

# Analyzing Massive Data Sets

## Summer Semester 2019

Prof. Dr. Peter Fischer

Institut für Informatik

Lehrstuhl für Datenbanken und Informationssysteme

## Chapter 6: Fulltext Retrieval – Part 2: Evaluation

# What to Evaluate?

- Evaluation is the key to build **effective** and **efficient** information retrieval engines
  - measurement usually carried out in controlled **laboratory experiments**
  - **online testing** can also be done
  - **Effectiveness, efficiency** and **cost** are related
    - e.g., if we want a particular level of effectiveness and efficiency, this will determine the cost of the system configuration
    - efficiency and cost targets may impact effectiveness
- What should be evaluated in IR?
  - **Effectiveness**: measures the ability of the search engine to find the right information
  - **Efficiency**: measures how quickly this is done

# Creating Assessment Data

# Evaluating Relevance

- How to **evaluate** a system's **result quality**?
  - Traditional approach: **Evaluation benchmarks**
    - A benchmark document **collection**
    - A benchmark suite of **information needs**, expressible as **queries**
    - An **assessment** of the relevance of each query-document pair, called **"gold standard"** or **"ground truth"**
    - Usually, relevance is assessed in binary fashion
  - *More recent, complementary approach: user studies with relevance feedback*
    - *Explicit studies very cumbersome*
    - *Request Log Mining*
    - *Crowdsourcing*
- Example of an information need:
  - *"What are the prospects of the Quebec separatists achieving independence from the rest of Canada?"*

# First approach: Test Collections

- Researchers assembled **test collections** (aka *evaluation corpus*) consisting of **documents**, **queries**, and **relevance judgments** to address effectiveness
  - **Goal:** Provide test collections such that results from different techniques can be compared
- The following *test collections* were created at intervals of about 10 years to reflect the changes in data and user communities for typical search applications
  - CACM: Titles and abstracts from the Communications of the ACM from 1958–1979. Queries and relevance judgments generated by computer scientists.
  - AP: Associated Press newswire documents from 1988–1990 (from TREC disks 1–3). Queries are the title fields from TREC topics 51–150. Topics and relevance judgments generated by government information analysts.
  - GOV2: Web pages crawled from websites in the .gov domain during early 2004. Queries are the title fields from TREC topics 701–850. Topics and relevance judgments generated by government analysts.

# Statistics for test collections

- **Text**

Collection	Number of documents	Size	Average number of words/doc.
CACM	3,204	2.2 MB	64
AP	242,918	0.7 GB	474
GOV2	25,205,179	426 GB	1073

- **Queries**

Collection	Number of queries	Average number of words/query	Average number of relevant docs/query
CACM	64	13.0	16
AP	100	4.3	220
GOV2	150	3.1	180

# Query Topic Example

- Queries for AP and GOV2 collections based on **TREC** topics
- Created by information analysts employed by NIST
  - Early topics were designed to reflect the needs of professional analysts in government and industry (quite complex)
  - Now, topics are meant to represent general information needs

<top>

<num> Number: 794

<title> pet therapy     **short query**

<desc> Description:

How are pets or animals used in therapy for humans and what are the benefits?

**long query**

<narr> Narrative:

## **Relevance criteria for humans**

Relevant documents must include details of how pet- or animal-assisted therapy is or has been used. Relevant details include information about pet therapy programs, descriptions of the circumstances in which pet therapy is used, the benefits of this type of therapy, the degree of success of this therapy, and any laws or regulations governing it.

</top>

# Relevance Judgments

- Obtaining **relevance judgments** is an expensive, time-consuming process
  - Who does it
  - What are the instructions?
  - What is the level of agreement?
- **TREC judgments**
  - Generally binary
  - but sometimes multiple level of relevance may be appropriate
  - Some TREC and GOV2 collections were judged as *not relevant*, *relevant*, and *highly relevant*
  - Depend on task being evaluated
  - TREC analysts judged a document as relevant if it contained information that could be used to help write a report on the query topic
  - Agreement good because of "narrative"
  - TREC primarily focused on *topical relevance*



# Request Log Mining

- **Query logs** capture user interactions with a search engine
  - Logs provide a **large amount of data** showing **how users browse the results** that a search engine provides for a query
  - Used for both **tuning** and **evaluating** search engines
  - **Number of users** and **queries** in query logs can number in the tens of **millions**
  - Compared to the hundreds of queries in TREC, **query log data** can **support** much more
- **extensive and realistic evaluation**
- **Drawback:**
  - This data is **not** as **precise** as explicit relevance judgments
  - **Privacy?** => **Anonymize** logged data
- **Typical contents**
  - User identifier or user session identifier
  - Query terms - stored exactly as user entered
  - List of URLs of results, their ranks on the result list, and whether they were **clicked** on
  - Timestamp(s) - records the time of user events such as query submission, clicks

# Clicks as user feedback

- **Clicks** on result pages are highly **correlated** with **relevance**, but cannot be used directly in place of explicit relevance judgments
  - they are very **biased** toward pages that are highly ranked or have other features such as being popular or having a good snippet on the result page
  - This means, that pages at the top rank are clicked on much more frequently than lower ranked pages, even when the relevant pages are at the lower ranks
- One can use **clickthrough data** to predict **preferences** between pairs of documents
  - A **preference** for document **D1** compared to **D2** means that **D1** is **more relevant**
  - Preferences are appropriate for search tasks with **multiple levels of relevance** and are **focused** on **user relevance** than purely *topical relevance*
  - **Relevance judgments** can be used to **generate preferences**, but preferences do not imply specific relevance levels

# Example Click Policy

- The bias in clickthrough data is addressed by "strategies" that generate preferences
- One strategy is ***Skip Above and Skip Next*** (Agichtein, Brill, Dumais, Ragno, 2006)
  - Assume a given **set of results** for a query,
  - given a **clicked result** at rank **position  $p$**
  - all **unclicked** results **ranked above  $p$**  are predicted to be **less relevant** than  $p$
  - Unclicked results immediately following a clicked results are less relevant than the clicked result

– click data

$d_1$   
 $d_2$   
 $d_3$  (clicked)  
 $d_4$

– generated preferences

$d_3 > d_2$   
 $d_3 > d_1$   
 $d_3 > d_4$

# Dealing with Noise

- This "**data**" can be **noisy** and **inconsistent** because of the variability in user's behavior
- **Query logs** typically contain **many instances** of the **same query** submitted by different users.  
Therefore, clickthrough data can be **aggregated** to **remove noise** from individual differences
- **Click distribution** information
  - can be used to identify clicks that have a **higher frequency** than would be expected
  - **high correlation** with **relevance**
  - e.g., using **click deviation** to filter clicks for preference-generation policies

# Effectiveness Metrics

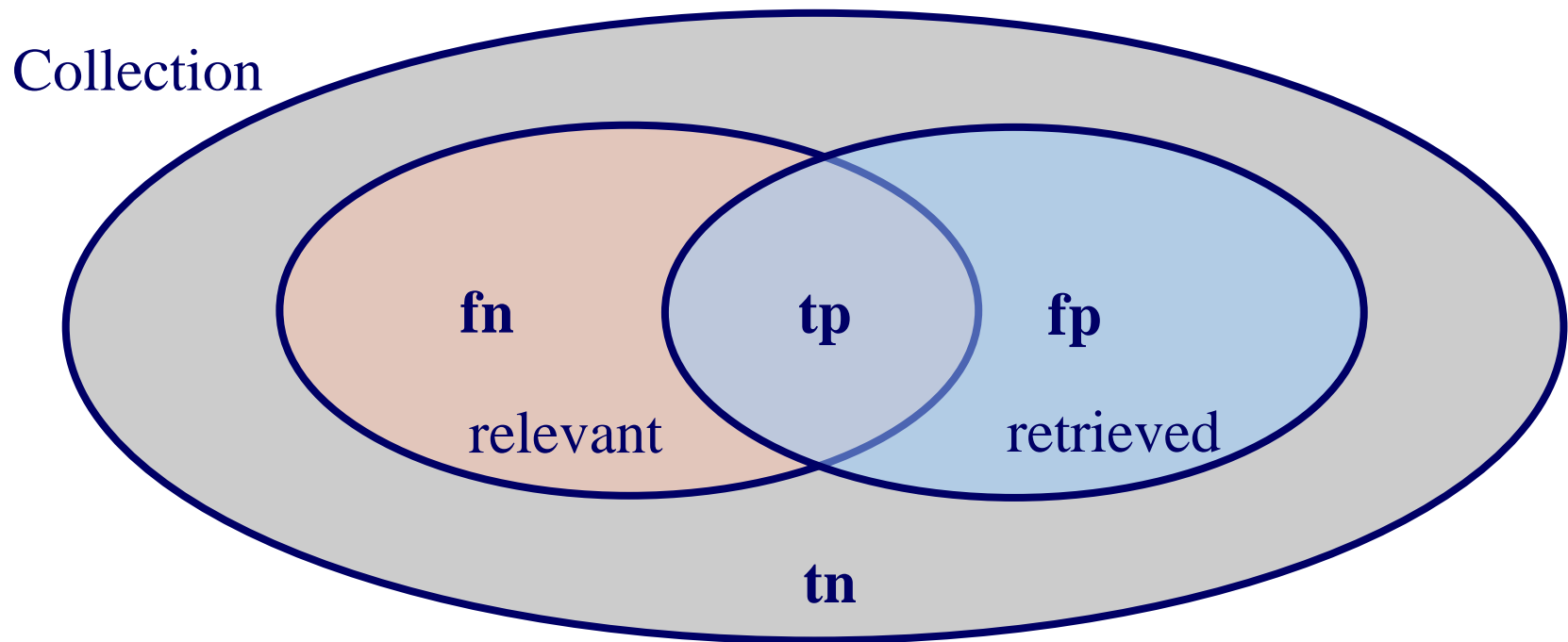
# Effectiveness Metrics

- Users wish useful results
  - Difficult to measure
- Approximated by relevant results on query, but
  - Irrelevant results may be useful ("lucky hit")
  - Relevant results may be useless (e.g., outdated)
- For assessment, typically binary
- Set oriented/global relevance metrics:
  - how many of the relevant results have been produced
  - How many of the returned results are correct
  - ...
- Ranked metrics:
  - Is a relevant results in the Top K?
  - How closely is the relevance order matched?

# Classification results

- Usually with binary relevance  
(if necessary using thresholds)
- **true positives**
- **true negatives**
- **false positives**
  - **non-relevant documents retrieved**
  - extend the result set unnecessarily
  - often inevitable
  - usually can be filtered out by the user quite easily
- **false negatives**
  - **relevant documents not retrieved**
  - Problematic, since the user usually is not aware of them
  - Often *worse than false positives*

# Visualizing Classification Classes



□  $|\text{Relevant}| = \text{tp} + \text{fn}$

□  $|\text{Retrieved}| = \text{fp} + \text{tp}$



# Confusion Matrices (I)

- Arrange (frequency) results along the two dimension in the matrix
  - Actual Relevance and Non-Relevance
  - Retrieved/Non-Retrieved
- Classes
  - $A$  = set of **relevant** documents
  - $\bar{A}$  = set of **non-relevant** documents
  - $B$  = set of **retrieved** documents
  - $\bar{B}$  = set of **not-retrieved** documents
- Provide a concise overview on distributions

	Relevant	Non-Relevant
Retrieved	$A \cap B$	$\bar{A} \cap B$
Non-Retrieved	$A \cap \bar{B}$	$\bar{A} \cap \bar{B}$

# Confusion Matrices (I)

- Slicing and aggregating a confusion matrix provide a wealth of possible metrics

		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, Power	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 $= \frac{\sum \text{True positive}}{\sum \text{Condition positive}}$	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) = $\frac{\text{LR+}}{\text{LR-}}$	F <sub>1</sub> score = $\frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$
		False negative rate (FNR), Miss rate $= \frac{\sum \text{False negative}}{\sum \text{Condition positive}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\sum \text{True negative}}{\sum \text{Condition negative}}$	Negative likelihood ratio (LR-) = $\frac{\text{FNR}}{\text{TNR}}$		

# Common IR Evaluation Metrics

- **Precision** and **Recall** were introduced to summarize and compare (unranked) search results
  - **Recall**: measures how well the search engine is doing at **finding** all the **relevant documents** for a query
  - **Precision**: measures how well it is doing at **rejecting non-relevant documents**

# Precision

- **How many of the returned documents are relevant?**
  - Values in **[0,1]**, where **1** is **best**
  - **High precision** is important in **Web search**
  - Uses the number of **true positives** as measure of result quality

$$Precision = \frac{|A \cap B|}{|B|} = \frac{\#relevant\ documents\ retrieved}{\#documents\ retrieved}$$

# Recall

- **How many of the relevant documents have been returned?**
  - Values in **[0,1]**, where **1** is **best**
  - **High recall** usually is important for **professional searchers** such as paralegals and intelligence analysts
  - Also uses the number of **true positives** as measure of result quality
- $Recall = \frac{|A \cap B|}{|A|} = \frac{\#relevant\ documents\ retrieved}{\#relevant\ documents}$

# Classification Errors

- **false positives - Type I error**

- a non-relevant document is retrieved
- How many retrieved documents have been non-relevant?
- Value in  $[0,1]$ , where 0 is best

$$\begin{aligned} \text{Fallout} &= \frac{|\bar{A} \cap \bar{B}|}{|\bar{A}|} \\ &= \frac{\#non - relevant\ documents\ retrieved}{\#non - relevant\ documents} \end{aligned}$$

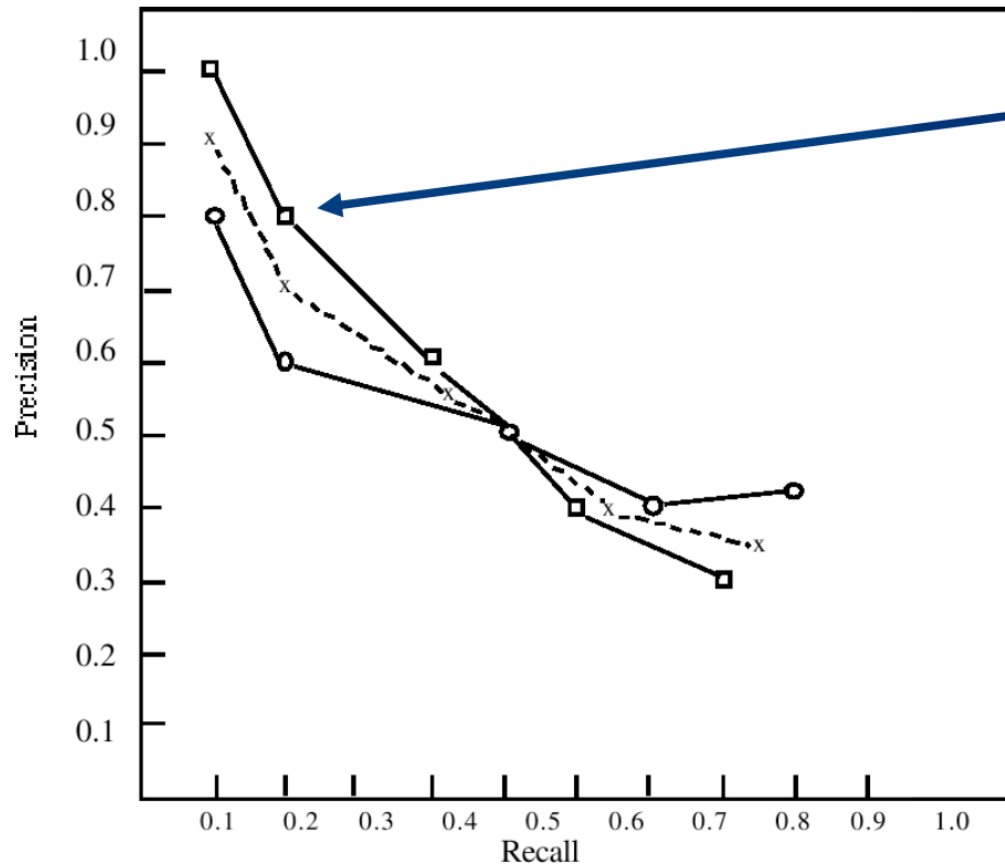
- **false negatives - Type II error**

- a relevant document is not retrieved
- Definition:  $1 - \text{Recall}$

# Precision vs. Recall

- Precision and recall clearly trade off against one another:
  - Achieve **perfect recall** (but awful precision) by always returning all documents in the collection
  - Achieve **very good precision** (but awful recall) by always returning only the single result that seems to match best
- Normally, this leads to tradeoffs in system tuning
  - Small result sets usually lead to better precision but worse recall
- What about measurement?
  - **Precision is easy to measure**
  - **Measuring recall is at least very difficult, and often impossible**

# Precision-Recall Curve



**Average precision  
of system 3 at  
recall level 0.2**

Which system is best?

**What's more  
important:  
Precision or recall?**



# F Measure

- **Fallout** and recall together characterize the effectiveness of a search
- **Precision is more meaningful** to the user
- **F measure** (sometimes also called F1) is an effectiveness measure based on **recall** and **precision** that is used for evaluating **classification performance**
- It's a weighted **harmonic mean** of *recall and precision*
- Advantage: Summarizes effectiveness in a **single number**
- Value in [0,1], where 1 is best

$$F = \frac{2}{\frac{1}{R} + \frac{1}{P}} = \frac{2RP}{(R + P)}$$

- »More general form, where  $\beta$  is a parameter that determines **relative importance** of **recall** and **precision**

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

# Ranking Metrics

# Ranking Effectiveness

- Retrieval models usually produce **ranked output**
  - Results in Top K more important than whole set
  - Order with Top K matters
- **Idea: Apply recall and precision on ranked documents**
  - Calculate recall and precision values at every rank position for a set of **top-k retrieved documents**, e.g., top-10 (as if it was the full
  - Then get **precision at k** and **recall at k**
  - top-10 documents of two possible rankings with recall and precision at every rank position
- **Example**
  - Relevant: 10, 582, 877, 10003
  - Result list: 582, 17, 5666, 10003, 10, 37, ...
  - **P@1:**
  - **P@2:**
  - **P@3:**
  - **P@4:**
  - **P@5:**

# Applying P@K and R@K

- 10 documents,
- 6 relevant results



- two rankings
- Which set scores?
  - **R = 1**
  - **P = 0.6**

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

- Which is better?
- Determine P/R@K!
- Do higher rank positions matter more?

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

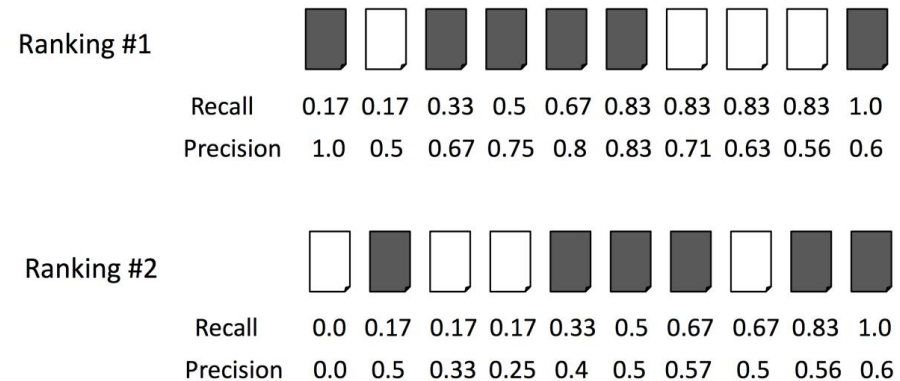
- Ranking #1 better,  
e.g., at position 4, Ranking #1 R = 0.5 / P = 0.75  
Ranking #2 R = 0.17 / P = 0.25

# Average Precision

- **Problem:** For a **large number of relevant documents** the list of recall-precision values will be long and unwieldy
- **Solutions:**
  - Calculating recall and precision at **fixed rank positions**
  - Calculating **precision** at **standard recall levels**, from 0.0 to 1.0 in increments of 0.1
    - » Here, only the **precision** values at levels **0.5** and **1.0** will be calculated
    - Rest requires *interpolation* (next section)
  - **Averaging the precision** values from the rank positions where a relevant document was retrieved
    - » Advantage: single number based on ranking of relevant documents
    - » **Measures effectiveness of ranking** for a **single query**



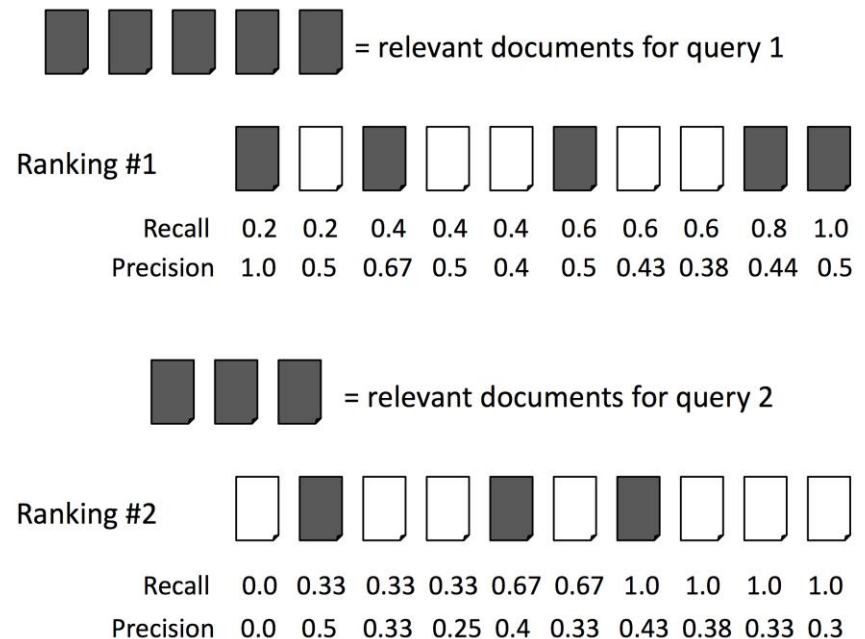
= the relevant documents



- » *Average precision* for the rankings
- R1:  $(1.0 + 0.67 + 0.75 + 0.8 + 0.83 + 0.6) / 6 = 0.78$
- R2:  $(0.5 + 0.4 + 0.5 + 0.57 + 0.56 + 0.6) / 6 = 0.52$

# Averaging Across Queries


- To provide a **realistic assessment** of the effectiveness of a ranking algorithm, it must be tested on a **number of queries**
- **Goal:** summarize the effectiveness of a ranking algorithm **across** a collection of **queries**
- Recall and precision values for rankings from two **different queries**













# Mean Precisions


- Given a benchmark with several queries + ground truth
- Then one can capture the quality of a system by taking the **mean** (average) of a given measure over all queries
  - **MP@k** = mean of the P@k values over all queries
  - **MP@R** = mean of the P@R values over all queries
  - **MAP** = mean of the mean average precision value for each query
    - Most commonly used measure in research papers
    - Assumes user is interested in finding many relevant documents for each query
    - Requires many relevance judgments in text collection

# MAP Example











 = 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

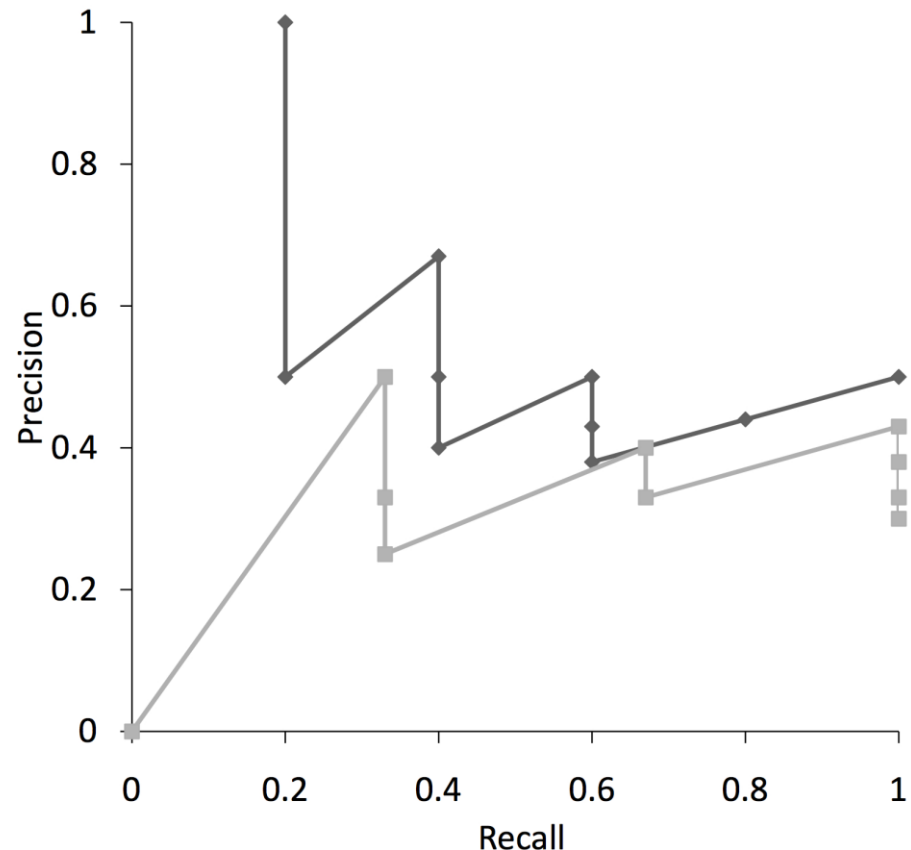
										
Recall	0.0	0.33	0.33	0.33	0.67	0.67	1.0	1.0	1.0	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.33	0.43	0.38	0.33	0.3

- AP Query 1 =  $(1.0 + 0.67 + 0.5 + 0.44 + 0.5) / 5 = 0.62$
- AP Query 2 =  $(0.5 + 0.4 + 0.43) / 3 = 0.44$
- MAP =  $(0.62 + 0.44) / 2 = 0.53$



# Recall-Precision Graph

- The **MAP** measure provides a very **succinct summary** of the **effectiveness** of a ranking algorithm over many queries
- However, sometimes too much **information** is **lost** in this process
- **Recall-precision graphs** give more detail on the effectiveness of ranking algorithms at different recall levels
- Graphs for individual queries have very different shapes and are **difficult to compare**



# Discounted Cumulative Gain (DCG, nDCG)

- Non-Binary relevance, e.g.,  
0 = not relevant, 1 = somewhat rel, 2 = very relevant
- There should be a "discount" in the score if very relevant documents come after only somewhat relevant documents
- **Cumulative gain:**  $CG@k = \sum_{i=1..k} rel_i$
- **Discounted CG:**  $DCG@k = rel_1 + \sum_{i=2..k} rel_i / \log_2 i$
- Problem: CG and DCG are larger for larger result lists
- Solution: normalize by maximally achievable value
- **Ideal DCG:**  $iDCG@k = DCG@k$  of ideal ranking
- **Normalized DCG:**  $nDCG@k = DCG@k / iDCG@k$

# nDCG Example

- Consider the following result list and relevances, assuming
- only 3 relevant documents overall (for this query)
- Hit #1: very relevant 2
- Hit #2: relevant 1
- Hit #3: not relevant 0
- Hit #4: very relevant 2
- Hit #5: not relevant 0
- Then **DCG@5** =
- And **iDCG@5** =
- Hence **nDGC@5** =