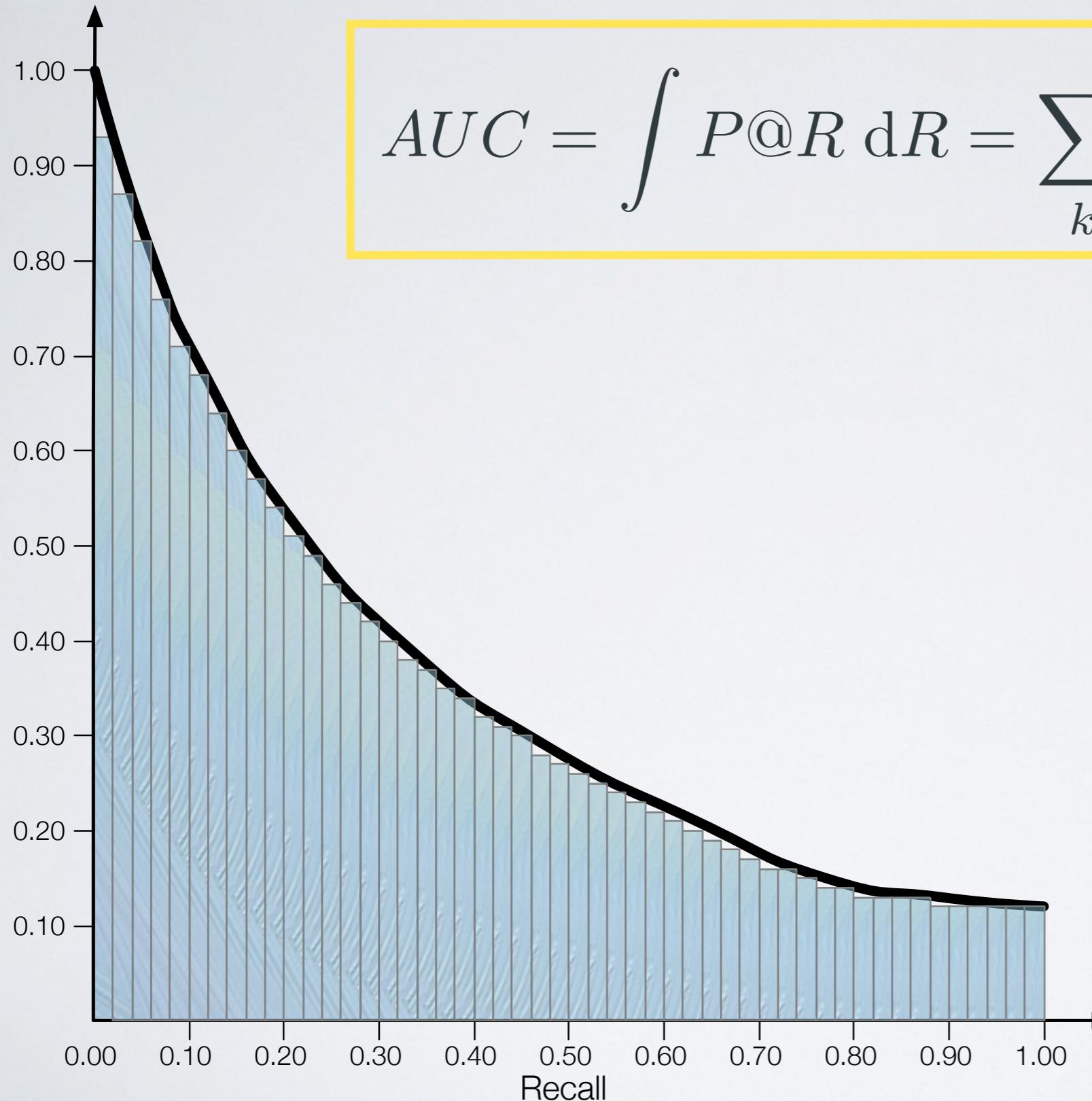


```
% A topic of a run.  
% Each position is a retrieved document; first element of the vector is  
% the top of the rank.  
% Assume binary relevance: 0 for not relevant; 1 for relevant.  
run = [0 1 1 0 1 0 0 1 1 0];  
  
% Recall Base  
RB = 5;  
  
% Find the rank positions of the relevant retrieved documents  
ranks = find(run);  
  
% compute average precision, if any relevant document has been retrieved  
if ~isempty(ranks)  
    ap = sum(cumsum(run(ranks)) ./ ranks) ./ RB;  
else  
    ap = 0;  
end
```

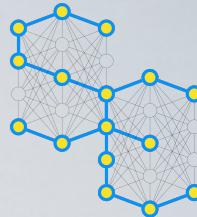
What happens if there are no relevant documents?



# Area under the Precision-Recall Curve



The **Area Under the Precision-Recall Curve (AUC)** is an important indicator of the overall system effectiveness, summarising the trade-off between Precision and Recall.

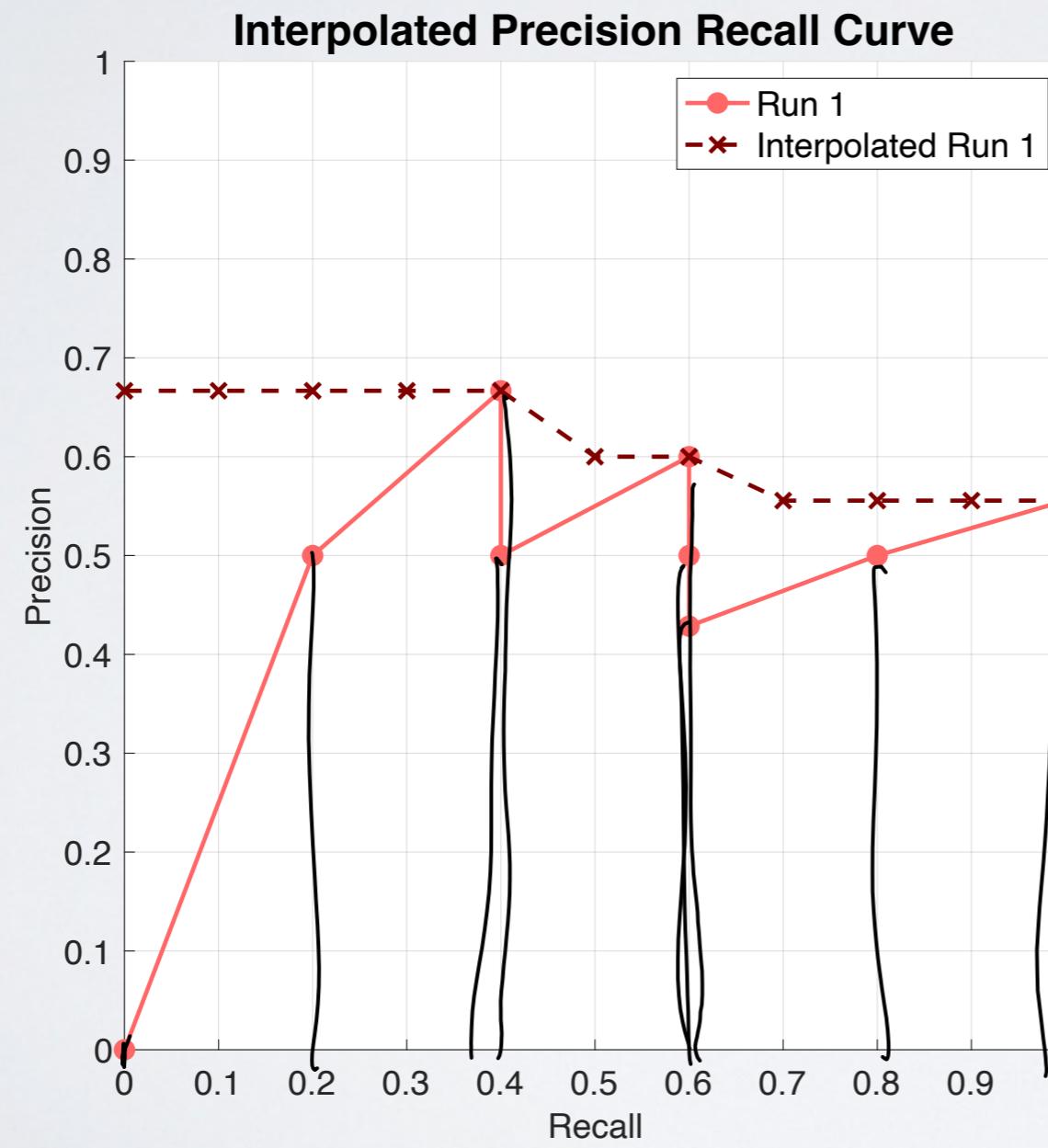


# Computing the Area under the Precision-Recall Curve

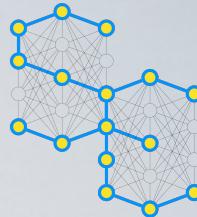
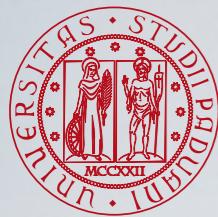
$$AUC = \sum_{n=1}^N P(n) (R(n) - R(n-1)) \quad \text{assuming } R(0) = 0$$

Run1

|   |                      |
|---|----------------------|
| 0 | P = 0.00<br>R = 0.00 |
| 1 | P = 0.50<br>R = 0.20 |
| 1 | P = 0.66<br>R = 0.40 |
| 0 | P = 0.50<br>R = 0.40 |
| 1 | P = 0.60<br>R = 0.60 |
| 0 | P = 0.50<br>R = 0.60 |
| 0 | P = 0.42<br>R = 0.60 |
| 0 | P = 0.50<br>R = 0.80 |
| 1 | P = 0.55<br>R = 1.00 |
| 0 | P = 0.50<br>R = 1.00 |



$$\begin{aligned} AUC &= 0.00(0.00 - 0.00) + 0.50(0.20 - 0.00) + \\ &\quad 0.66(0.40 - 0.20) + 0.50(0.40 - 0.40) + \\ &\quad 0.60(0.60 - 0.40) + 0.50(0.60 - 0.60) + \\ &\quad 0.42(0.60 - 0.60) + 0.50(0.80 - 0.60) + \\ &\quad 0.55(1.00 - 0.80) + 0.50(1.00 - 1.00) = \\ &= 0.5620 \end{aligned}$$

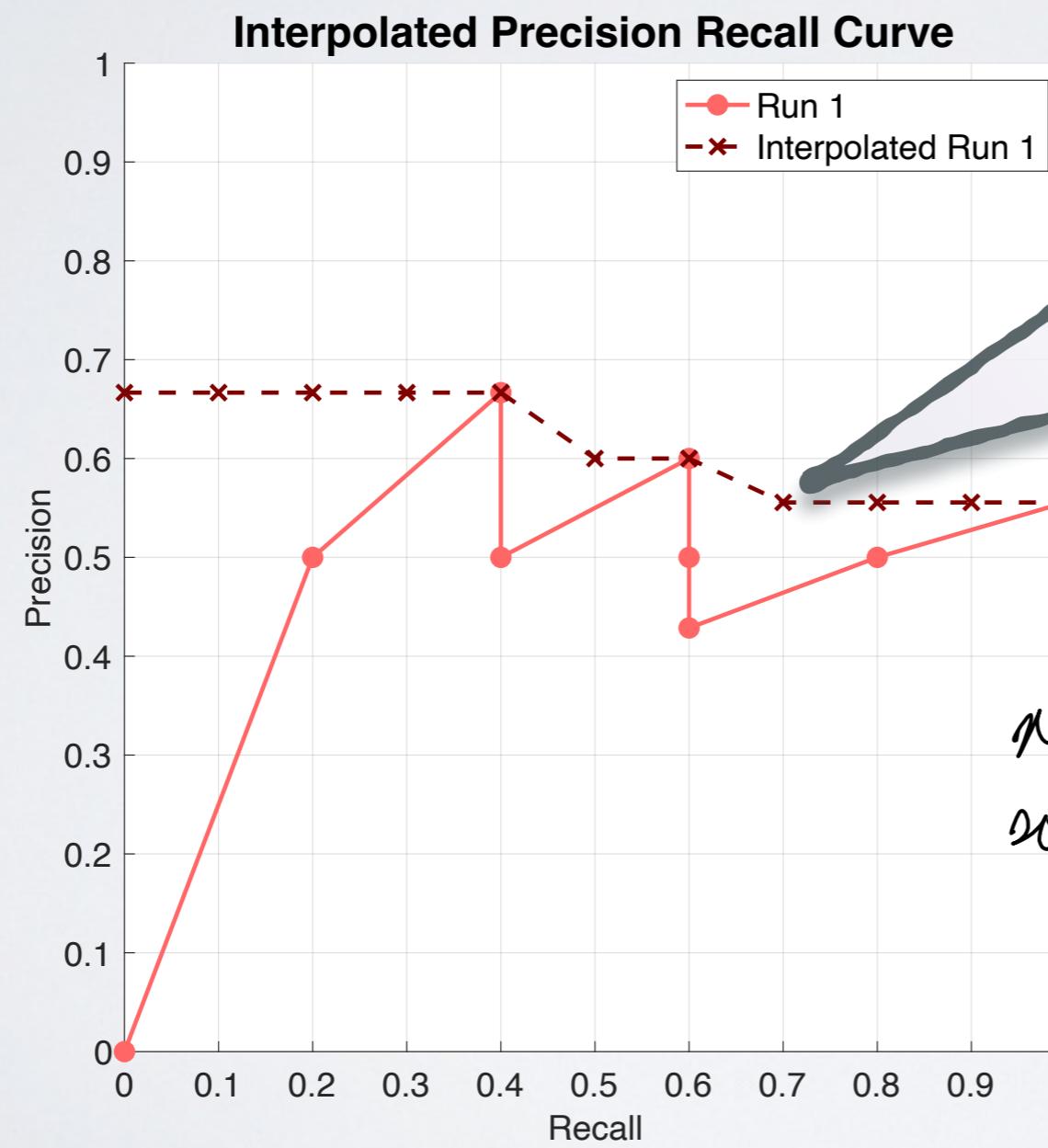


# Computing the Area under the Precision-Recall Curve

$$AUC = \sum_{n=1}^N P(n) (R(n) - R(n-1)) \quad \text{assuming } R(0) = 0$$

Run1

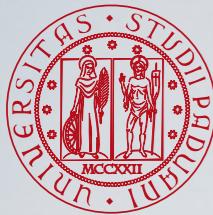
|   |                      |
|---|----------------------|
| 0 | P = 0.00<br>R = 0.00 |
| 1 | P = 0.50<br>R = 0.20 |
| 1 | P = 0.66<br>R = 0.40 |
| 0 | P = 0.50<br>R = 0.40 |
| 1 | P = 0.60<br>R = 0.60 |
| 0 | P = 0.50<br>R = 0.60 |
| 0 | P = 0.42<br>R = 0.60 |
| 1 | P = 0.50<br>R = 0.80 |
| 1 | P = 0.55<br>R = 1.00 |
| 0 | P = 0.50<br>R = 1.00 |



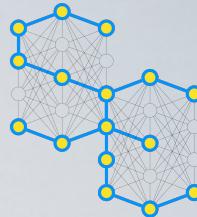
What happens if the interpolated Precision at Standard Recall Levels curve is used instead?

$$\begin{aligned} AUC &= 0.00(0.00 - 0.00) + 0.50(0.20 - 0.00) + \\ &+ 0.66(0.40 - 0.20) + 0.50(0.40 - 0.40) + \\ &+ 0.60(0.60 - 0.40) + 0.50(0.60 - 0.60) + \\ &+ 0.42(0.60 - 0.60) + 0.50(0.80 - 0.60) + \\ &+ 0.55(1.00 - 0.80) + 0.50(1.00 - 1.00) = \\ &= 0.5620 \end{aligned}$$

*performing a conservative estimate*



# Area under the Precision-Recall Curve and Average Precision



$$AUC = \sum_{n=1}^N P(n) (R(n) - R(n-1))$$

- When the n-th document is not relevant,  $R(n)$  is equal to  $R(n-1)$  and their difference goes to zero
- Therefore, we can sum only on  $\mathcal{R}$ , i.e. the set of the rank positions of the relevant retrieved documents

$$AUC = \sum_{k \in \mathcal{R}} P(k) (R(k) - R(k-1))$$

# Area under the Precision-Recall Curve and Average Precision

$$AUC = \sum_{k \in \mathcal{R}} P(k) (R(k) - R(k-1))$$

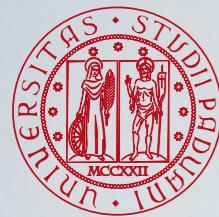
- Two adjacent rank positions differ just for one relevant document and thus

$$R(k) - R(k-1) = \underbrace{\frac{1}{RB} \sum_{n=1}^k r_n}_{R(k)} - \underbrace{\frac{1}{RB} \sum_{n=1}^{k-1} r_n}_{R(k-1)} = \frac{r_k}{RB} = \frac{1}{RB}$$

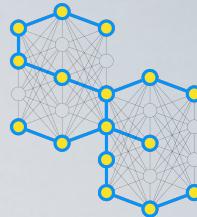
- Therefore AUC is equal to AP  $\Rightarrow$  per curva reale, non interpolata

- this is one motivation of the importance of AP

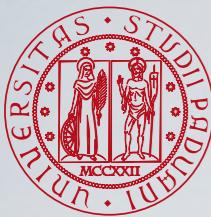
$$AUC = \frac{1}{RB} \sum_{k \in \mathcal{R}} P(k) = AP$$



# Compute it Yourself!



```
% A topic of a run.  
% Each position is a retrieved document; first element of the vector is  
% the top of the rank.  
% Assume binary relevance: 0 for not relevant; 1 for relevant.  
run = [0 1 1 0 1 0 0 1 1 0];  
  
% Recall Base  
RB = 5;  
  
% Recall at each rank position  
recall = cumsum(run) ./ RB;  
  
% Precision at each rank position  
precision = cumsum(run) ./ (1:length(run));  
  
% Find the rank positions of the relevant retrieved documents  
ranks = find(run);  
  
% compute average precision, if any relevant document has been retrieved  
if ~isempty(ranks)  
    ap = sum(cumsum(run(ranks)) ./ ranks) ./ RB;  
else  
    ap = 0;  
end  
  
% compute area under the precision-recall curve  
% Note that [0 recall] is the same as assuming R(0) = 0  
auc = sum(precision .* diff([0 recall]));
```



# Rank-based Measures: Discounted Cumulated Gain



rilevanza  
non è  
sempre  
la stessa  
↓  
considerare  
quanto abbiamo  
scrollato

$$DCG(k) = \begin{cases} \sum_{n=1}^k r_n & \text{if } k < b \\ DCG(k-1) + \frac{r_k}{\log_b(k)} & \text{if } k \geq b \end{cases}$$

considerare  
positivite in  
ranking

$$= \sum_{n=1}^k \frac{r_n}{\max(1, \log_b(n))}$$

questo caso  
per non  
considerare  
divisioni  
= negative

usato logaritmo ⇒ dividere  
per posizione per scendere  
troppo veloce

- where the base of the logarithm  $b$  indicates the patience of the user in scanning the result list

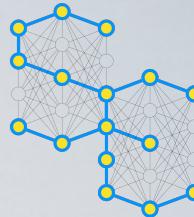
- $b = 2$  is an impatient user
- $b = 10$  is a patient user

- DCG naturally handles multi-graded relevance
- DCG does not depend on the recall base
- DCG is not bounded in  $[0, 1]$



Kalervo Järvelin    Jaana Kekäläinen

Järvelin, K. and Kekäläinen, J. (2002). Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446



$$DCG(k) = \begin{cases} \sum_{n=1}^k r_n & \text{if } k < b \\ DCG(k-1) + \frac{r_k}{\log_b(k)} & \text{if } k \geq b \end{cases}$$
$$= \sum_{n=1}^k \frac{r_n}{\max(1, \log_b(n))}$$

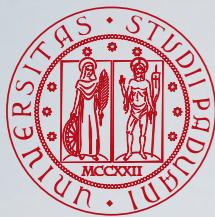
Cumulated Gain (CG)

- where the base of the logarithm  $b$  indicates the patience of the user in scanning the result list
  - $b = 2$  is an impatient user
  - $b = 10$  is a patient user
- DCG naturally handles multi-graded relevance
- DCG does not depend on the recall base
- DCG is not bounded in  $[0, 1]$

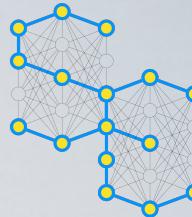


Kalervo Järvelin    Jaana Kekäläinen

Järvelin, K. and Kekäläinen, J. (2002). Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446



# Rank-based Measures: Discounted Cumulated Gain



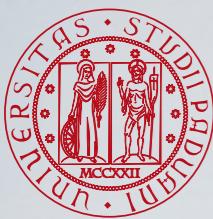
$$DCG(k) = \begin{cases} \sum_{n=1}^k r_n & \text{if } k < b \\ DCG(k-1) + \frac{r_k}{\log_b(k)} & \text{if } k \geq b \end{cases} = \sum_{n=1}^k \frac{r_n}{\max(1, \log_b(n))}$$

- where the base of the logarithm  $b$  indicates the patience of the user in scanning the result list
  - $b = 2$  is an impatient user
  - $b = 10$  is a patient user
- DCG naturally handles multi-graded relevance
- DCG does not depend on the recall base
- DCG is not bounded in  $[0, 1]$

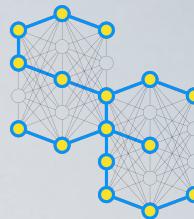


Kalervo Järvelin    Jaana Kekäläinen

Järvelin, K. and Kekäläinen, J. (2002). Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446



# Rank-based Measures: Example of Discounted Cumulated Gain



Assume

- $RB = 8$  relevant documents in total
- An impatient user

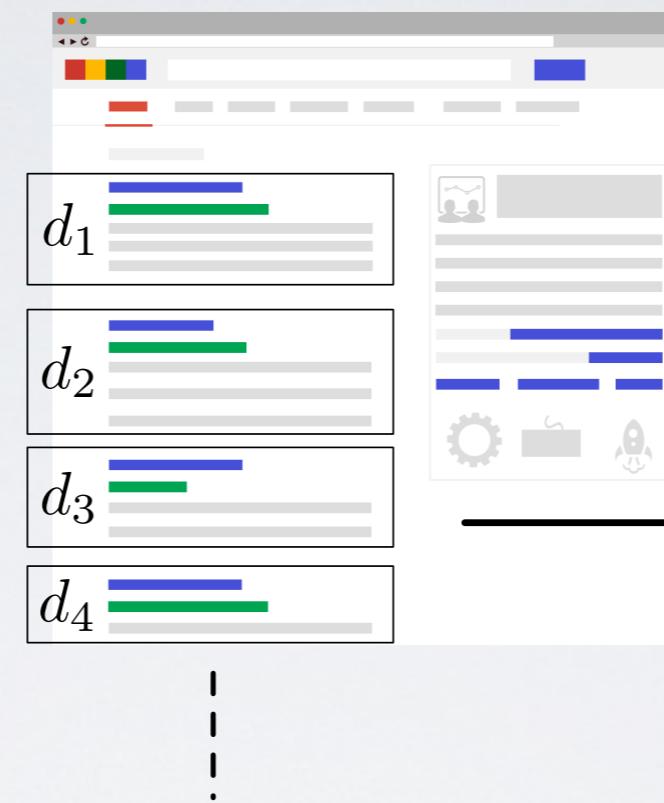


Topic

Run

Assessed Run

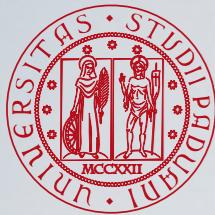
Weighted  
Assessed Run



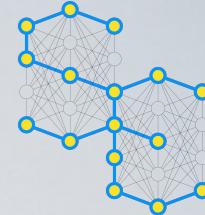
|    |                    |
|----|--------------------|
| 1  | Highly Relevant    |
| 2  | Not Relevant       |
| 3  | Partially Relevant |
| 4  | Fairly Relevant    |
| 5  | Not Relevant       |
| 6  | Not Relevant       |
| 7  | Not Relevant       |
| 8  | Fairly Relevant    |
| 9  | Not Relevant       |
| 10 | Not Relevant       |

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

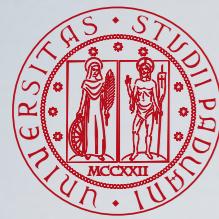
$$DCG = 3 + \frac{1}{\log_2(3)} + \frac{2}{\log_2(4)} + \frac{2}{\log_2(8)} = 5.2976$$



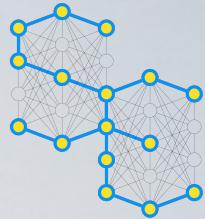
# Compute it Yourself!



```
% A topic of a run.  
% Each position is a retrieved document; first element of the vector is  
% the top of the rank.  
% Assume graded relevance: 0 for not relevant; 1 for partially relevant;  
% 2 for fairly relevant; 3 for highly relevant  
run = [3 0 1 2 0 0 0 2 0 0];  
  
% the base of the logarithm  
b = 2;  
  
% DCG at each rank position  
if (b == 2) % impatient user  
    dcg = cumsum(run ./ max(1, log2(1:length(run))));  
else % patient user  
    dcg = cumsum(run ./ max(1, log10(1:length(run))));  
end
```

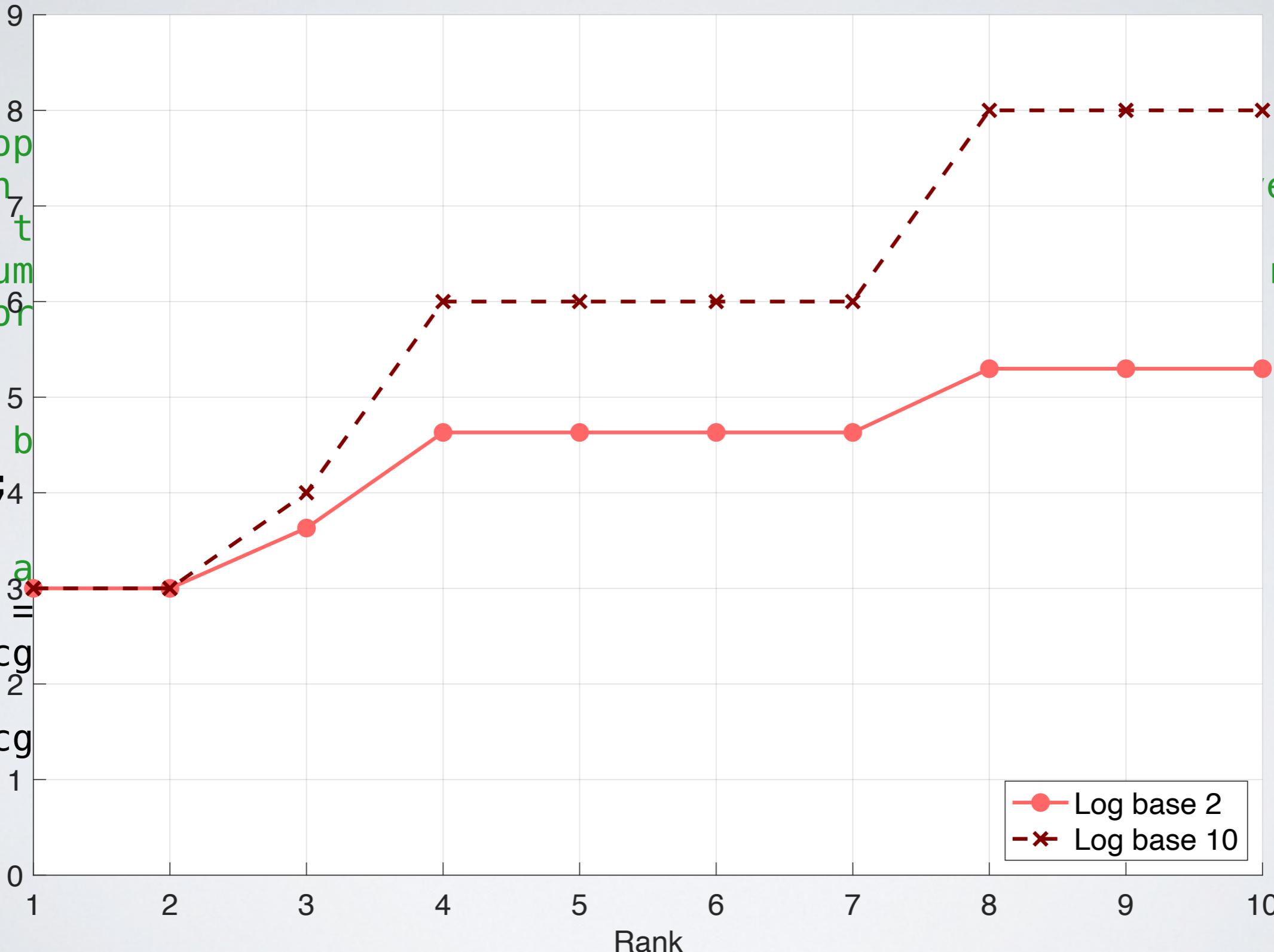


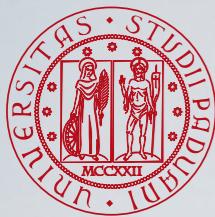
# Compute it Yourself!



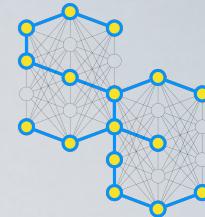
```
% A top  
% Each  
% the t  
% Assum  
% 2 for  
run =  
% the b  
b = 2;4  
% DCG  
if (b =  
dcg  
else  
dcg  
end
```

vector is  
relevant;





## Rank-based Measures: Normalized Discounted Cumulated Gain

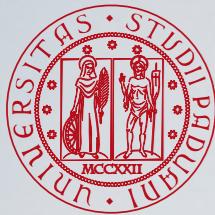


- To normalize DCG in  $[0, 1]$ , you need to compute the ideal run, i.e. the pool sorted in descending order of relevance, which represents the best retrieval possible and the maximum value of DCG

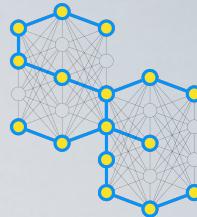
$$nDCG(k) = \frac{DCG(k)}{iDCG(k)}$$

info da gradi

- nDCG is given by the DCG of the run divided by the DCG of the ideal run



# Rank-based Measures: Example of Normalized Discounted Cumulated Gain



Topic

Run

Assessed Run

Weighted  
Assessed Run

Weighted  
Assessed  
Ideal Run



Assume

- $RB = 8$  relevant documents in total
- An impatient user

$$DCG = 5.2976$$

$$iDCG = 10.1996$$

$$nDCG = 0.5194$$

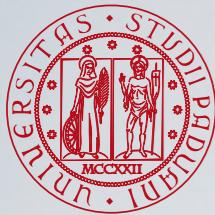


|    |                    |
|----|--------------------|
| 1  | Highly Relevant    |
| 2  | Not Relevant       |
| 3  | Partially Relevant |
| 4  | Fairly Relevant    |
| 5  | Not Relevant       |
| 6  | Not Relevant       |
| 7  | Not Relevant       |
| 8  | Fairly Relevant    |
| 9  | Not Relevant       |
| 10 | Not Relevant       |

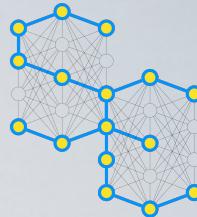
→

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

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



# Computing nDCG Using trec\_eval



```
[ferro~trec_eval.9.0]$ ./trec_eval -q -m all_trec ../data/CLEF2009-Persian/pool/AH-PERSIAN-CLEF2009.txt ../data/CLEF2009-Persian/runs/UNINEPE1.txt
```

|                      |        |        |
|----------------------|--------|--------|
| num_ret              | 601-AH | 1000   |
| num_rel              | 601-AH | 89     |
| num_rel_ret          | 601-AH | 67     |
| map                  | 601-AH | 0.1856 |
| Rprec                | 601-AH | 0.2360 |
| bpref                | 601-AH | 0.1788 |
| recip_rank           | 601-AH | 1.0000 |
| iprec_at_recall_0.00 | 601-AH | 1.0000 |
| iprec_at_recall_0.10 | 601-AH | 0.4737 |
| iprec_at_recall_0.20 | 601-AH | 0.2500 |

```
[ferro~trec_eval.9.0]$ ./trec_eval -q -m all_trec ../data/CLEF2009-Persian/pool/AH-PERSIAN-CLEF2009.txt ../data/CLEF2009-Persian/runs/UNINEPE1.txt
```

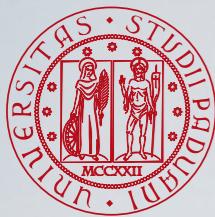
|               |               |               |
|---------------|---------------|---------------|
| G             | 601-AH        | 0.1405        |
| <b>ndcg</b>   | <b>601-AH</b> | <b>0.5920</b> |
| ndcg_rel      | 601-AH        | 0.4902        |
| Rndcg         | 601-AH        | 0.4411        |
| ndcg_cut_5    | 601-AH        | 0.5531        |
| ndcg_cut_10   | 601-AH        | 0.4323        |
| ndcg_cut_15   | 601-AH        | 0.4697        |
| ndcg_cut_20   | 601-AH        | 0.4927        |
| ndcg_cut_30   | 601-AH        | 0.4246        |
| ndcg_cut_100  | 601-AH        | 0.3061        |
| ndcg_cut_200  | 601-AH        | 0.4219        |
| ndcg_cut_500  | 601-AH        | 0.5536        |
| ndcg_cut_1000 | 601-AH        | 0.5920        |
| map_cut_5     | 601-AH        | 0.0225        |

trec\_eval computes only nDCG (and not also DCG) and it uses  $b = 2$  as base of the logarithm

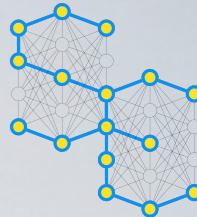
trec\_eval uses the values specified in the pool file as nDCG weights

- To compute all the evaluation measures run

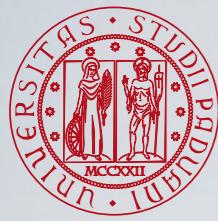
```
trec_eval -q -m all_trec pool.txt run.txt
```



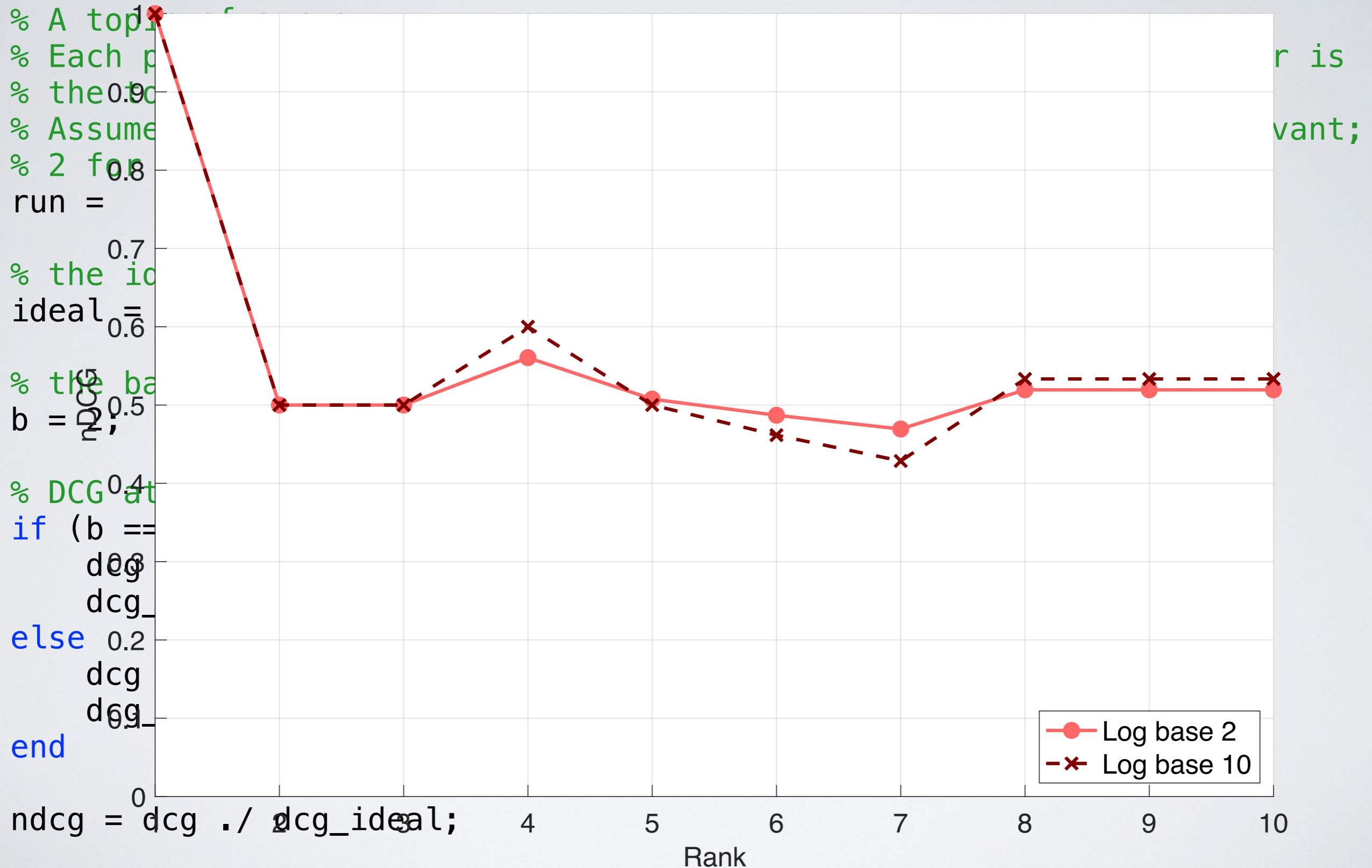
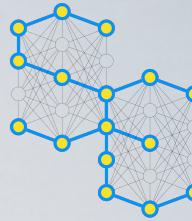
# Compute it Yourself!

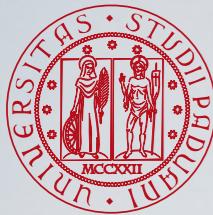


```
% A topic of a run.  
% Each position is a retrieved document; first element of the vector is  
% the top of the rank.  
% Assume graded relevance: 0 for not relevant; 1 for partially relevant;  
% 2 for fairly relevant; 3 for highly relevant  
run = [3 0 1 2 0 0 0 2 0 0];  
  
% the ideal run  
ideal = [3 3 2 2 2 1 1 1 0 0];  
  
% the base of the logarithm  
b = 2;  
  
% DCG at each rank position  
if (b == 2)      % impatient user  
    dcg = cumsum(run ./ max(1, log2(1:length(run))));  
    dcg_ideal = cumsum(ideal ./ max(1, log2(1:length(ideal))));  
else              % patient user  
    dcg = cumsum(run ./ max(1, log10(1:length(run))));  
    dcg_ideal = cumsum(ideal ./ max(1, log10(1:length(ideal))));  
end  
  
ndcg = dcg ./ dcg_ideal;
```

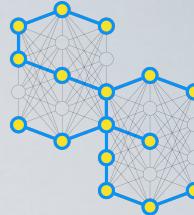


# Compute it Yourself!





# Rank-based Measures: User Models

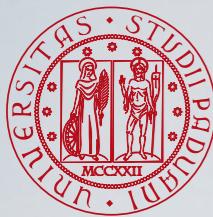


- Rank-based evaluation measures, implicitly or explicitly, embed a **user model** comprising
  - a **browsing model** that describes how a user interacts with results;
  - a **model of document utility**, describing how a user derives utility from individual relevant documents;
  - a **utility accumulation model** that describes how a user accumulates utility in the course of browsing.
- User models may be more or less **artificial** and may be more or less **correlated** with **actual user behaviour** and preferences
- In the case of **DCG**
  - a **browsing model**: user steps down the ranked results one-by-one, until s/he reaches the stopping rank  $k$  which is picked with a probability proportional to the log of the rank
  - a **model of document utility**: user gains something from each relevant document, proportional to its relevance degree
  - a **utility accumulation model**: user gains from all of the relevant documents from ranks 1 through  $k$

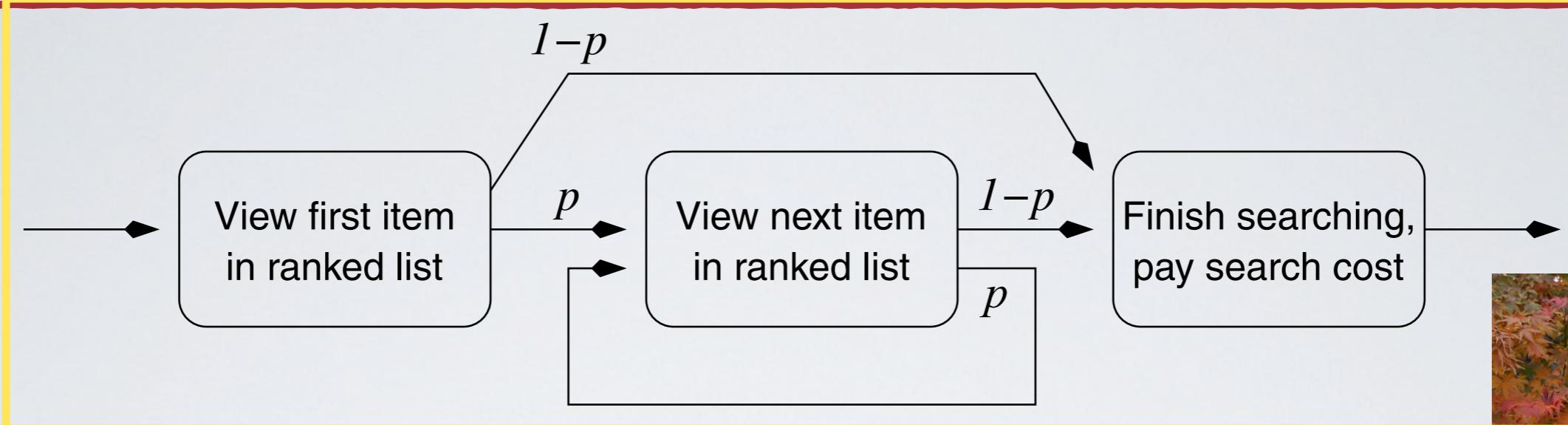
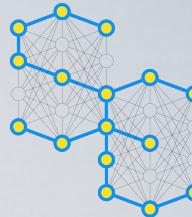


Ben Carterette

Carterette, B. A. (2011). System Effectiveness, User Models, and User Utility: A Conceptual Framework for Investigation. In Ma, W.-Y., Nie, J.-Y., Baeza-Yautes, R., Chua, T.-S., and Croft, W. B., editors, *Proc. 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011)*, pages 903–912. ACM Press, New York, USA.



# Rank-based Measures: Rank-Biased Precision



- The user starts from the top ranked document and with probability  $p$ , called **persistence**, goes to the next document or with probability  $1 - p$  stops
- typical value for  $p$  are: 0.5 for impatient users, 0.8 for patient users, and 0.95 for extremely patient users

$$RBP = (1 - p) \sum_{n=1}^N p^{n-1} r_n = (1 - p) \sum_{k \in R} p^{k-1}$$

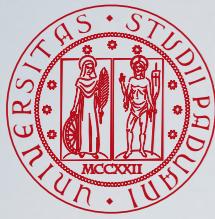


Alistair Moffat

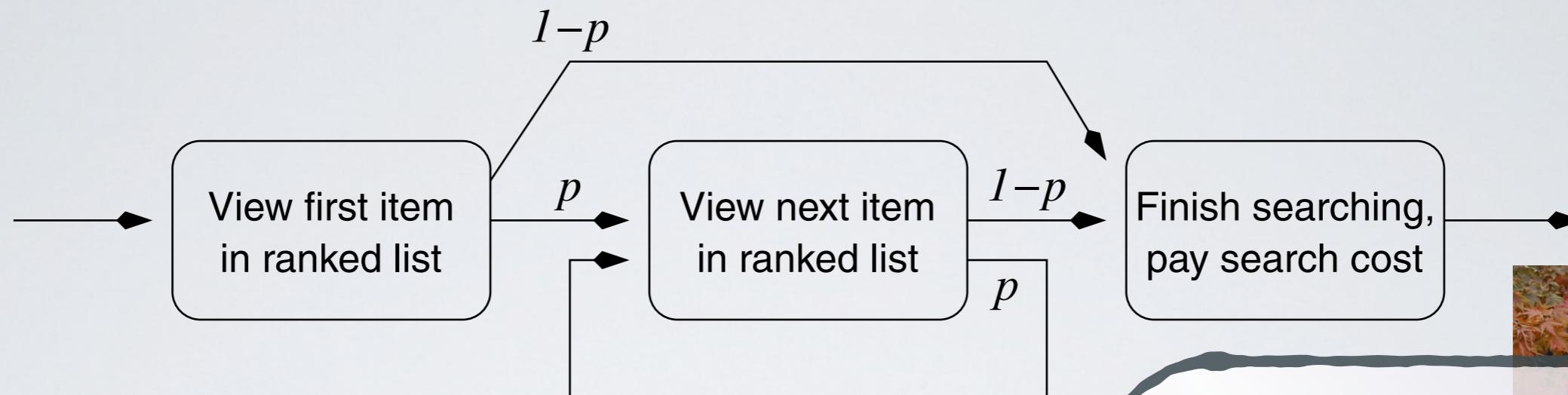
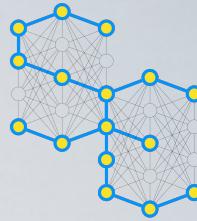


Justin Zobel

Moffat, A. and Zobel, J. (2008). Rank-biased Precision for Measurement of Retrieval Effectiveness. *ACM Transactions on Information Systems (TOIS)*, 27(1):2:1–2:27.



# Rank-based Measures: Rank-Biased Precision



- The user starts from the top ranked document and with probability  $p$ , called **persistence**, goes to the next document or with probability  $1 - p$  stops
- typical value for  $p$  are: 0.5 for impatient users, 0.8 for patient users, and 0.95 for extremely patient users

$$RBP = (1 - p) \sum_{n=1}^N p^{n-1} r_n = (1 - p) \sum_{k \in \mathcal{R}} p^{k-1}$$

RBP is independent from the recall base



Alistair Moffat

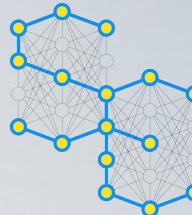


Justin Zobel

Moffat, A. and Zobel, J. (2008). Rank-biased Precision for Measurement of Retrieval Effectiveness. *ACM Transactions on Information Systems (TOIS)*, 27(1):2:1–2:27.

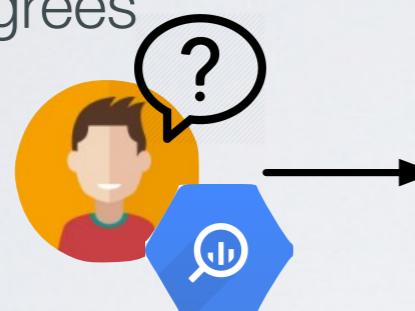


## Rank-based Measures: Example of Rank-biased Precision



Topic

- $p = 0.8$  a patient user
  - Lenient mapping to binary relevance degrees



$$RBP = (1 - 0.8) \left( 0.8^{1-1} + 0.8^{3-1} + 0.8^{4-1} + 0.8^{8-1} \right) = \\ = 0.4723$$

The diagram illustrates the process of generating a binary weighted run from a run and an assessed run.

**Run**

A screenshot of a user interface showing four documents ( $d_1, d_2, d_3, d_4$ ) with colored bars (blue, green, grey) indicating relevance levels. A vertical ellipsis indicates more documents.

**Assessed Run**

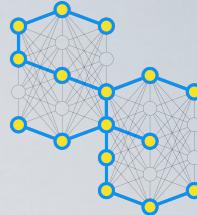
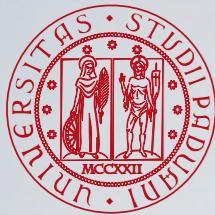
A table mapping document IDs to relevance levels:

| Document ID | Relevance Level    |
|-------------|--------------------|
| 1           | Highly Relevant    |
| 2           | Not Relevant       |
| 3           | Partially Relevant |
| 4           | Fairly Relevant    |
| 5           | Not Relevant       |
| 6           | Not Relevant       |
| 7           | Not Relevant       |
| 8           | Fairly Relevant    |
| 9           | Not Relevant       |
| 10          | Not Relevant       |

**Binary Weighted Assessed Run**

A table showing the binary weighted run, generated by applying the relevance levels from the assessed run to the corresponding documents in the run:

| Document ID | Binary Weight |
|-------------|---------------|
| 1           | 1             |
| 2           | 0             |
| 3           | 1             |
| 4           | 1             |
| 5           | 0             |
| 6           | 0             |
| 7           | 0             |
| 8           | 1             |
| 9           | 0             |
| 10          | 0             |



# Compute it Yourself!

```
% A topic of a run.  
% Each position is a retrieved document; first element of the vector is  
% the top of the rank.  
% Assume binary relevance: 0 for not relevant; 1 for relevant.  
run = [0 1 1 0 1 0 0 1 1 0];  
  
% the persistence  
p = 0.8;  
  
% Find the rank positions of the relevant retrieved documents  
ranks = find(run);  
  
% compute rank-biased precision, if any relevant document has been retrieved  
if ~isempty(ranks)  
    rbp = (1 - p) .* sum(repmat(p, 1, length(ranks)) .^ (ranks - 1));  
else  
    rbp = 0;  
end
```

# questions?

It works...It  
doesn't....It  
works.....

Just a hunch....maybe we do  
need a better way to  
measure results....

