

Big Data and Data Mining

Information Retrieval Evaluation

Flavio Bertini

flavio.bertini@unipr.it

Measuring Relevance

- Methods pioneered by Cyril Cleverdon (a British librarian and computer scientist) in the *Cranfield Experiments* in the 1960s
- Three elements:
 1. A benchmark **document collection**
 2. A benchmark suite of **queries**
 3. A **human assessment** (relevance judgments) of either Relevant or Nonrelevant for each query and each document



Cyril Cleverdon

Assessments

- Note: **user need** is translated into a **query**
- Relevance is assessed relative to the **user need**, *not* the **query**, for example:
 - Information need: *My swimming pool bottom is becoming black and needs to be cleaned*
 - Query: ***pool cleaner***
- Assess whether the doc addresses the underlying need, not whether it has these words

Relevance judgments

- Binary (relevant vs. non-relevant) in the simplest case, or more precisely (0, 1, 2, 3 ...) in others
- If, for each query, we consider all the set of documents to be judged, the relevance assessment can be huge and expensive
- The depth-**k** pooling solution:
 - Take in consideration the top-**k** (e.g. 100) documents of **N** (e.g. 100) different information retrieval systems
 - Humans must judge a “pool” of no more than **k x N** documents (e.g. 10'000), which is far less than the entire document collection (could be millions of documents)



Qualified Test Collections

<i>Collection</i>	<i>NDocs</i>	<i>NQrys</i>	<i>Size (MB)</i>	<i>Term/Doc</i>	<i>Q-D RelAss</i>
ADI	82	35			
AIT	2109	14	2	400	>10,000
CACM	3204	64	2	24.5	
CISI	1460	112	2	46.5	
Cranfield	1400	225	2	53.1	
LISA	5872	35	3		
Medline	1033	30	1		
NPL	11,429	93	3		
OSHMED	34,8566	106	400	250	16,140
Reuters	21,578	672	28	131	
GOV2	25 Mln	10,000	426,000	956	1,000

Typical
TREC



TREC Collections

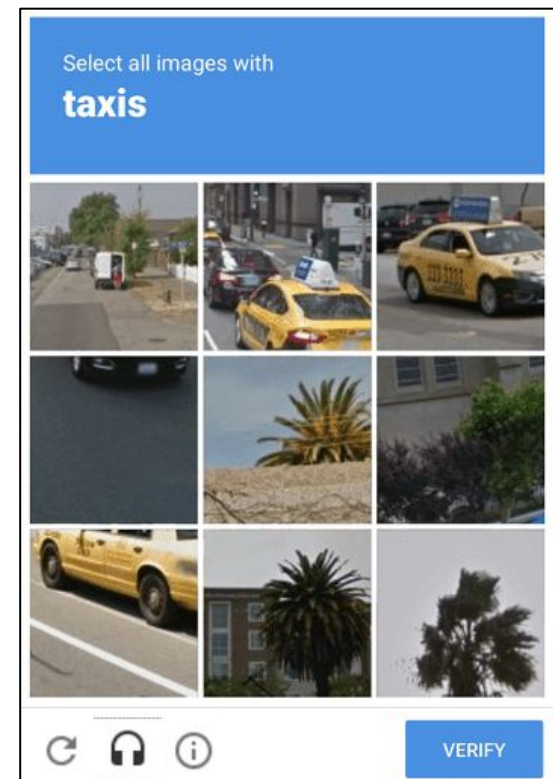
Text REtrieval Conference (TREC)

... to encourage research in information retrieval
from large text collections.

- The U.S. *National Institute of Standards and Technology* (NIST) has run a large IR test bed evaluation series since 1992. Within this framework, there have been **many tracks** over a range of **different test collections**
- TREC GOV2 is now the largest Web collection easily available for research purposes, including 25 million pages

Mechanical Turk

- Present query-document pairs to low-cost labor on online crowd-sourcing platforms
 - Hope that this is cheaper than hiring qualified assessors
- [Amazon Mechanical Turk](#)
- Lots of literature on using crowd-sourcing for such tasks
 - From Carnegie Mellon OCR to Google login captcha
- Main takeaway – you get some signal, but the variance in the resulting judgments is very high
 - MIT Technology Review, [Death to captchas](#)



Confusion matrix

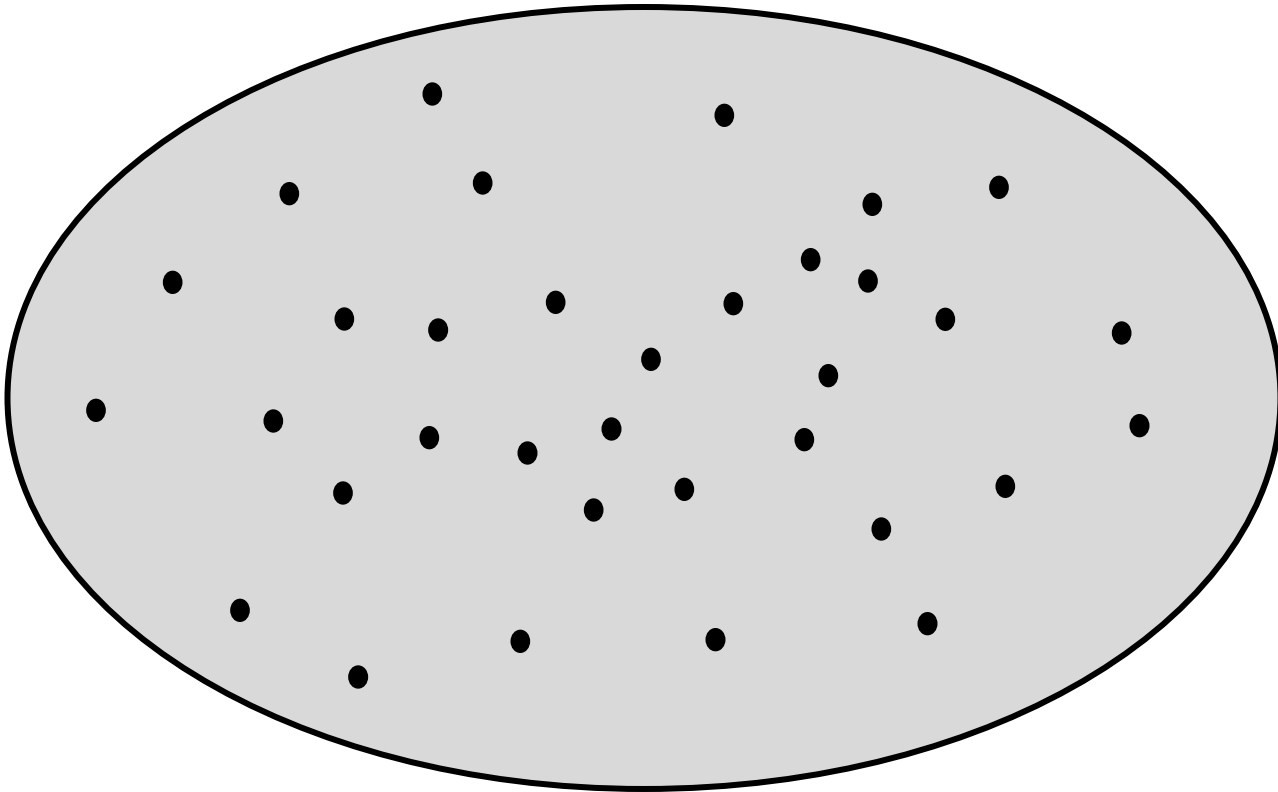
		True condition			
Total population		Condition positive	Condition negative	Prevalence = $\frac{\sum \text{Condition positive}}{\sum \text{Total population}}$	Accuracy (ACC) = $\frac{\sum \text{True positive} + \sum \text{True negative}}{\sum \text{Total population}}$
Predicted condition	Predicted condition positive	True positive	False positive, Type I error	Positive predictive value (PPV), Precision = $\frac{\sum \text{True positive}}{\sum \text{Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\sum \text{False positive}}{\sum \text{Predicted condition positive}}$
	Predicted condition negative	False negative, Type II error	True negative	False omission rate (FOR) = $\frac{\sum \text{False negative}}{\sum \text{Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\sum \text{True negative}}{\sum \text{Predicted condition negative}}$
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power = $\frac{\sum \text{True positive}}{\sum \text{Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\sum \text{False positive}}{\sum \text{Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR-}}$ F₁ score = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
		False negative rate (FNR), Miss rate = $\frac{\sum \text{False negative}}{\sum \text{Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\sum \text{True negative}}{\sum \text{Condition negative}}$	Negative likelihood ratio (LR-) = $\frac{\text{FNR}}{\text{TNR}}$	

Effectiveness measures

- To assess the *effectiveness* of an IR system (the quality of its search results), there are two parameters about the system's returned results for a query:
 - **Precision**: What fraction of the **returned** documents are relevant to the information need?
 - **Recall**: What fraction of the **relevant** documents in the collection were returned by the system?

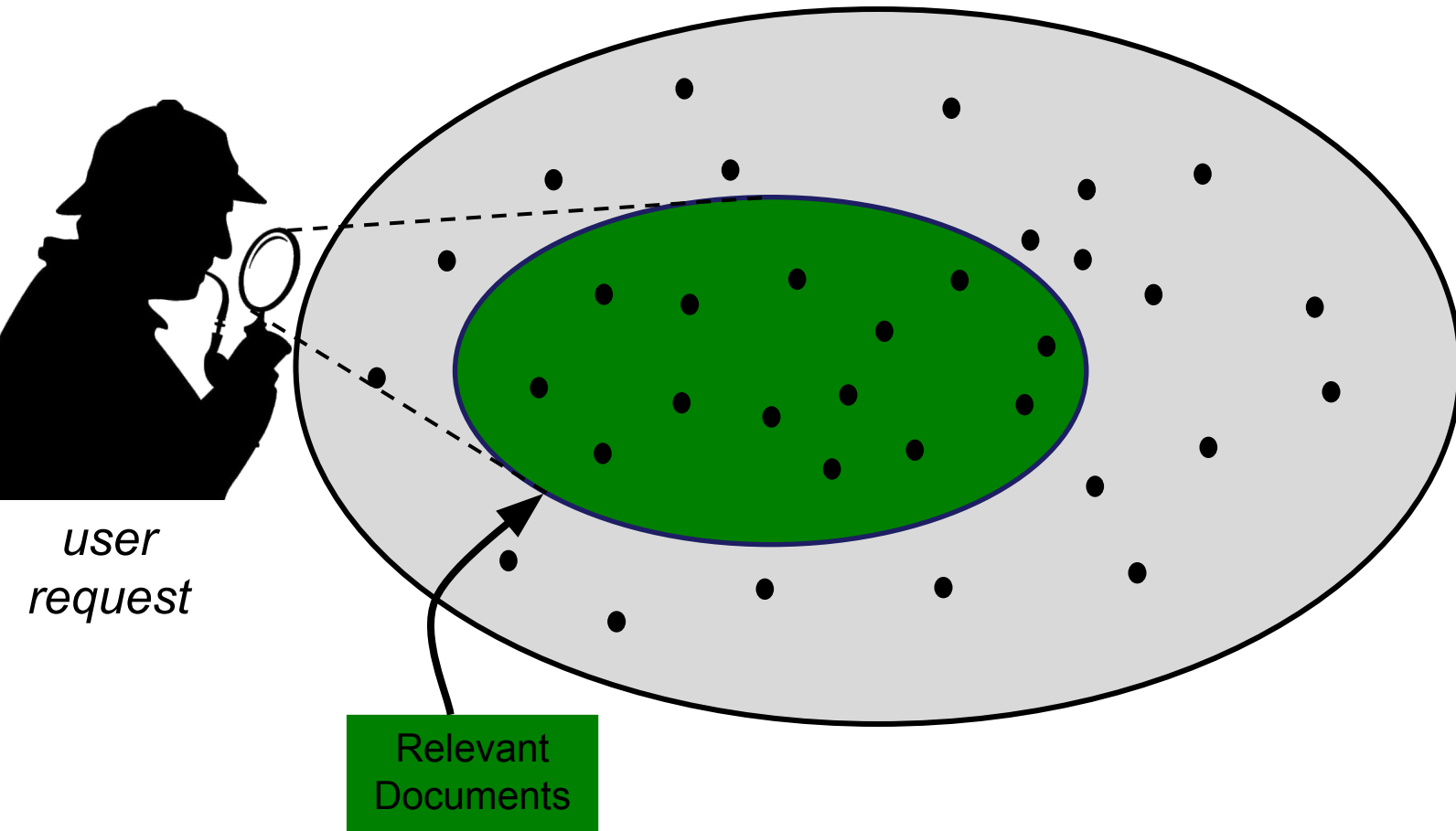
Collection of documents

Each dot • is a document of the collection



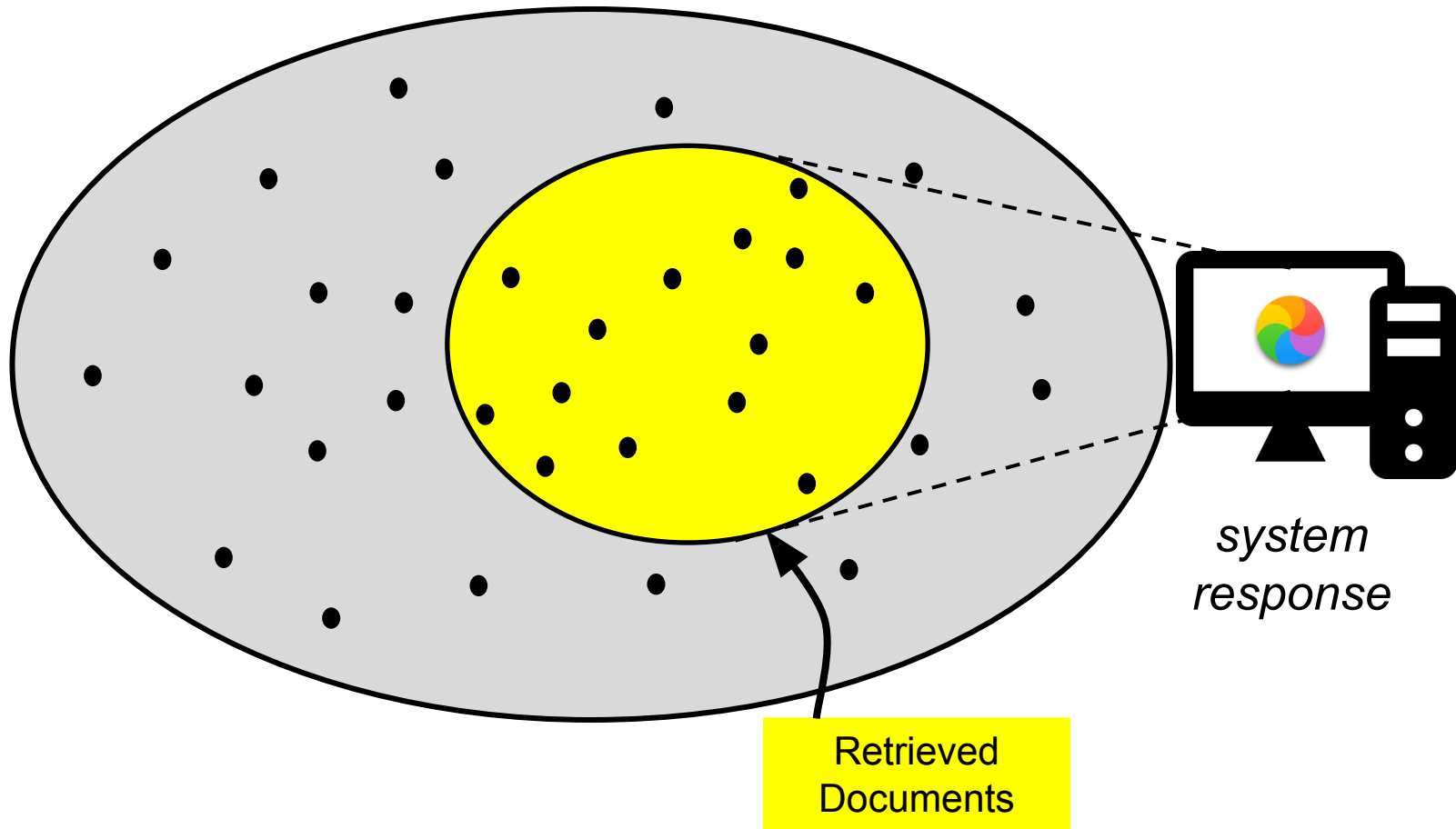
Relevant documents

Given a query, the relevant documents is the set of all the documents **really relevant** to the query

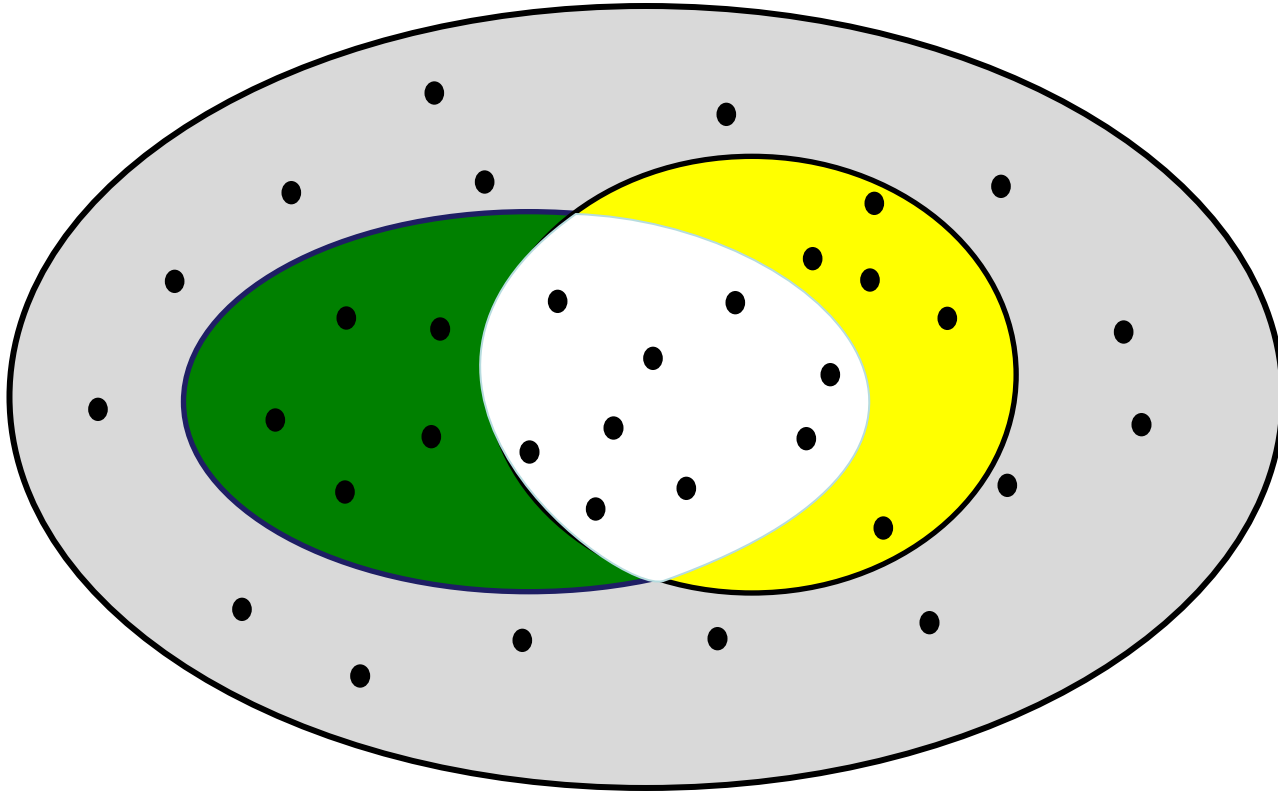


Retrieved documents

Given the same query, the Retrieved documents is the set of all the documents **returned by the system** we want to **evaluate**



Relevant retrieved documents



Relevant Retrieved
Documents

=

Relevant
Documents

\cap

Retrieved
Documents

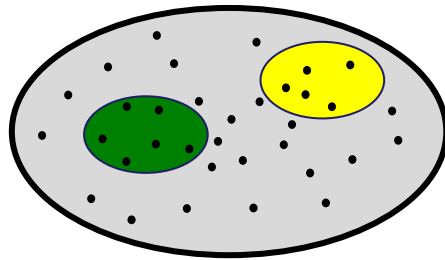
Precision

$$\text{Precision} = \frac{\text{Relevant Retrieved Documents}}{\text{Retrieved Documents}}$$

The accuracy of positive predictions.

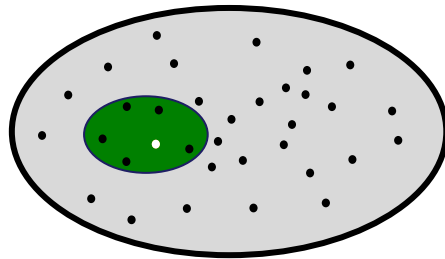
Range of values $[0, 1]$

- Relevant Retrieved Documents = \emptyset relevant documents \Rightarrow Precision = 0



The worst IR system

- Retrieved Documents = One relevant document \Rightarrow Precision = 1



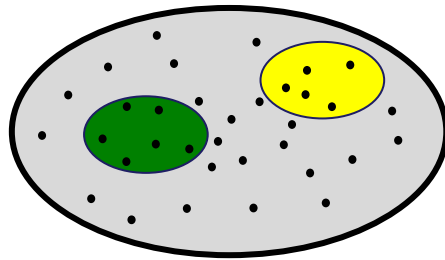
The miraculous IR system

Recall

$$\text{Recall} = \frac{\text{Relevant Retrieved Documents}}{\text{Relevant Documents}}$$

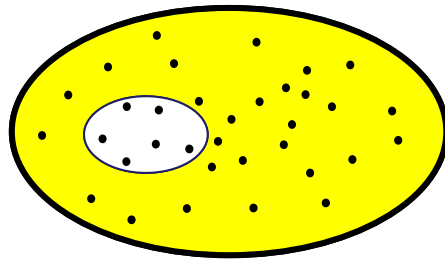
*The proportion of positives that are correctly identified.
Range of values $[0, 1]$*

- Relevant Retrieved Documents = \emptyset relevant documents \Rightarrow Recall = 0



The worst IR system

- Retrieved Documents = All documents \Rightarrow Recall = 1

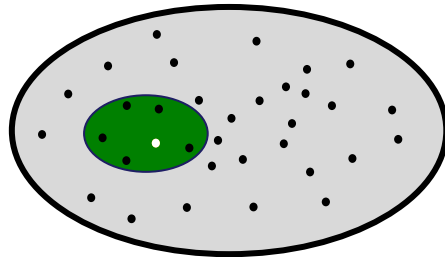


The naive IR system

Precision Recall Extremes

- **The miraculous IR system**

Relevant Retrieved Documents = One relevant document



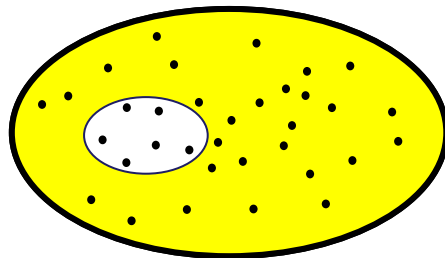
Precision = 1

Recall is very low!

Just one relevant document

- **The naive IR system**

Retrieved Documents = All the documents



Recall = 1

Precision is very low!

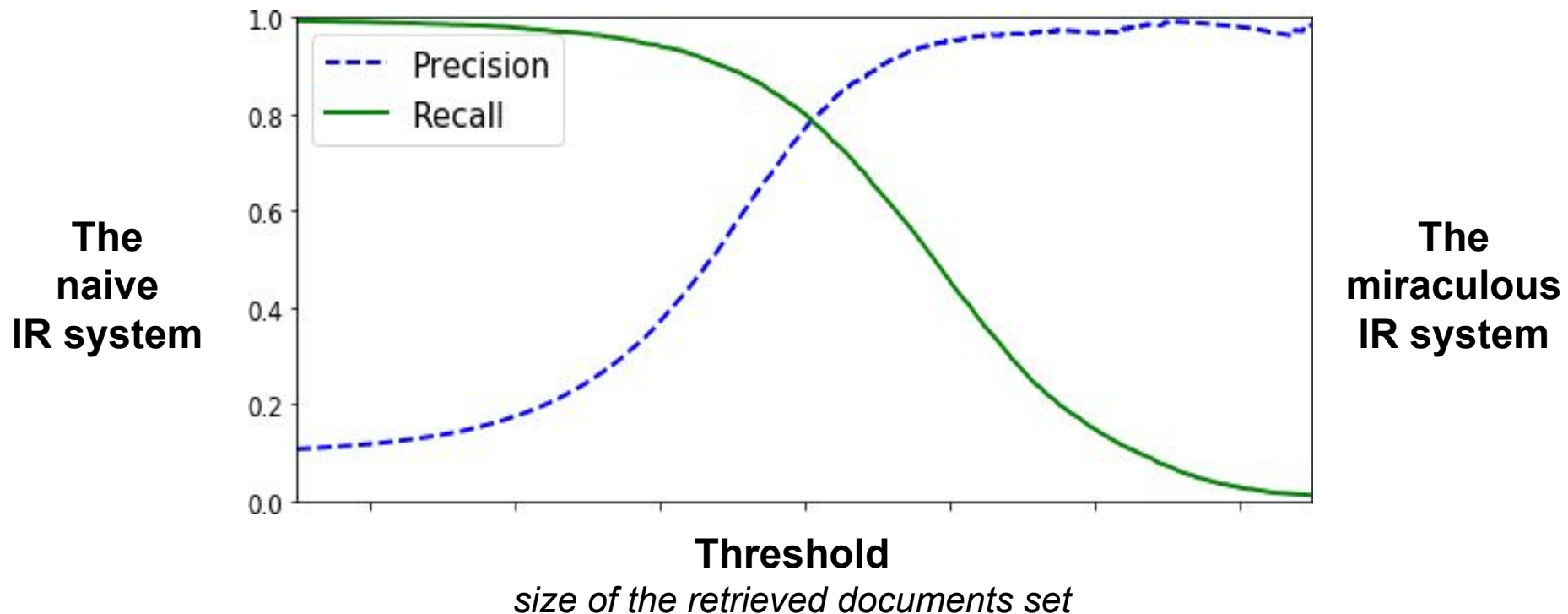
A lot of irrelevant documents



Precision Recall Tradeoff

In an ideal scenario where there is a perfectly separable data, both **precision and recall can get maximum value of 1**

Unfortunately, **in most of the practical situations**, you can't have both precision and recall high. **If you increase precision, it will reduce recall, and vice versa**



F-Measure

Alone, neither precision or recall tells the whole story. We can have excellent precision with terrible recall, or alternately, terrible precision with excellent recall

F-Measure combines both into a single measure that **provides a overall evaluation of the accuracy of the IR system.** The traditional F-Measure is calculated as follows:

$$\text{F-Measure} = \frac{(2 * \text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

This is the **harmonic mean a.k.a. F-Score or the F1-Score.** The more generic F_β score applies additional weights (β), valuing one of precision or recall more than the other






Evaluation of Ranked Results

- **Precision, recall, and the F-measure** are set-based measures **for unordered sets of documents**
- **We need to** extend these measures if we are to **evaluate the ranked retrieval results** that are now standard with **search engines**, where what matters is how many **good results there are on the first page**
- In a ranked retrieval context, appropriate sets of retrieved documents are naturally given by the **top k retrieved documents**
- This leads to measuring precision and recall at a fixed low level of retrieved results, that is the k documents. This is referred to as **Precision @ K** and **Recall @ K** (a.k.a. **P@K** and **R@K**)

Precision @ K

- Set a rank threshold K: the number of retrieved documents
- Compute the proportion of the top k returned documents that are relevant
- Ignores documents ranked lower than K

Ranked results list

-  #1 is relevant
-  #2 is not relevant
-  #3 is relevant
-  #4 is not relevant
-  #5 is relevant

Precision* at different K

$$P@1: 1/1 = 1.00$$

$$P@2: 1/2 = 0.50$$

$$P@3: 2/3 = 0.67$$

$$P@4: 2/4 \dots$$






...

**Precision = Relevant Retrieved / Retrieved*

Recall @ K

- Set a rank threshold K: the number of retrieved documents
- Compute the proportion of relevant items found in the top k returned documents
- Ignores documents ranked lower than K

Ranked results list

-  #1 is relevant
-  #2 is not relevant
-  #3 is relevant
-  #4 is not relevant
-  #5 is relevant

Recall* at different K

$$R@1: 1/3 = 0.33$$

$$R@2: 1/3 = 0.33$$

$$R@3: 2/3 = 0.67$$

$$R@4: 2/3 \dots$$

...

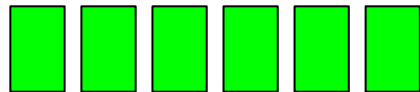
**Recall = Relevant Retrieved / Relevant*













Average Precision (AP)

- As for the F-Measure, the need to use an aggregate value is even stronger with a variable value of K
- **Average Precision** is an **aggregated measure** for ranked results
- It is computed as follows:
 - Instead of setting an arbitrary K , we **stop only when all the relevant documents** are retrieved: that value for K for which **Recall @ K is equal to 1**
 - We **compute the Precisions @ K only for those K where relevant result is retrieved**
 - The **average** of this precision measures is the **Average Precision (AP)**

AP example











 = the relevant documents

IR system 1 \Rightarrow Ranking #1

										
Recall@k	0.17	0.17	0.33	0.50	0.67	0.83	0.83	0.83	0.83	1.00
Precision@k	1.00	0.50	0.67	0.75	0.80	0.83	0.71	0.63	0.56	0.60

We stop
when
Recall@k=1

IR system 2 \Rightarrow Ranking #2

										
Recall@k	0.00	0.17	0.17	0.17	0.33	0.50	0.67	0.67	0.83	1.00
Precision@k	0.00	0.50	0.33	0.25	0.40	0.50	0.57	0.50	0.56	0.60

We stop
when
Recall@k=1

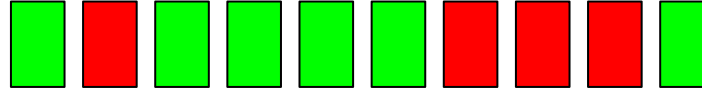
Compute AP using P@K only for those K where relevant result is retrieved

Ranking #1: $(1.00 + 0.67 + 0.75 + 0.80 + 0.83 + 0.60)/6 = 0.78$

Ranking #2: $(0.50 + 0.40 + 0.50 + 0.57 + 0.56 + 0.60)/6 = 0.52$

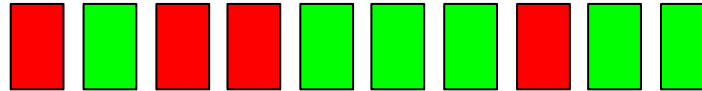
AP observations

IR system 1 \Rightarrow Ranking #1



1.00 = Recall @ 10
0.60 = Precision @ 10

IR system 2 \Rightarrow Ranking #2



1.00 = Recall @ 10
0.60 = Precision @ 10

- Recall @ 10 and Precision @ 10 is equal for the two rankings
- However, AP is able to capture that Ranking #1 is better, as it ranks **more relevant documents in higher positions**
 - AP IR system 1 = 0.78
 - AP IR system 2 = 0.52

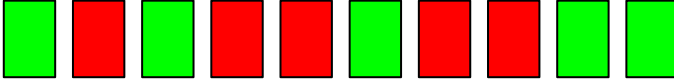


Mean Average Precision (MAP)

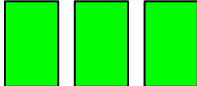
- When evaluating a system we usually measure the effectiveness over **more than one query** (10,000 in GOV2 TREC collection)
- The **number of queries** is another **dimension of aggregation** of the measures to evaluate the IR system
- After computing the Average Precision of each query in the test collection, the **Mean Average Precision (MAP)** is the average of the Average Precision over all the queries

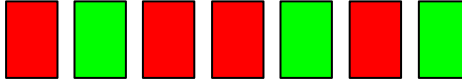
MAP example

 = relevant documents for **Query 1**

Ranking #1 

Recall@k	0.20	0.20	0.40	0.40	0.40	0.60	0.60	0.60	0.80	1.00
Precision@k	1.00	0.50	0.67	0.50	0.40	0.50	0.43	0.38	0.44	0.50

 = relevant documents for **Query 2**

Ranking #2 

Recall@k	0.00	0.33	0.33	0.33	0.67	0.67	1.00
Precision@k	0.00	0.50	0.33	0.25	0.40	0.33	0.43

Compute MAP as the average of the Average Precision over all the queries

AP query 1: $(1.00 + 0.67 + 0.50 + 0.44 + 0.50)/5 = 0.62$

AP query 2: $(0.50 + 0.40 + 0.43)/3 = 0.44$

MAP: $(0.62 + 0.44)/2 = 0.53$

MAP observations

- MAP assumes user is **interested in finding many relevant documents for each query**
 - If a relevant document never gets retrieved, we assume the precision corresponding to that relevant doc to be zero
- MAP is macro-averaging: **each query counts equally**
- There is normally **more agreement in MAP for an individual information need across systems** than for MAP scores for different information needs for the same system

Beyond binary relevance

- We assumed a **binary notion of relevance**:
 - either a document is relevant to the query or
 - it is non relevant to the query
- Some documents can be **less relevant** than others, but still relevant (non binary notion)
 - Specific measure with non-binary assessments: **DCG** (Discounted Cumulative Gain) or **NDCG** (Normalized Discounted Cumulative Gain)
- Binary relevance is still more common and provide a good estimation for IR evaluation

References

Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze
Introduction to Information Retrieval
Cambridge University Press. 2008

The book is also online for free:

- HTML edition (2009.04.07)
- PDF of the book for online viewing
(with nice hyperlink features,
2009.04.01)
- PDF of the book for printing
(2009.04.01)

