“ALEXANDRU IOAN CUZA” UNIVERSITATY OF IAŞI
FACULTY OF COMPUTER SCIENCE

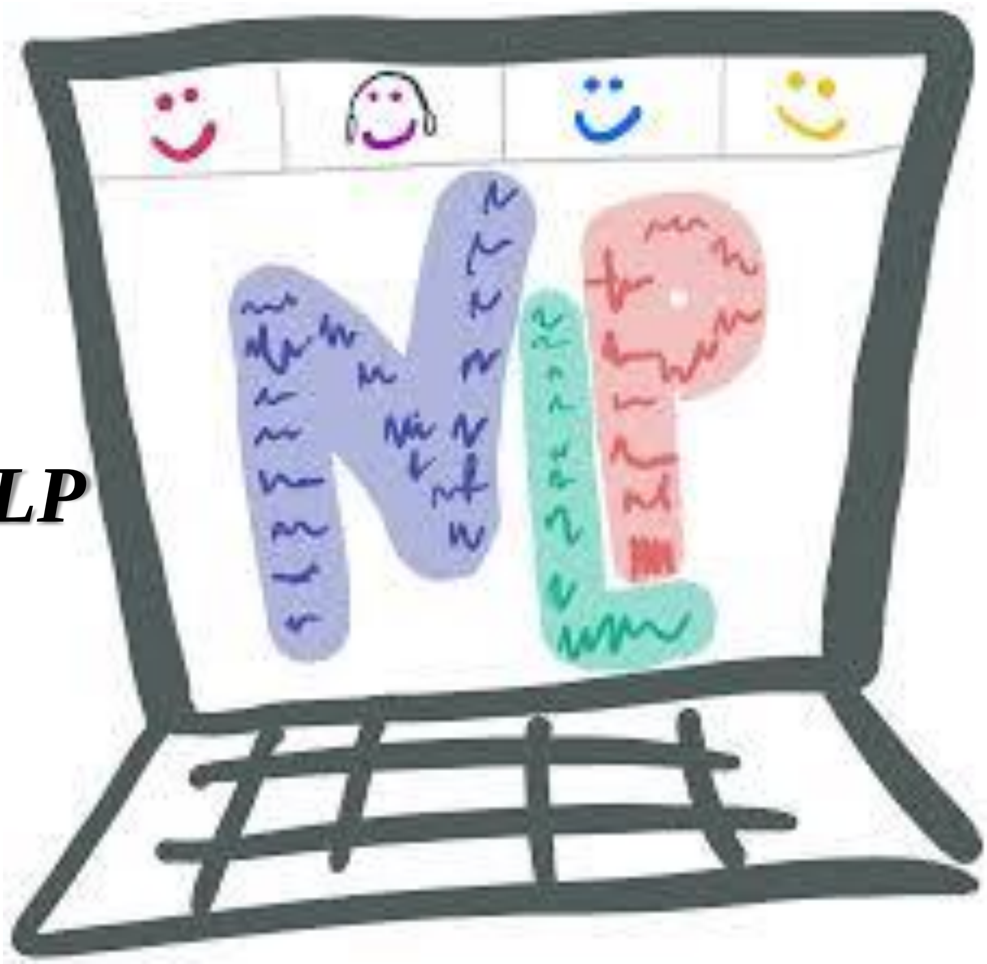
The Semantics and Pragmatics of Natural Language

Daniela GÎFU

http://profs.info.uaic.ro/~daniela.gifu/

Course 10

Evaluation Metrics in NLP



Why Evaluation Metrics?

- ➢ to measure the goodness of our model…
- ➢ in a Machine Learning (ML) context we need the measure of a model's performance on new instances that weren’t a part of the training data;
- ➢ the success of a model depends on 2 key factors:

  - ✓ Whether the evaluation metric we have selected is the correct one for our problem.
  - ✓ If we are following the correct evaluation process.

Types of Evaluation Metrics

- depends on the type of NLP task;
- the stage the project is at also affects the evaluation metric used for it;

Ex: during the model building and deployment phase = metric 1
the production phase = metric 2

2 buckets for categorize evaluation metrics:

- Intrinsic Evaluation - focuses on intermediary objectives

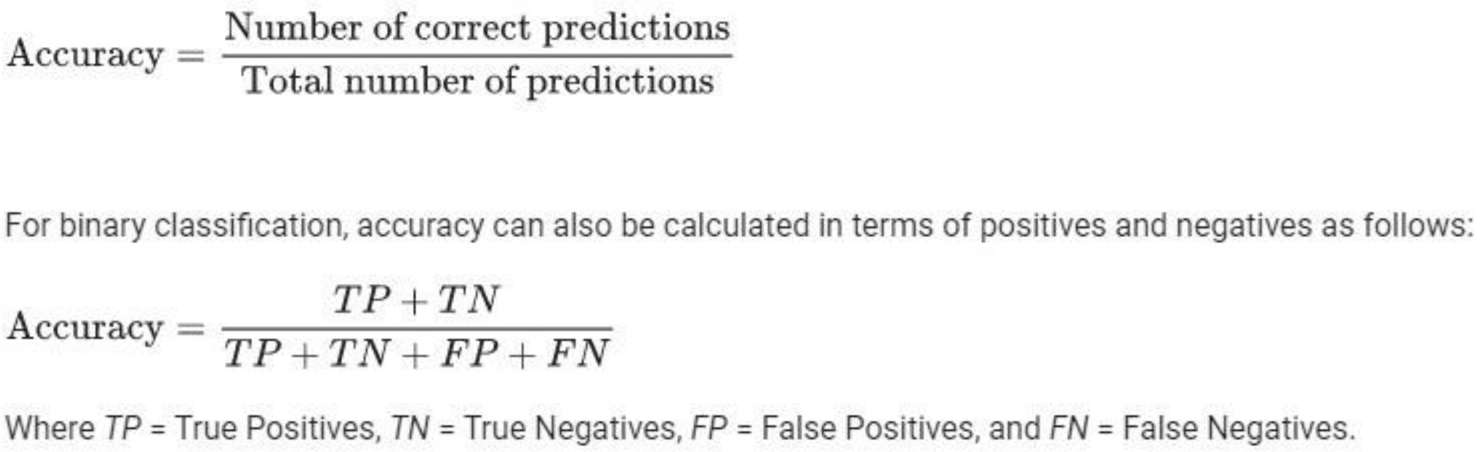
Ex: the performance of an NLP component on a defined subtask)

- Extrinsic Evaluation - focuses on the performance of the final objective

Ex: the performance of the component on the complete application)

Intrinsic Metrics to evaluate NLP systems_1

- Accuracy – a metric for evaluating classification models.

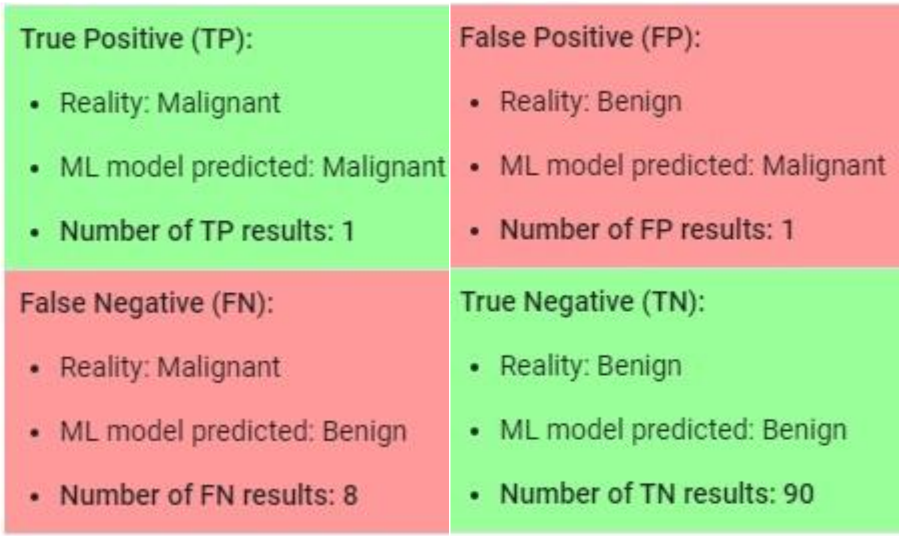


$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives.



True Positive (TP): • Reality: Malignant • ML model predicted: Malignant • Number of TP results: 1	False Positive (FP): • Reality: Benign • ML model predicted: Malignant • Number of FP results: 1
False Negative (FN): • Reality: Malignant • ML model predicted: Benign • Number of FN results: 8	True Negative (TN): • Reality: Benign • ML model predicted: Benign • Number of TN results: 90

Acc?

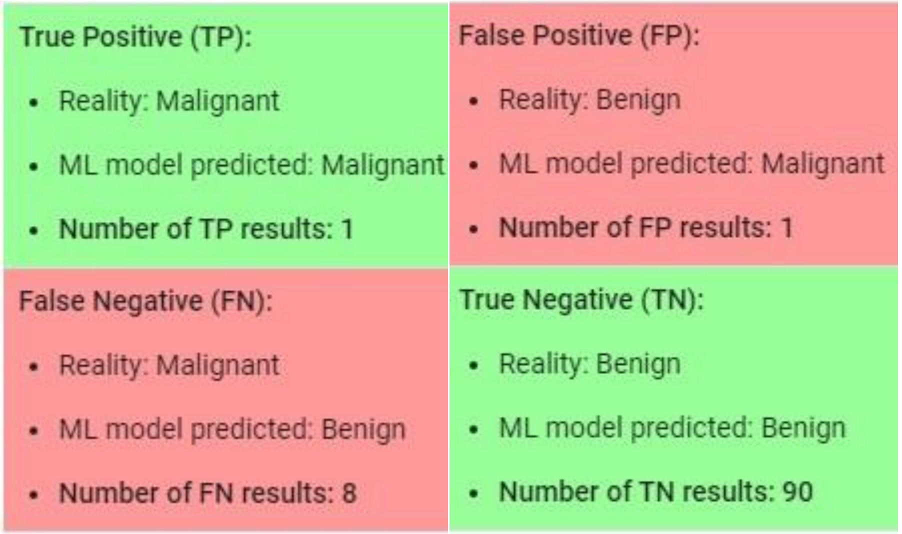
Intrinsic Metrics to evaluate NLP systems_2

- Precision – a metric for how exact the model's predictions are.

What proportion of positive identifications was actually correct?

$$\text{Precision} = \frac{TP}{TP+FP}$$

P?



True Positive (TP): • Reality: Malignant • ML model predicted: Malignant • **Number of TP results: 1**	False Positive (FP): • Reality: Benign • ML model predicted: Malignant • **Number of FP results: 1**
False Negative (FN): • Reality: Malignant • ML model predicted: Benign • **Number of FN results: 8**	True Negative (TN): • Reality: Benign • ML model predicted: Benign • **Number of TN results: 90**

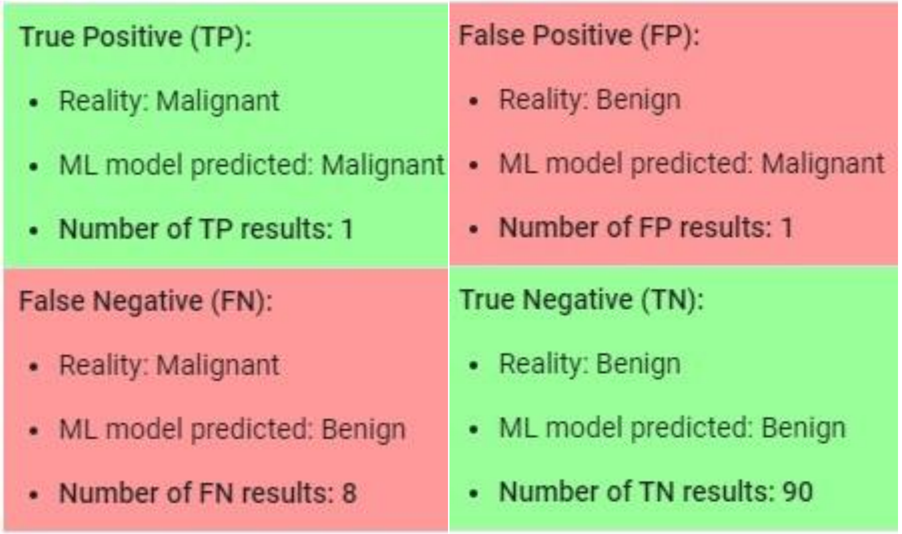
Intrinsic Metrics to evaluate NLP systems_3

- Recall – a metric for how well the model can recall the positive class.

What proportion of actual positives was identified correctly?

$$\text{Recall} = \frac{TP}{TP + FN}$$

R?



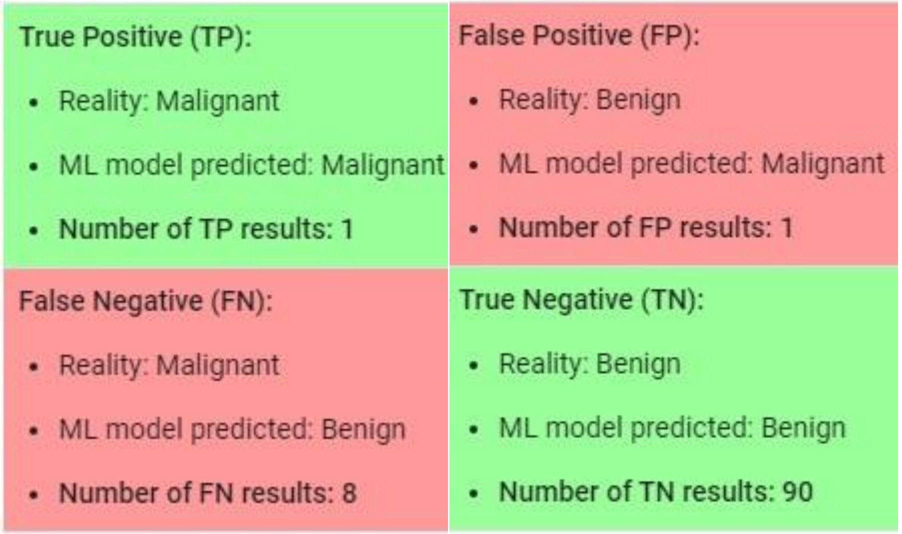
True Positive (TP): • Reality: Malignant • ML model predicted: Malignant • Number of TP results: 1	False Positive (FP): • Reality: Benign • ML model predicted: Malignant • Number of FP results: 1
False Negative (FN): • Reality: Malignant • ML model predicted: Benign • Number of FN results: 8	True Negative (TN): • Reality: Benign • ML model predicted: Benign • Number of TN results: 90

Intrinsic Metrics to evaluate NLP systems_4

- F1 Score – to combine P & R into a single metric.
- F1 Score – popular performance measure for multi-class classifier.

```
F1-score = 2 × (precision × recall)/(precision + recall)
```

F1-score?



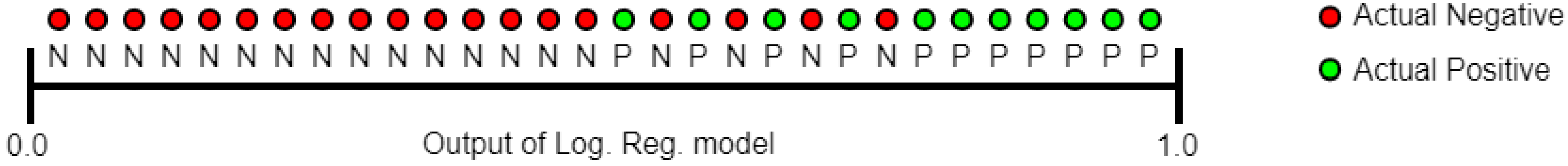
True Positive (TP): • Reality: Malignant • ML model predicted: Malignant • **Number of TP results: 1**	**False Positive (FP):** • Reality: Benign • ML model predicted: Malignant • **Number of FP results: 1**
False Negative (FN): • Reality: Malignant • ML model predicted: Benign • **Number of FN results: 8**	**True Negative (TN):** • Reality: Benign • ML model predicted: Benign • **Number of TN results: 90**

Intrinsic Metrics to evaluate NLP systems_5

- AUC (*Area Under the Curve*) – to quantify the model’s ability to separate the classes by capturing the count of positive predictions, which are correct, against the count of positive predictions that are incorrect at different thresholds.

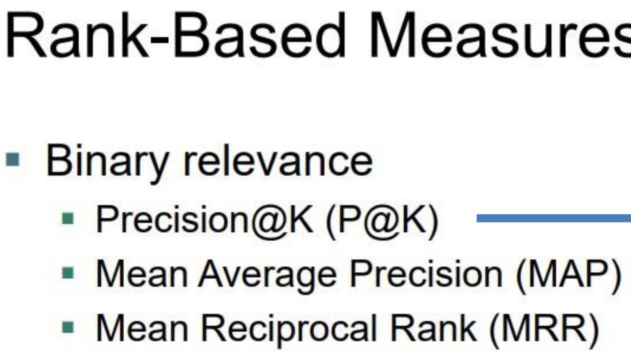
Ex:

Predictions ranked in ascending order of logistic regression score.


N N N N N N N N N N N N N N N P N P N P N P N P P P P P P P
0.0
Output of Log. Reg. model
1.0
Actual Negative
Actual Positive

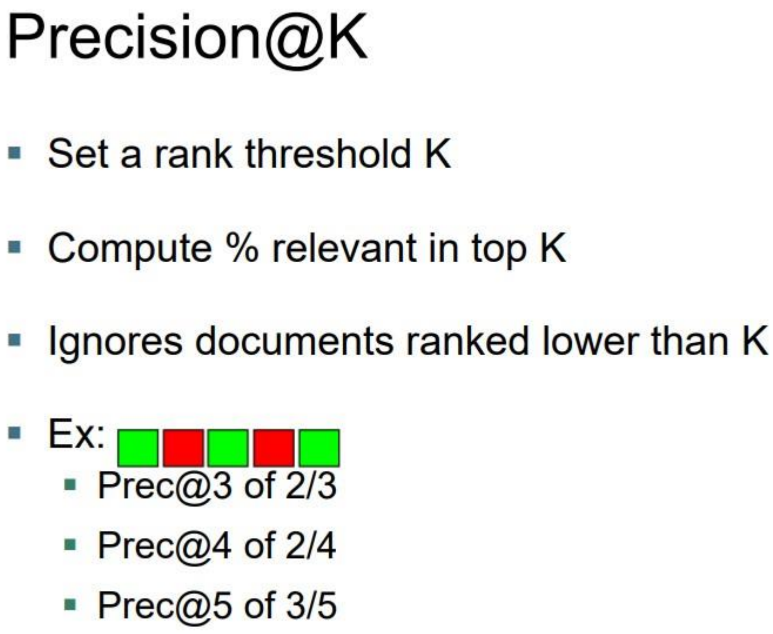
- AUC ranges in values from 0 to 1.

Information Retrieval_1



Rank-Based Measures

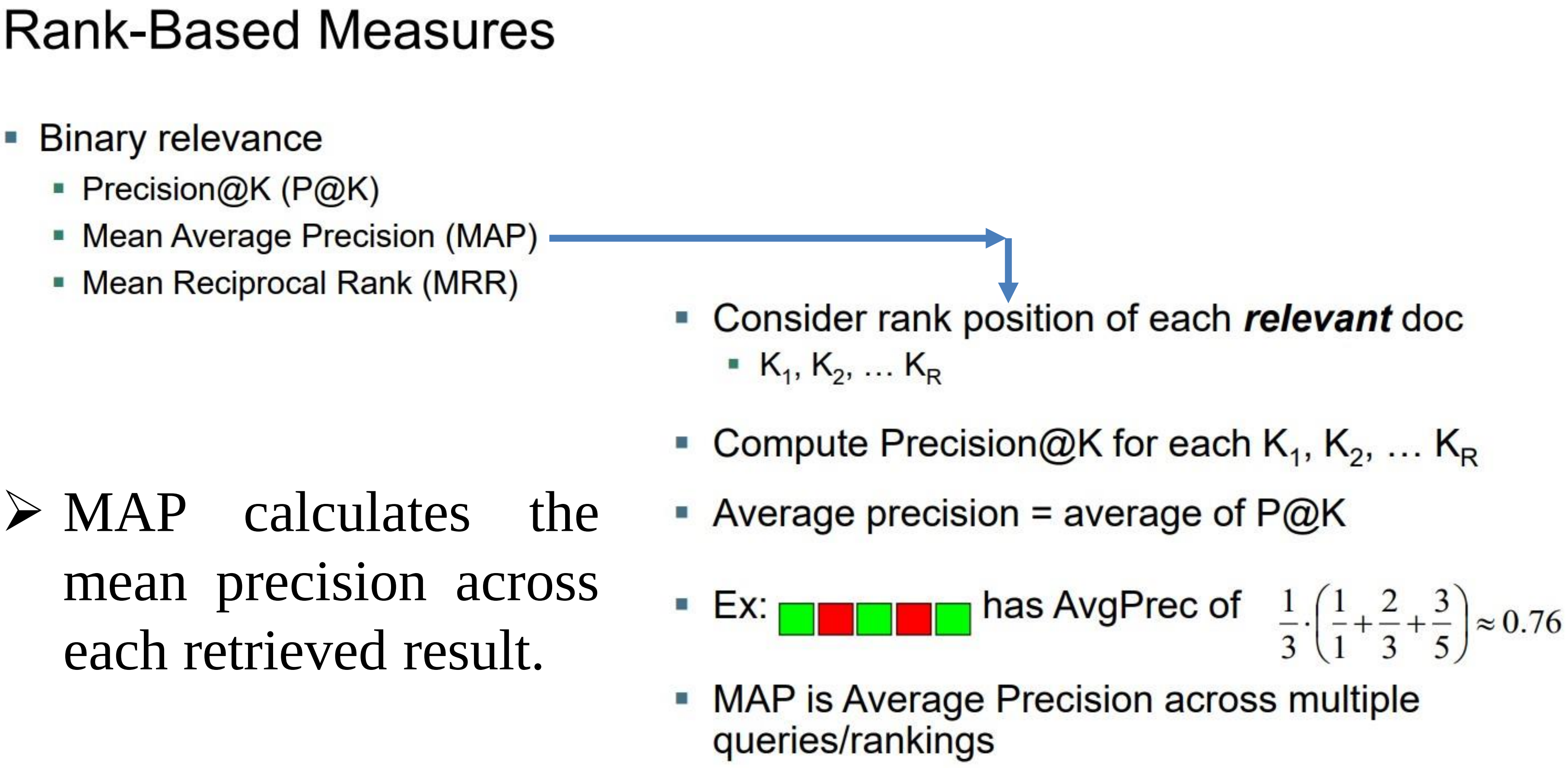
- Binary relevance
  - Precision@K (P@K)
  - Mean Average Precision (MAP)
  - Mean Reciprocal Rank (MRR)



Precision@K

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

Information Retrieval_2



Rank-Based Measures

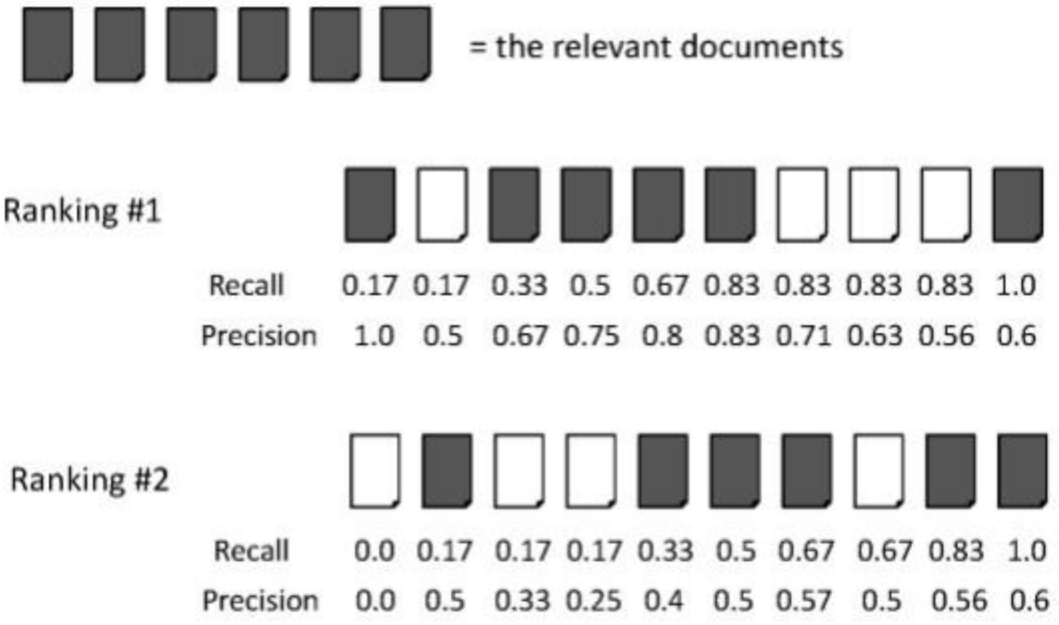
- Binary relevance
  - Precision@K (P@K)
  - Mean Average Precision (MAP)
  - Mean Reciprocal Rank (MRR)

- Consider rank position of each ***relevant*** doc
  - $K_1, K_2, \ldots K_R$
- Compute Precision@K for each $K_1, K_2, \ldots K_R$
- Average precision = average of P@K
- Ex: 🟩🟥🟩🟥🟩 has AvgPrec of $\frac{1}{3}\cdot\left(\frac{1}{1}+\frac{2}{3}+\frac{3}{5}\right)\approx 0.76$
- MAP is Average Precision across multiple queries/rankings

➢ MAP calculates the mean precision across each retrieved result.

Information Retrieval_3

AVERAGE PRECISION



■■■■■■ = the relevant documents

Ranking #1

	■	□	■	■	■	■	□	□	□	■
Recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	0.83	1.0
Precision	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.6

Ranking #2

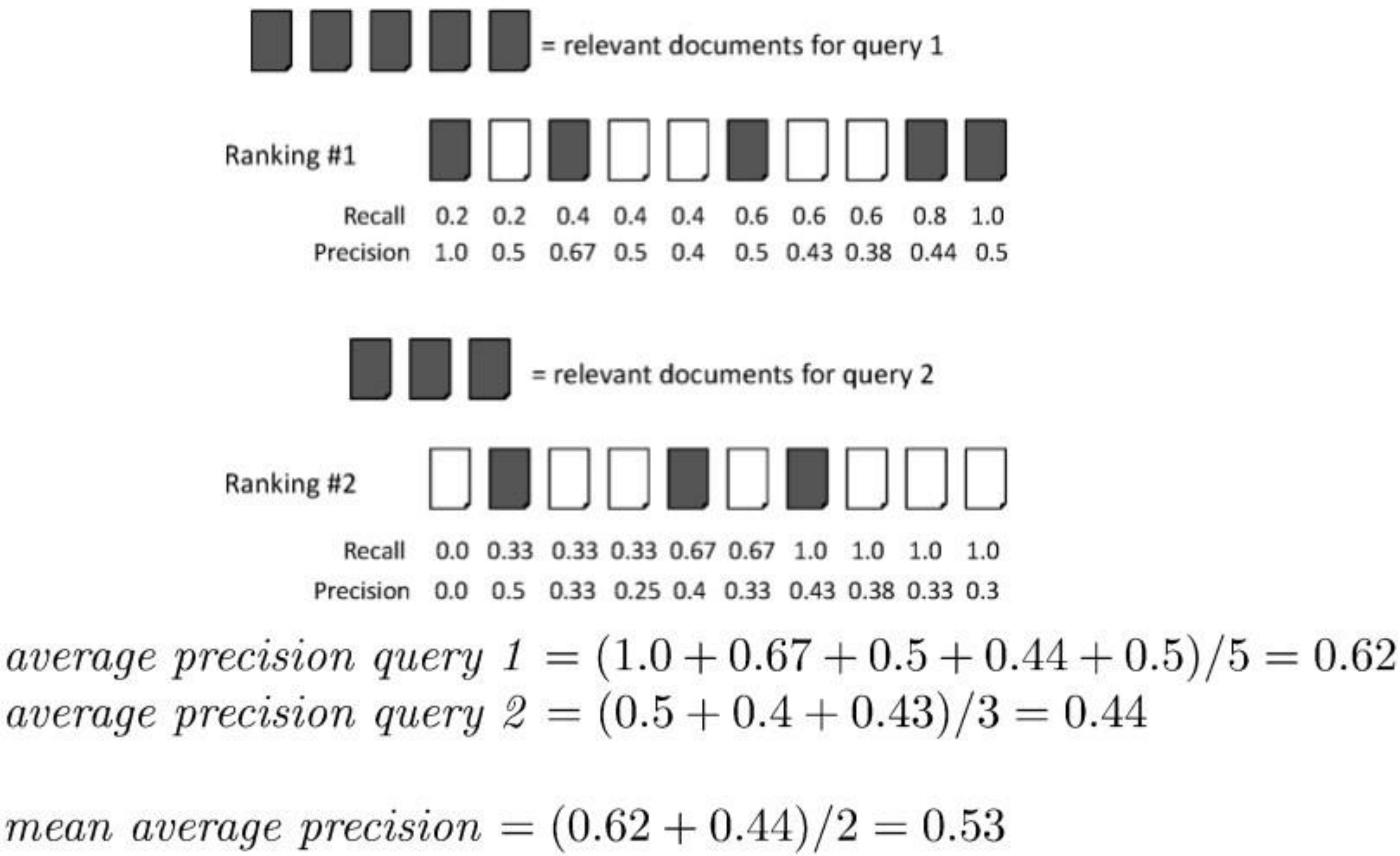
	□	■	□	□	■	■	■	□	■	■
Recall	0.0	0.17	0.17	0.17	0.33	0.5	0.67	0.67	0.83	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.5	0.57	0.5	0.56	0.6

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

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

Information Retrieval_4

MAP



■ ■ ■ ■ ■ = relevant documents for query 1

Ranking #1	■	□	■	□	□	■	□	□	■	■
Recall	0.2	0.2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	1.0
Precision	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5

■ ■ ■ = relevant documents for query 2

Ranking #2	□	■	□	□	■	□	■	□	□	□
Recall	0.0	0.33	0.33	0.33	0.67	0.67	1.0	1.0	1.0	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.33	0.43	0.38	0.33	0.3

$$\textit{average precision query 1} = (1.0 + 0.67 + 0.5 + 0.44 + 0.5)/5 = 0.62$$

$$\textit{average precision query 2} = (0.5 + 0.4 + 0.43)/3 = 0.44$$

$$\textit{mean average precision} = (0.62 + 0.44)/2 = 0.53$$

Information Retrieval_5

MAP

- 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;
- Now perhaps most commonly used measure in research papers
- Good for web search?
- MAP assumes user is interested in finding many relevant documents for each query
- MAP requires many relevance judgments in text collection.

Information Retrieval_6

When there's inly 1 relevant document?

✓ Scenarios:

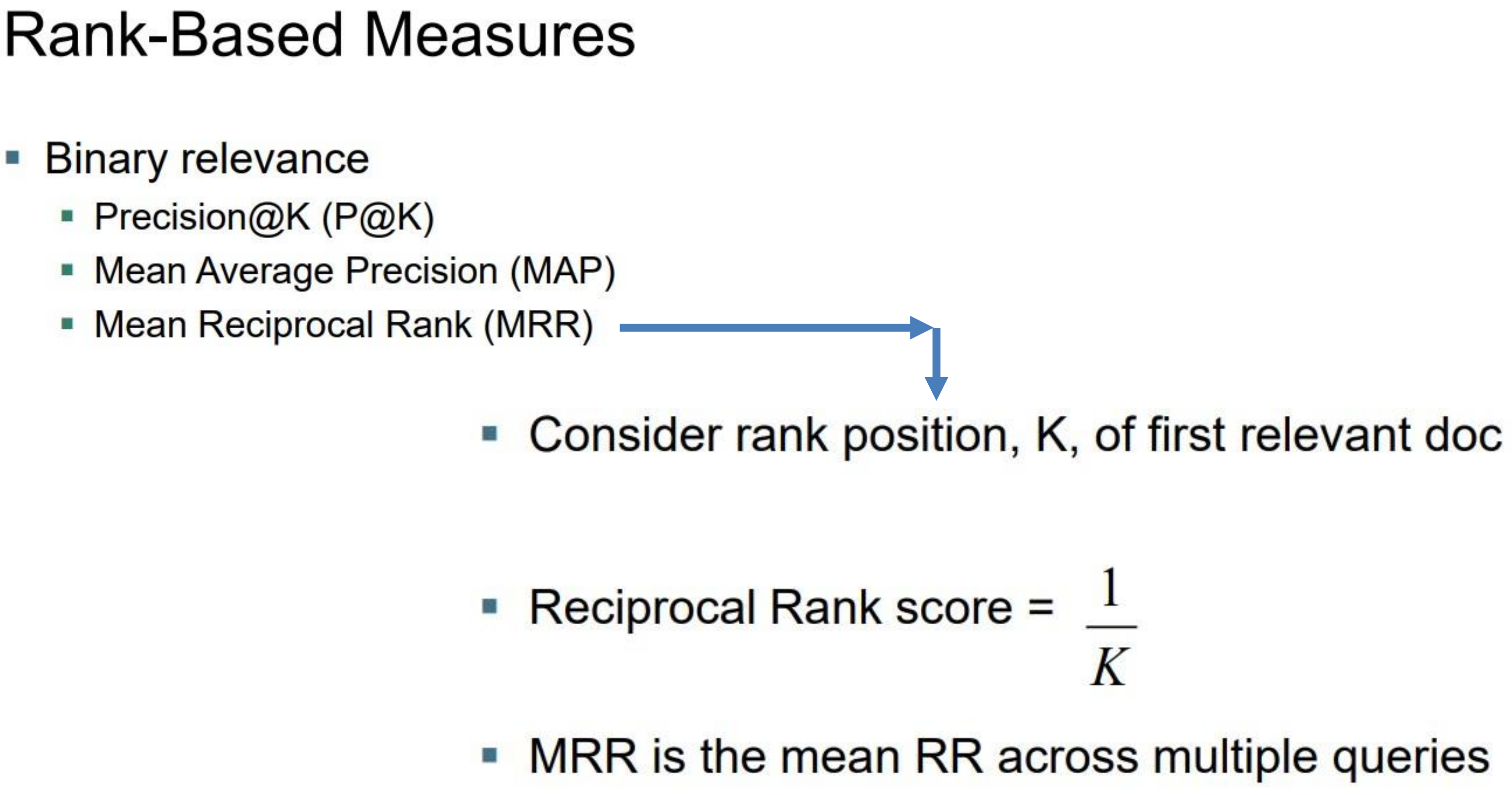
1. known-item search
2. navigational queries
3. looking for a fact

✓ Search Length = Rank of the answer

1. measures a user's effort

Information Retrieval_7

- MRR (*Mean Reciprocal Rank*) – to evaluate the responses retrieved, in correspondence to a query, given their probability of correctness.
- MRR = typically used in informational retrieval (IR) tasks.



Rank-Based Measures

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

- Consider rank position, K, of first relevant doc
- Reciprocal Rank score = $\frac{1}{K}$
- MRR is the mean RR across multiple queries

Summary & Machine Translation_1

- ROUGE (*Recall-Oriented Understudy for Gisting Evaluation*) – measures the recall.
- ROUGE = typically used for evaluating the quality of generated text, summarization, and in machine translation (ML) tasks.
- ROUGE Evaluation Packages:

  (1) https://github.com/kylehg/summarizer/blob/master/rouge/ROUGE-1.5.5.pl (in Perl)

  (2) https://github.com/kenlimmj/rouge (in Java)

Summary & Machine Translation_2

5 evaluation metrics

ROUGE-N: Overlap of N-grams between the system and reference summaries.

ROUGE-1: Overlap of 1-gram (each word) between the system and reference summaries.

ROUGE-2: Overlap of bigrams between the system and reference summaries.

ROUGE-L: Longest Common Subsequence (LCS) based statistics.

ROUGE-W: Weighted LCS-based statistics that favors consecutive LCSes.

ROUGE-S: Skip-bigram based co-occurrence statistics.

ROUGE-SU: Skip-bigram plus unigram-based co-occurrence

ROUGE_1

Example:

System Summary: the cat was found under the bed

Reference Summary: the cat was under the bed

System Summary Bigrams: the cat,
cat was,
was found,
found under,
under the,
the bed

Reference Summary Bigrams: the cat,
cat was,
was under,
under the,
the bed

$$ROUGE2_{Recall} = \frac{4}{5} = 0.8 \qquad ROUGE2_{Precision} = \frac{4}{6} = 0.67$$

ROUGE_2

Python code - https://github.com/pcyin/PyRouge

```
from PyRouge.pyrouge import Rouge

r = Rouge()

system_generated_summary = "The Kyrgyz President pushed through the law requiring the use of
ink during the upcoming Parliamentary and Presidential elections In an effort to live up to
its reputation in the 1990s as an island of democracy. The use of ink is one part of a
general effort to show commitment towards more open elections. improper use of this type of
ink can cause additional problems as the elections in Afghanistan showed. The use of ink and
readers by itself is not a panacea for election ills."

manual_summmary = "The use of invisible ink and ultraviolet readers in the elections of the
Kyrgyz Republic which is a small, mountainous state of the former Soviet republic, causing
both worries and guarded optimism among different sectors of the population. Though the
actual technology behind the ink is not complicated, the presence of ultraviolet light (of
the kind used to verify money) causes the ink to glow with a neon yellow light. But, this use
of the new technology has caused a lot of problems. "

[precision, recall, f_score] = r.rouge_l([system_generated_summary], [manual_summmary])

print("Precision is :"+str(precision)+"\nRecall is :"+str(recall)+"\nF Score is
:"+str(f_score))

#output
""""
Precision is :0.446058091286
Recall is :0.439672801636
F Score is :0.442843380487
""″″″
```

Installing PyRouge - https://github.com/pltrdy/rouge

Installing pyrouge in ubunut 16.04 -
https://sagorbrur.github.io/install_rouge.html

Thank you