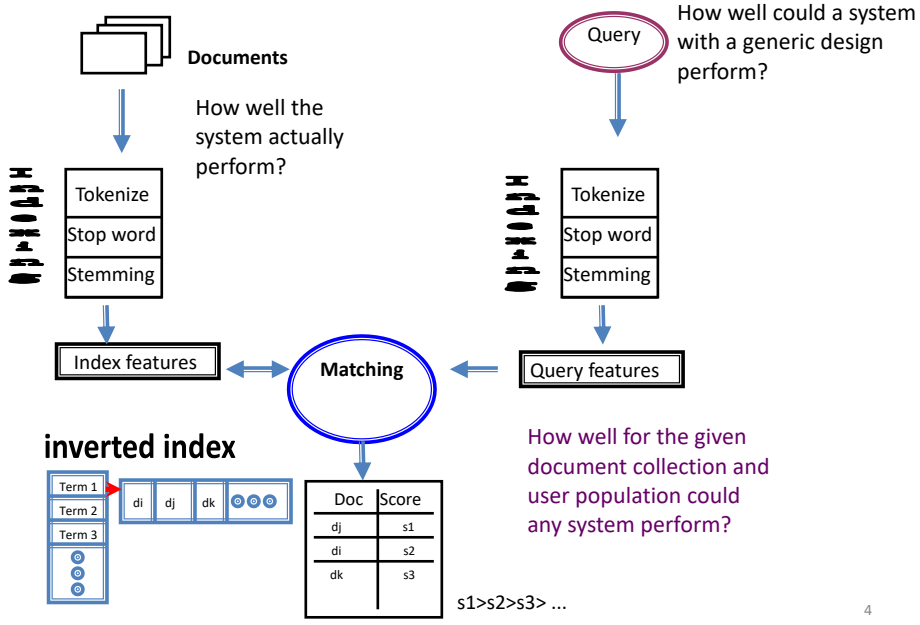
Why Evaluate?

- There are three main reasons for evaluating IR systems
  - Economic reasons: if people are going to buy the technology, they want to know how effective it is
  - Scientific progress:
    - Researchers want to know if progress is being made or not. So, we need a measure for **progress**
    - Scientists want to show that their method is better than someone else's. Therefore, we need a measure for **better**
  - Verification: if you built a system you need to verify its performance

Architecture of an IR system



What to Evaluate?

- **Efficiency**
  - E.g. query latency (time lag), query throughput
- Coverage
  - Does this system cover the domain very well?
    - How many pages does a web search engine index?
- Presentation of results/query formulation
  - Effort of the user
  - User engagement aspects; Cognition; User attention
- **Effectiveness**
  - How effective the system is?
    - Correctness

Evaluation

- Evaluation is key to building *effective* and *efficient* search engines
  - Usually carried out in controlled laboratory experiments
  - *Online* testing/evaluation can also be done
- Effectiveness, efficiency and *cost* are related
  - High efficiency may be obtained at the price of effectiveness
    - 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

Test Collections

- Collections of documents that also have associated with them:
  - A set of queries for which **relevance assessments** are available.
  - Relevance assessments are often called "**qrels**" for short.
- Documents? What are they?
  - Genuine documents like email messages, journal articles, memos etc. kind of material for which we tend to look for some information
- Queries? What are they?
  - Kind of questions users of such collection will ask. How do we collect them? Co-operation of user population, query logs, etc.

GROUND TRUTH

TREC: Text Retrieval Conference

- Started in 1992, organised by NIST, USA and funded by the US government.
- Introduced a new standard for retrieval system evaluation
- Developed diverse test collections of different sizes (recent collections have > 1 billion docs)
  - News articles, web documents, blogs, tweets, etc.
- Avoid exhaustive assessment of documents using the pooling method

Evaluation Corpus

- ***Test collections*** consisting of documents, queries, and relevance judgments, e.g.,
  - 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.

9

TREC Topic Example

<top>
<num> Number: 794

<title> pet therapy

<desc> Description:
How are pets or animals used in therapy for humans and what are the benefits?

<narr> Narrative:
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>

10

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
  - Depend on task being evaluated
  - Binary vs. Graded relevance
  - Reasonable agreement because of "narrative"

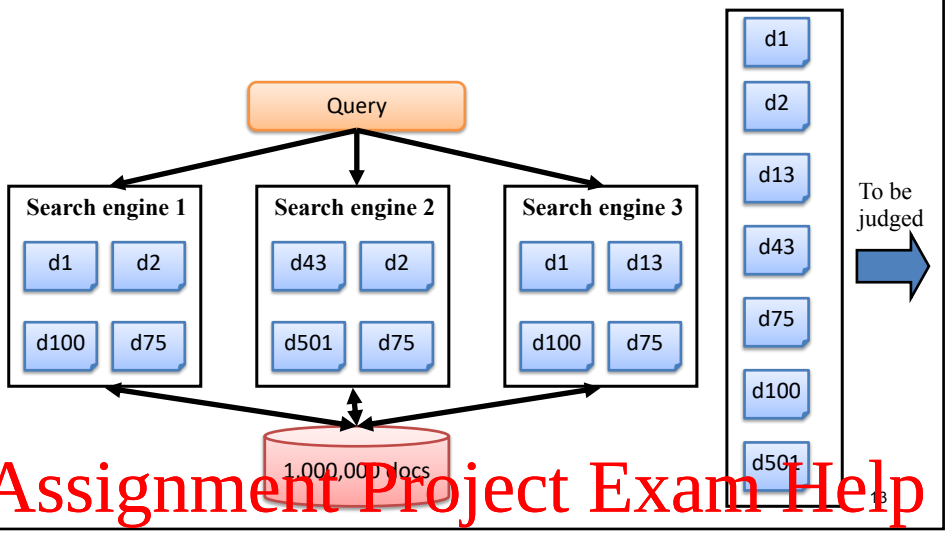
Pooling

- Exhaustive judgments for all documents in a collection is not practical
  - i.e. we can't judge 1 billion of documents
- Pooling technique is used in TREC
  - Top *k results* (*k varied between 50 and* 200) from the rankings obtained by different search engines are merged into a **pool**
  - Duplicates are removed
  - Documents are presented in some random order to the relevance judges
- Produces a large number of relevance judgments for each query, although possibly incomplete

where are top k results from?
how to generate them?

12

Pooling Technique



Relevance Assessments @ TREC

Assessors inspecting documents for relevance at TREC - a thankless job!

IR Experimental Setup

- A test collection: docs, queries, & relevance assessments (Ground Truth)
- Measures of performance
  - Precision, Recall, ….
- Systems to compare
  - *System A vs System B*
  - *E.g. A uses TF weighting; B uses TF-IDF:*
- An experimental design
  - Traditionally, the same queries and documents are used repeatedly, with different systems or variants of systems

Describe the classical IR experimental setup

Evaluation Method

- **Run your** experiment:
  - Input the documents to the systems
  - Issue each query to the systems
- **Collect the output**
  - Then you need to evaluate the output using the **ground truth**
  - Analyse the results
    - Quantitatively and statistically!
      - How to do this?

16

Comparing Systems

- Given a query, the IR system provides a ranked list after searching the underlying collection of documents
- The assumption is
  - The better system will provide a ***better ranked list***
  - A better ranked list satisfies the users overall
- So?
  - How to compare (2 or more) ranked lists of documents?
  - How to verify whether one system is more **effective** than another?
  - We need a **measure** (metric)!

Effectiveness Measures

- The function of an IR system is to:
  - **Retrieve all relevant documents**
    - Measured by **recall**
  - **Retrieve no non-relevant documents,**
    - Measured by **precision**
- Classical **effectiveness** Measures
  - Recall & Precision

Effectiveness Measures

A is set of relevant documents,
B is set of retrieved documents

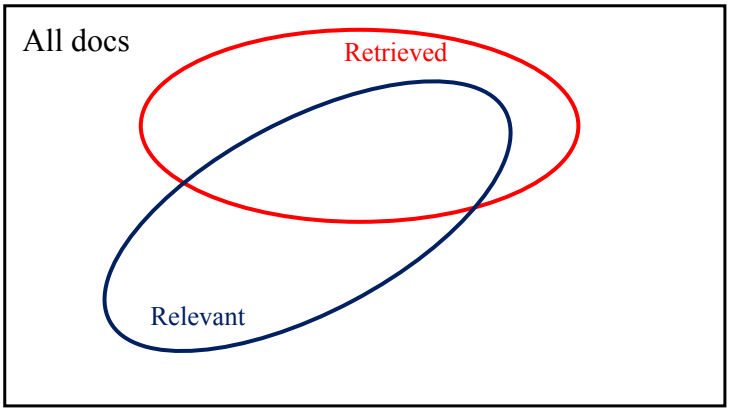
	Relevant	Non-Relevant
Retrieved	$A \cap B$	$\overline{A} \cap B$
Not Retrieved	$A \cap \overline{B}$	$\overline{A} \cap \overline{B}$

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

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

Relevant vs. Retrieved

For a given query and a collection

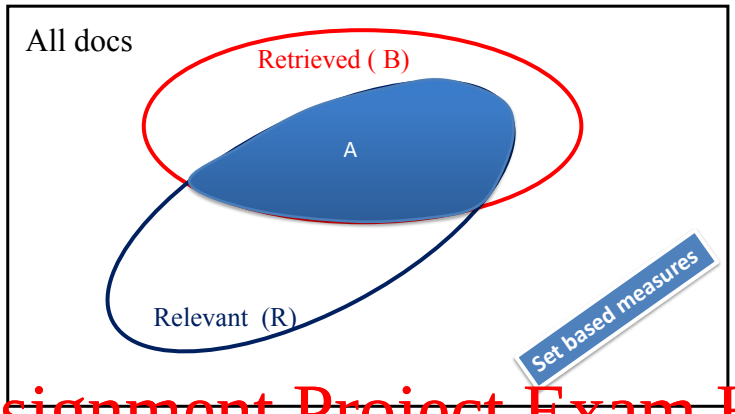


20

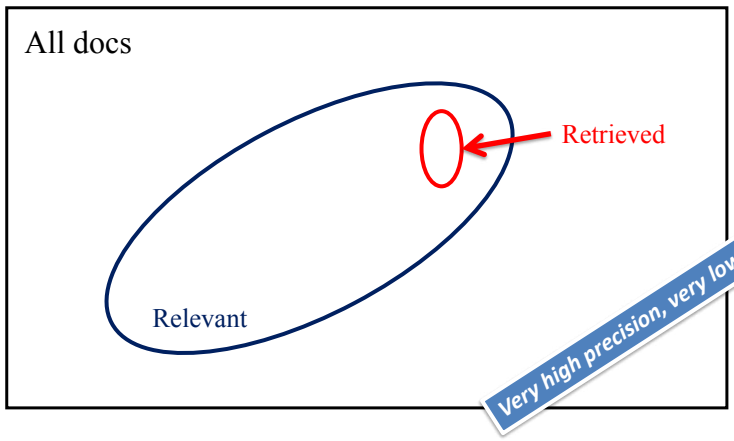
Relevant vs. Retrieved Precision & Recall

Precision = $\frac{|A|}{|B|}$

Recall = $\frac{|A|}{|R|}$

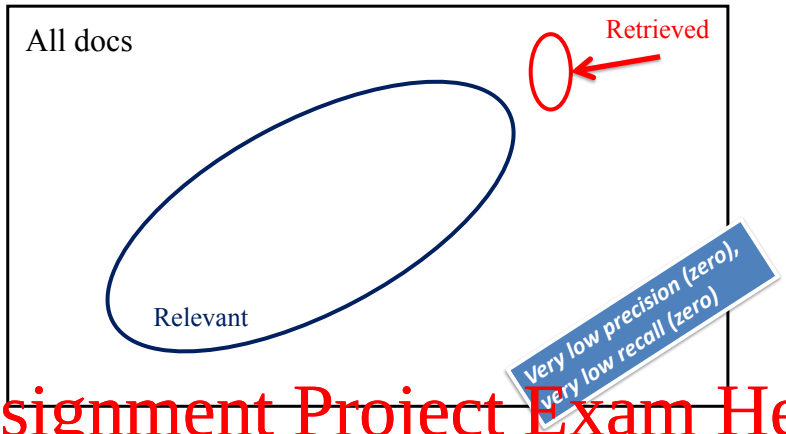


Relevant vs. Retrieved

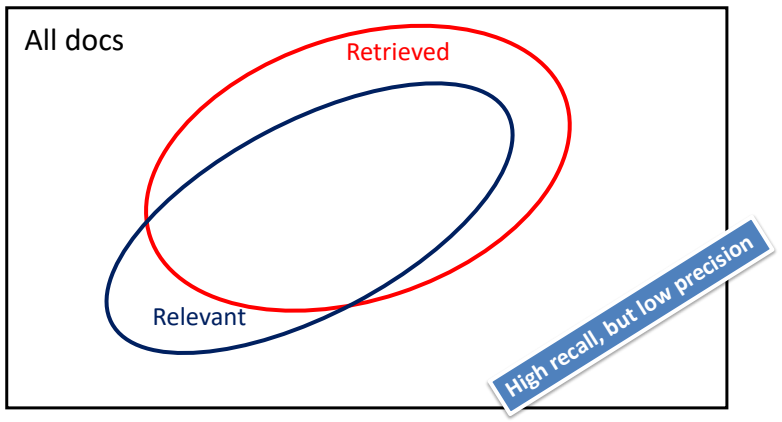


22

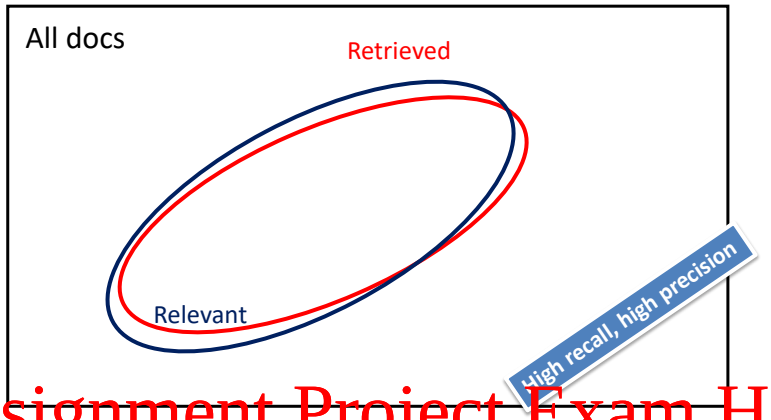
Relevant vs. Retrieved



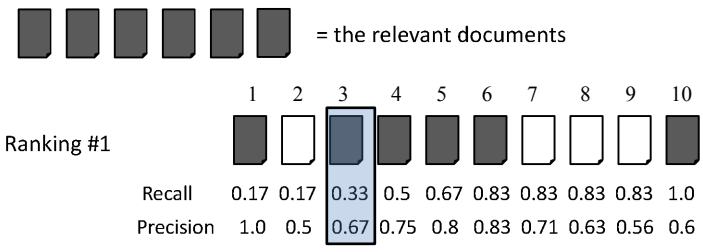
Relevant vs. Retrieved



Relevant vs. Retrieved



Ranking Effectiveness (1)



Precision = 2/3 = 0.67

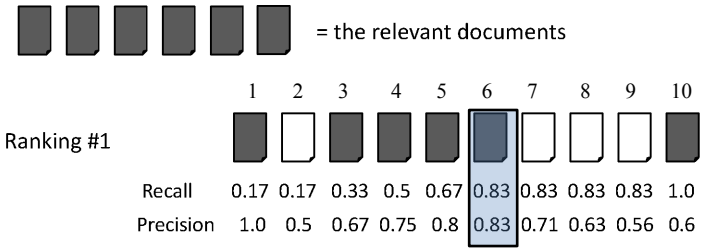
Recall = 2/6 = 0.33

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

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

26

Ranking Effectiveness (2)



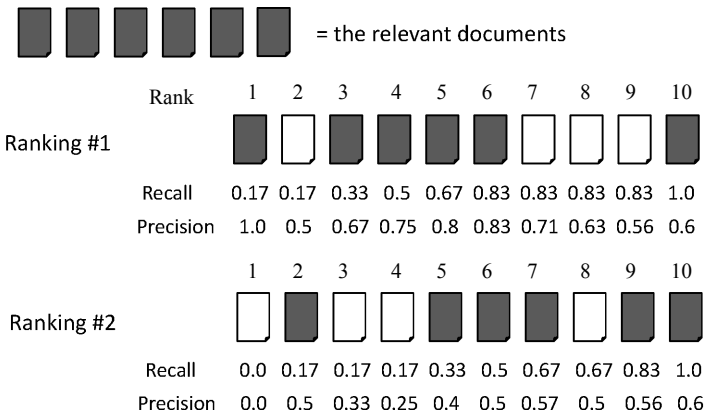
Precision = 5/6 = 0.83

Recall = 5/6 = 0.83

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

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

Precision @ Rank R



- Popular metric in Web search
- Precision @ Rank 10 of both rankings is the same (0.6)
- If Precision @ rank R is higher for Ranking 1 than for Ranking 2, recall will also be higher

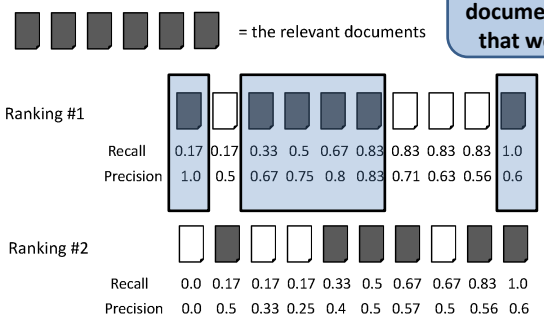
Summarising a Ranking

- Method 1: Calculating recall and precision at fixed rank positions (Precision @ Rank R)
- Method 2: Calculating precision at standard recall levels, from 0.0 to 1.0
  - Requires *interpolation* *(discussed later)*
- Method 3: **Average Precision**
  - Averaging the precision values from the rank positions where a relevant document was retrieved
  - Set precision values to be zero for the not retrieved documents

29

Average Precision (1)

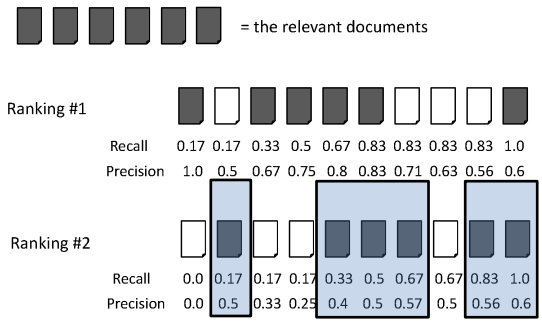
NB: 6 is the TOTAL number of relevant documents, not just those that we have observed.



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

30

Average Precision (2)

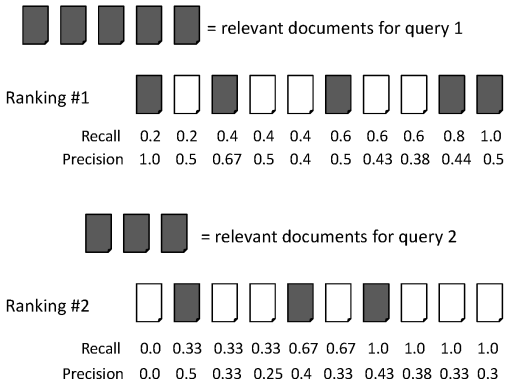


Ranking #1: $(1.0 + 0.67 + 0.75 + 0.8 + 0.83 + 0.6)/6 = 0.78$

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

31

Mean Average Precision (MAP)



$average\ precision\ query\ 1 = (1.0 + 0.67 + 0.5 + 0.44 + 0.5)/5 = 0.62$
$average\ precision\ query\ 2 = (0.5 + 0.4 + 0.43)/3 = 0.44$

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

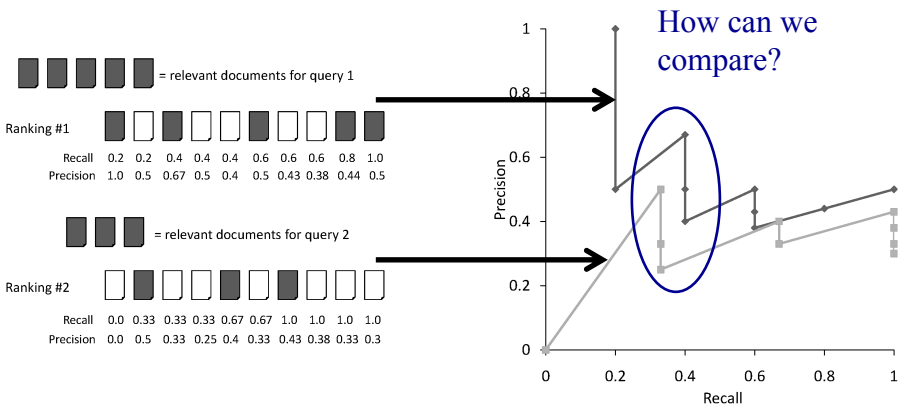
32

Mean Average Precision (MAP)

- Summarise rankings from multiple queries by averaging average precision
- A widely used measure in research papers
- Assumes user is interested in finding **many** relevant documents for each query
- Requires **many** relevance judgments in test collection
- Recall-precision graphs are also useful summaries

33

Recall-Precision Graph



34

Interpolation

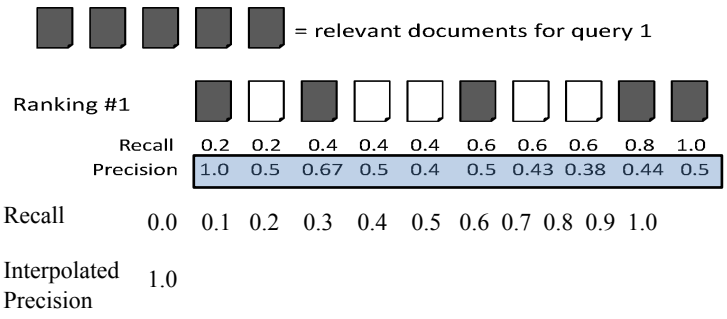
- To average graphs, calculate precision at standard recall levels:

$$P(R) = \max\{P' : R' \geq R \wedge (R', P') \in S\}$$

  - where S is the set of observed (R,P) points
- Defines precision at any recall level as the *maximum* precision observed in any recall-precision point at a higher recall level
  - Produces a step function
  - Defines precision at recall 0.0

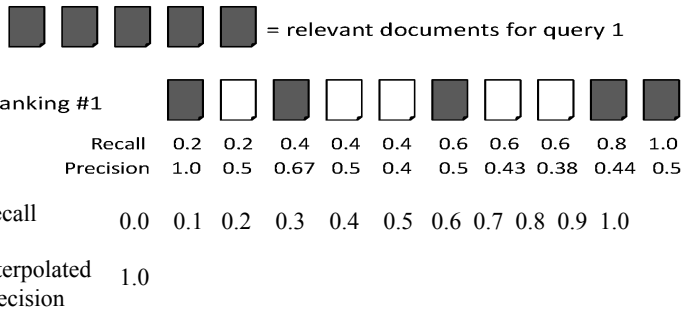
35

Interpolation



36

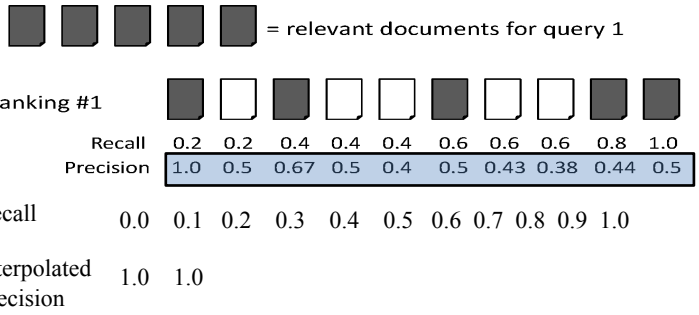
Interpolation



Recall	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Interpolated Precision	1.0										

37

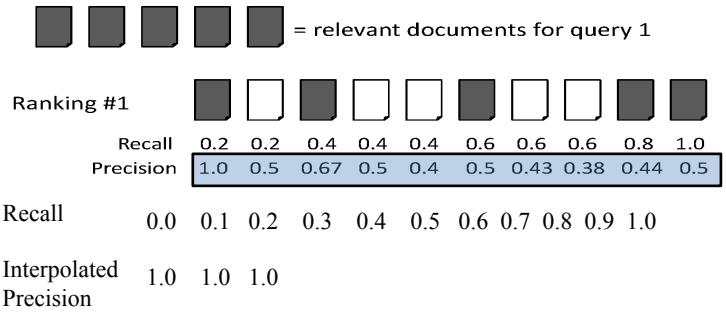
Interpolation



Recall	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Interpolated Precision	1.0	1.0									

38

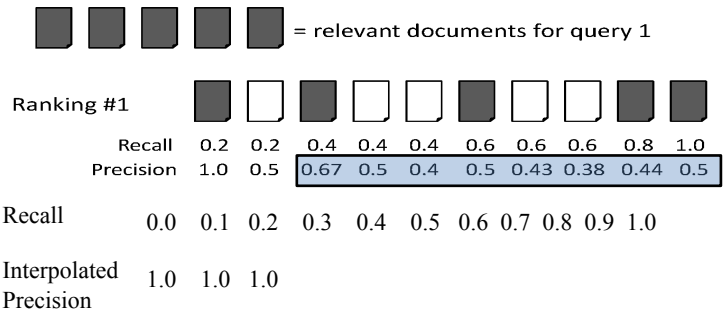
Interpolation



Recall	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Interpolated Precision	1.0	1.0	1.0								

39

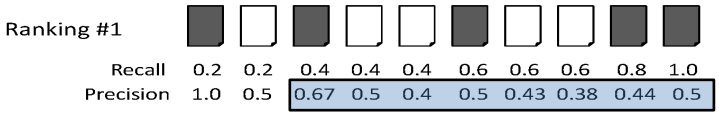
Interpolation



Recall	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Interpolated Precision	1.0	1.0	1.0								

40

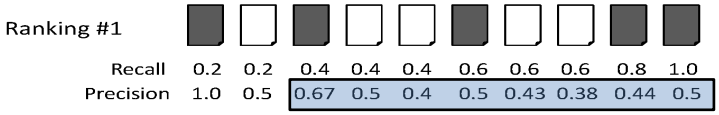
Interpolation



Recall	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Interpolated Precision	1.0	1.0	1.0	0.67							

41

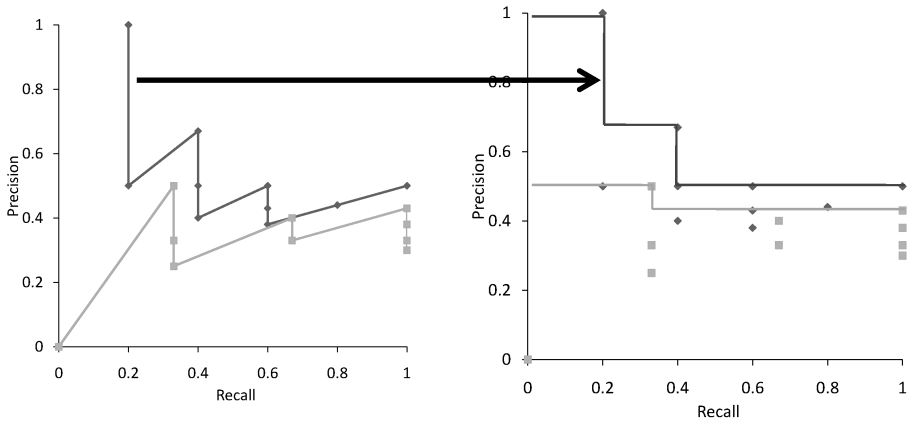
Interpolation



Recall	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Interpolated Precision	1.0	1.0	1.0	0.67	0.67	0.5	0.5	0.5	0.5	0.5	0.5

42

Interpolation



43

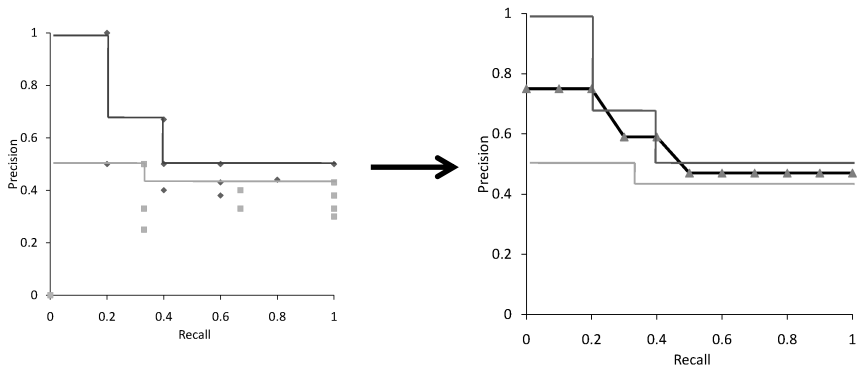
Average Precision at Standard Recall Levels

Recall	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Ranking 1	1.0	1.0	1.0	0.67	0.67	0.5	0.5	0.5	0.5	0.5	0.5
Ranking 2	0.5	0.5	0.5	0.5	0.43	0.43	0.43	0.43	0.43	0.43	0.43
Average	0.75	0.75	0.75	0.59	0.55	0.47	0.47	0.47	0.47	0.47	0.47

- Only consider **standard recall levels**: varying from 0.0 to 1.0 at the incremental of 0.1
- **Recall-precision graph** plotted by simply joining the average precision points at the standard recall levels

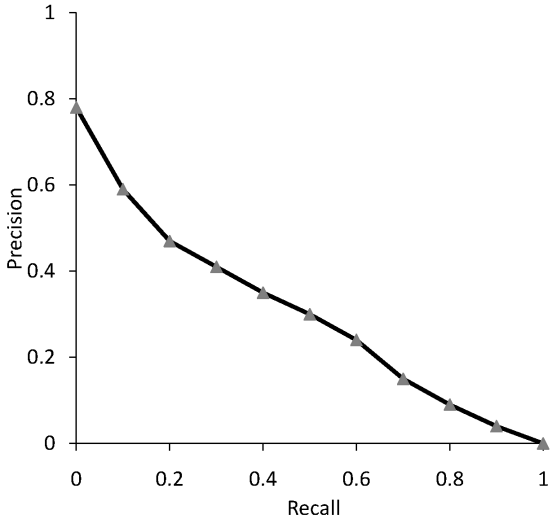
44

Average Recall-Precision Graph



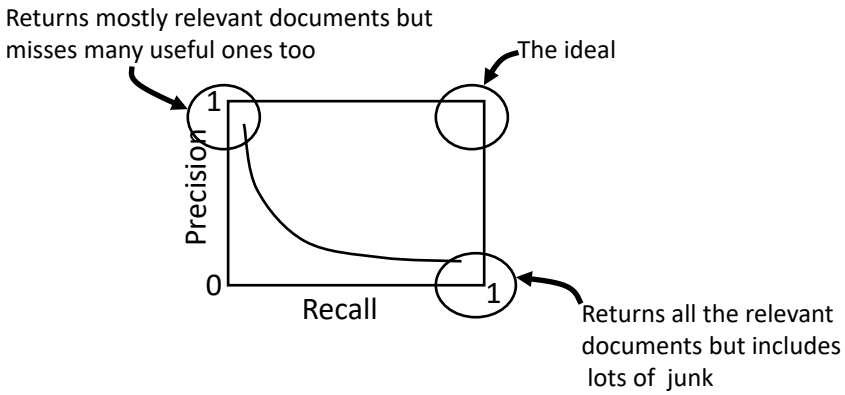
45

Graph for 50 Queries



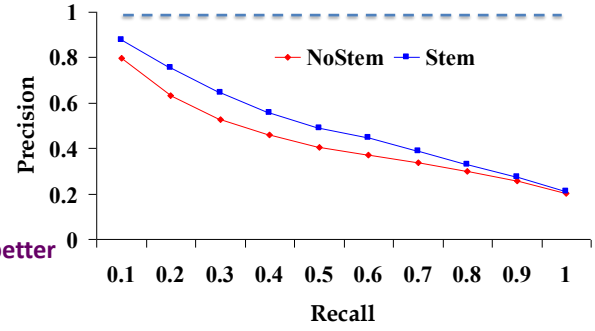
46

Trade-off between Recall and Precision



Comparing Two or More Systems

- The curve closest to the upper right-hand corner of the graph indicates the best performance



It seems that stemming is better than no stemming,
esp. towards the top of the ranking (i.e. where recall is low)

48

Van Rijsbergen's F Measure

- ***Harmonic mean*** of recall and precision

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

  - Why harmonic mean?
  - Harmonic mean emphasises the importance of small values, whereas the arithmetic mean is affected more by outliers that are unusually large

- More general form

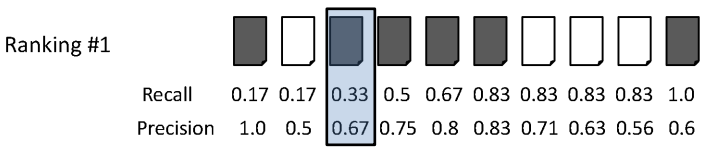
$$F_\beta = (\beta^2+1)RP/(R+\beta^2 P)$$

  - β is a parameter that determines relative importance of recall and precision

49

F Measure Example (1)

= the relevant documents



Recall = 2/6 = 0.33

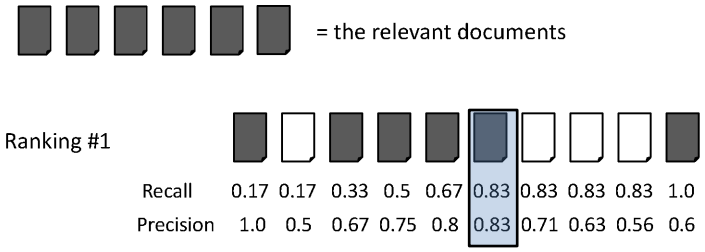
Precision = 2/3 = 0.67

F = 2*Recall*Precision/(Recall + Precision)

= 2*0.33*0.67/(0.33 + 0.67) = 0.22

50

F Measure Example (2)



Recall = 5/6 = 0.83

Precision = 5/6 = 0.83

F = 2*Recall*Precision/(Recall + Precision)

= 2*0.83*0.83/(0.83 + 0.83) = 0.83

51

Problems with Precision/Recall

- Can't know true recall value
  - Except in small collections
  - See Pooling technique
- Precision/Recall are related
  - A combined measure is sometimes more appropriate
- Does not take into account degree of relevance
- Assumes batch mode
  - Interactive IR is important and has different criteria for successful searches (more on this in a later lecture)

Discuss some of the issues with P;R

52

Focusing on Top Documents

- Users tend to look at only the top part of the ranked result list to find relevant documents
- Some search tasks have only one relevant document
  - e.g., navigational search, question answering
- Recall not appropriate
  - Instead need to measure how well the search engine does at retrieving relevant documents at very high ranks

53

Focusing on Top Documents

- Precision at Rank R (P@5, P@10, etc.)
  - R typically 5, 10, 20
  - Easy to compute, average, understand
  - Not sensitive to rank positions less than R
- Reciprocal Rank
  - Reciprocal of the rank at which the ***first*** relevant document is retrieved
  - *Mean Reciprocal Rank (MRR)* is the average of the reciprocal ranks over a set of queries
  - Very sensitive to rank position

54

Mean Reciprocal Rank (MRR)

55

Discounted Cumulative Gain (DCG)

- Popular measure for evaluating web search and related tasks
  - Use graded relevance
- Two assumptions:
  - Highly relevant documents are more useful than marginally relevant documents
  - The lower the ranked position of a relevant document, the less useful it is for the user, since it is less likely to be examined

56

Discounted Cumulative Gain

- Uses *graded relevance* as a measure of the usefulness, or *gain,* from examining a document
- Gain is accumulated starting at the top of the ranking and is *discounted* at lower ranks
  - Typical discount is 1/*log (rank)*
  - With base 2, the discount at rank 4 is 1/2, and at rank 8 it is 1/3

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$$

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log(1+i)}$$

57

DCG Example

- 10 ranked documents judged on 0-3 relevance scale:

  3, 2, 3, 0, 0, 1, 2, 2, 3, 0
- Discounted gain:

  3, 2/1, 3/1.59, 0, 0, 1/2.59, 2/2.81, 2/3, 3/3.17, 0

  = 3, 2, 1.89, 0, 0, 0.39, 0.71, 0.67, 0.95, 0
- DCG:

  3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61

58

Normalized DCG

- DCG numbers are averaged across a set of queries at specific rank values
  - e.g., DCG at rank 5 is 6.89 and at rank 10 is 9.61
- DCG values are often *normalised* by comparing the DCG at each rank with the DCG value for the *perfect ranking*
  - makes averaging easier for queries with different numbers of relevant documents

59

NDCG Example

- Perfect ranking:

  3, 3, 3, 2, 2, 2, 1, 0, 0, 0
- Ideal DCG values:

  3, 6, 7.89, 8.89, 9.75, 10.52, 10.88, 10.88, 10.88, 10.88
- NDCG values (divide actual by ideal):

  1, 0.83, 0.87, 0.76, 0.71, 0.69, 0.73, 0.8, 0.88, 0.88

  – NDCG ≤ 1 at any rank position

60

Evaluation Results

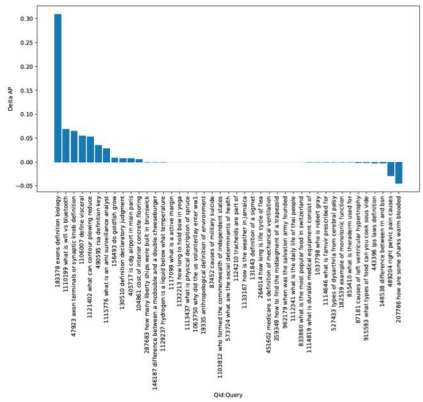
Queryid (Num): 49
Total number of documents over all queries
Retrieved: 49000
Relevant: 1670
Rel_ret: 1258
Interpolated Recall - Precision Averages:
at 0.00 0.6880
at 0.10 0.5439
at 0.20 0.4773
at 0.30 0.4115
at 0.40 0.3741
at 0.50 0.3174
at 0.60 0.2405
at 0.70 0.1972
at 0.80 0.1721
at 0.90 0.1337
at 1.00 0.1113
Average precision (non-interpolated) for all rel docs(averaged over queries)
0.3160

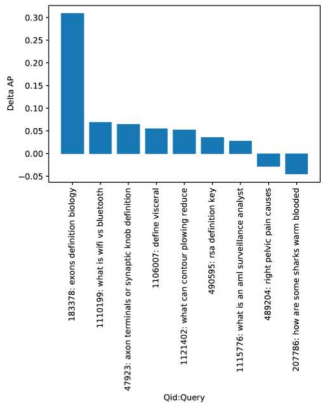
Precision:
At 5 docs: 0.3837
At 10 docs: 0.3408
At 15 docs: 0.3102
At 20 docs: 0.2806
At 30 docs: 0.2422
At 100 docs: 0.1365
At 200 docs: 0.0883
At 500 docs: 0.0446
At 1000 docs: 0.0257
R-Precision (precision after R (= num_rel for a query) docs retrieved):
Exact: 0.3068

Evaluation tools produce many measures (MAP, NDCG, P@R, etc): e.g. trec_eval is the NIST, TREC standard

Analysing Results

- Quantitative evaluation (P@5, MAP, etc) gives overall performance
- We also introduced recall-precision graphs as a mechanism to analyse the query results
- **Per-query evaluation** results wrt a baseline are also very useful
  - How many queries were **improved**/**degraded** with respect to a baseline?
  - How per-query performance changes wrt baseline (**per-query bar chart**) - are there recognisable patterns about what queries are improved or not
  - Complemented with **statistical significance testing** (discussed later)





62

Significance Tests

- Given the results from a number of queries, how can we conclude that ranking algorithm A is **better** than algorithm B?

- A significance test
  - *Null hypothesis:* no difference between A and B
  - *Alternative hypothesis:* B is better than A
  - The *power* of a test is the probability that the test will reject the null hypothesis correctly
  - Increasing the number of queries in the experiment also increases power of a test

63

Example Experimental Results

Query	A	B	B-A
1	25	35	10
2	43	84	41
3	39	15	-24
4	75	75	0
5	43	68	25
6	15	85	70
7	20	80	60
8	52	50	-2
9	49	58	9
10	50	75	25

Significance level: $\alpha = 0.05$

Is System B better than System A (**baseline**)?

64

Example Experimental Results

Query	A	B	B-A
1	25	35	10
2	43	84	41
3	39	15	-24
4	75	75	0
5	43	68	25
6	15	85	70
7	20	80	60
8	52	50	-2
9	49	58	9
10	50	75	25
Avg	41.1	62.5	

t-test $t = \frac{\overline{B-A}}{\sigma_{B-A}} \cdot \sqrt{N}$

t = 2.33 p-value = 0.02

p-value= 0.02 < 0.05

→ **Reject null hypothesis and conclude that B is better than A**

Significance level: α = 0.05

Is System B better than System A (**baseline**)?

65

Summary

- No single measure is the correct one for any application
  - Choose measures appropriate for task
  - Use a combination
  - Shows different aspects of the system effectiveness
- Analyse performance of individual queries
  - Number of queries that improved/degraded relative to a baseline
  - Type of queries that benefited, etc
- Use significance tests (t-test)

66