Lecture 10.
Retrieval Evaluation

Introduction

- To evaluate an IR system is to measure how well the system meets the information needs of the users
  - This is troublesome, given that a same result set might be interpreted differently by distinct users
  - To deal with this problem, some metrics have been defined that, on average, have a correlation with the preferences of a group of users
- Without proper *retrieval evaluation*, one cannot
  - determine how well the IR system is performing
  - compare the performance of the IR system with that of other systems, objectively
- **Retrieval evaluation** is a critical and integral component of any modern IR system

Introduction

- Systematic evaluation of the IR system allows answering questions such as:
  - a modification to the ranking function is proposed, should we go ahead and launch it?
  - a new probabilistic ranking function has just been devised, is it superior to the vector model and BM25 rankings?
  - for which types of queries, such as business, product, and geographic queries, a given ranking modification works best?
- Lack of evaluation prevents answering these questions and precludes fine tunning of the ranking function

Introduction

- *Retrieval performance evaluation* consists of associating a quantitative metric to the results produced by an IR system
  - This metric should be directly associated with the relevance of the results to the user
  - Usually, its computation requires comparing the results produced by the system with results suggested by humans for a same set of queries

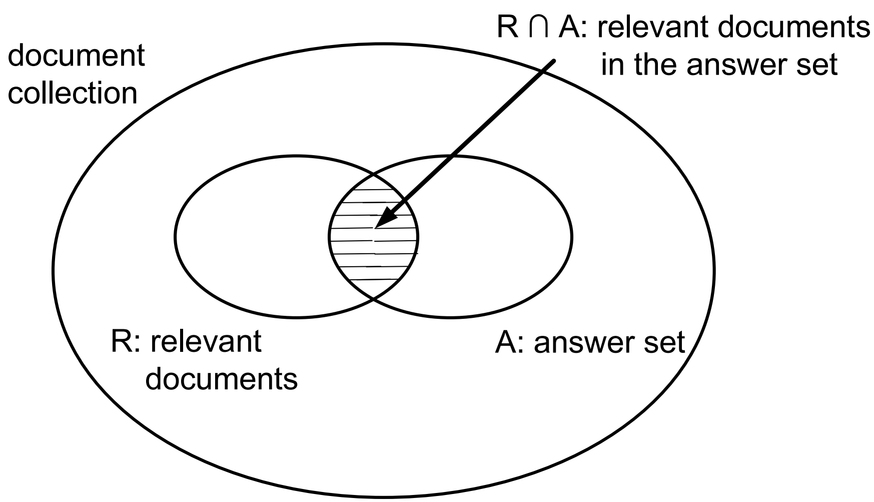
Reference Collections

- Reference collections, which are based on the foundations established by the Cranfield experiments, constitute the most used evaluation method in IR
- A reference collection is composed of:
  - A set $\mathcal{D}$ of pre-selected documents
  - A set I of information need descriptions used for testing
  - A set of relevance judgements associated with each pair $[i_m, d_j]$, $i_m \in \mathcal{I}$ and $d_j \in \mathcal{D}$
- The relevance judgement has a value of 0 if document d_j is non-relevant to i_m, and 1 otherwise
- These judgements are produced by human specialists

Precision and Recall

Precision and Recall

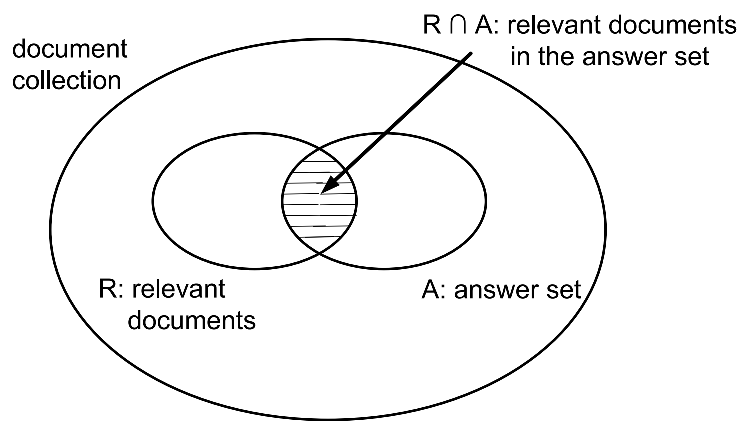
- Consider,
  - I: an information request
  - R: the set of relevant documents for I
  - A: the answer set for I, generated by an IR system
  - $R \cap A$: the intersection of the sets R and A



Precision and Recall

- The recall and precision measures are defined as follows
  - **Recall** is the fraction of the relevant documents (the set R) which has been retrieved i.e.,

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



  - **Precision** is the fraction of the retrieved documents (the set A) which is relevant i.e.,

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

Precision and Recall

- The definition of precision and recall assumes that all docs in the set A have been examined
- However, the user is not usually presented with all docs in the answer set A at once
  - User sees a ranked set of documents and examines them starting from the top
- Thus, precision and recall vary as the user proceeds with their examination of the set A
- Most appropriate then is to plot a **curve of precision versus recall**

Precision and Recall

- Consider a reference collection and a set of test queries
- Let R_{q_1} be the set of relevant docs for a query q_1:
  - $R_{q_1} = \{d_3, d_5, d_9, d_{25}, d_{39}, d_{44}, d_{56}, d_{71}, d_{89}, d_{123}\}$
- Consider a new IR algorithm that yields the following answer to q_1 (relevant docs are marked with a bullet):

(100%,10%)	01. d_{123} •	(50%,30%) 06. d_9 •	11. d_{38}
	02. d_{84}	07. d_{511}	12. d_{48}
(66%,20%)	03. d_{56} •	08. d_{129}	13. d_{250}
	04. d_6	09. d_{187}	14. d_{113}
	05. d_8	10. d_{25} • (40%,40%)	15. d_3 • (33%,50%)

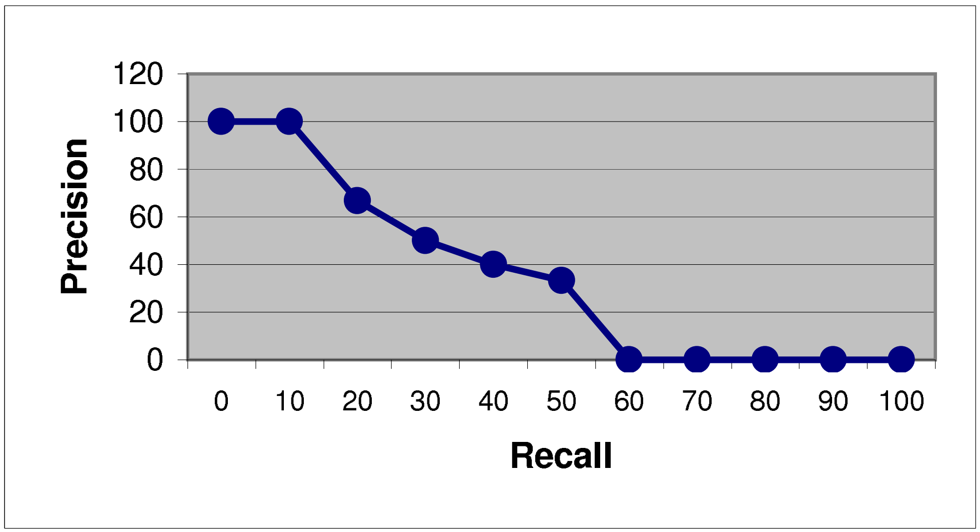
Precision and Recall

- If we examine this ranking, we observe that
  - The document d_{123}, ranked as number 1, is relevant
    - This document corresponds to 10% of all relevant documents
    - Thus, we say that we have a precision of 100% at 10% recall
  - The document d_{56}, ranked as number 3, is the next relevant
    - At this point, two documents out of three are relevant, and two of the ten relevant documents have been seen
    - Thus, we say that we have a precision of 66.6% at 20% recall

01. d_{123} $\bullet$	06. d_9 $\bullet$	11. d_{38}
02. d_{84}	07. d_{511}	12. d_{48}
03. d_{56} $\bullet$	08. d_{129}	13. d_{250}
04. d_6	09. d_{187}	14. d_{113}
05. d_8	10. d_{25} $\bullet$	15. d_3 $\bullet$

Precision and Recall

- If we proceed with our examination of the ranking generated, we can plot a curve of precision versus recall as follows:



Recall	Precision
0	100
10	100
20	66.6
30	50
40	40
50	33.3
60	0
70	0
80	0
90	0
100	0

Precision and Recall

- Consider now a second query q_2 whose set of relevant answers is given by

$$R_{q_2} = \{d_3, d_{56}, d_{129}\}$$

- The previous IR algorithm processes the query q_2 and returns a ranking, as follows

		(25%,66.6%)	
	01. d_{425}	06. d_{615}	11. d_{193}
	02. d_{87}	07. d_{512}	12. d_{715}
(33.3%,33.3%)	03. d_{56} $\bullet$	08. d_{129} $\bullet$	13. d_{810}
	04. d_{32}	09. d_{4}	14. d_{5}
	05. d_{124}	10. d_{130}	15. d_{3} $\bullet$
			(20%,100%)

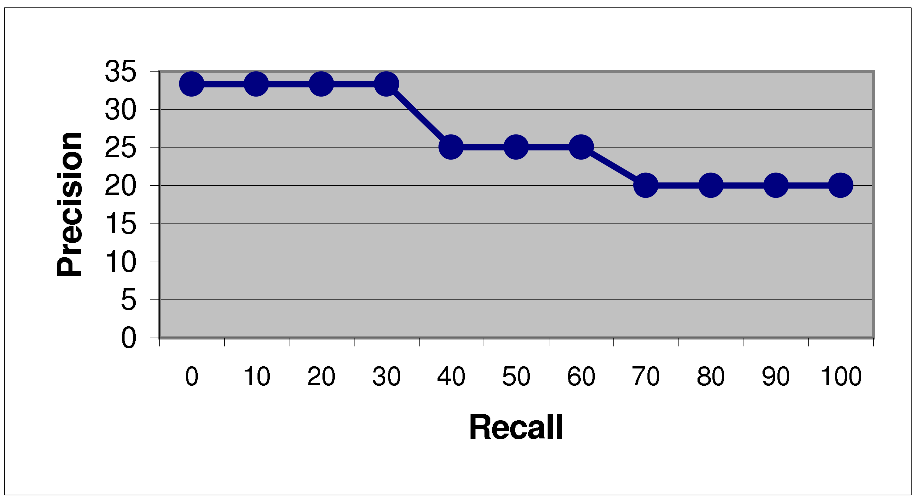
Precision and Recall

- If we examine this ranking, we observe
  - The first relevant document is d_{56}
    - It provides a recall and precision levels equal to 33.3%
  - The second relevant document is d_{129}
    - It provides a recall level of 66.6% (with precision equal to 25%)
  - The third relevant document is d_3
    - It provides a recall level of 100% (with precision equal to 20%)

01. d_{425}	06. d_{615}	11. d_{193}
02. d_{87}	07. d_{512}	12. d_{715}
03. d_{56} •	08. d_{129} •	13. d_{810}
04. d_{32}	09. d_4	14. d_5
05. d_{124}	10. d_{130}	15. d_3 •

Precision and Recall

- The precision figures at the 11 standard recall levels are interpolated as follows
- Let r_j, $j \in \{0, 1, 2, \ldots, 10\}$, be a reference to the j-th standard recall level
- Then, $\boxed{P(r_j) = max_{\forall r \mid r_j \leq r} \; P(r)}$
- In our last example, this interpolation rule yields the precision and recall figures illustrated below



Recall	Precision
0	33.3
10	33.3
20	33.3
30	33.3
40	25
50	25
60	25
70	20
80	20
90	20
100	20

Precision and Recall

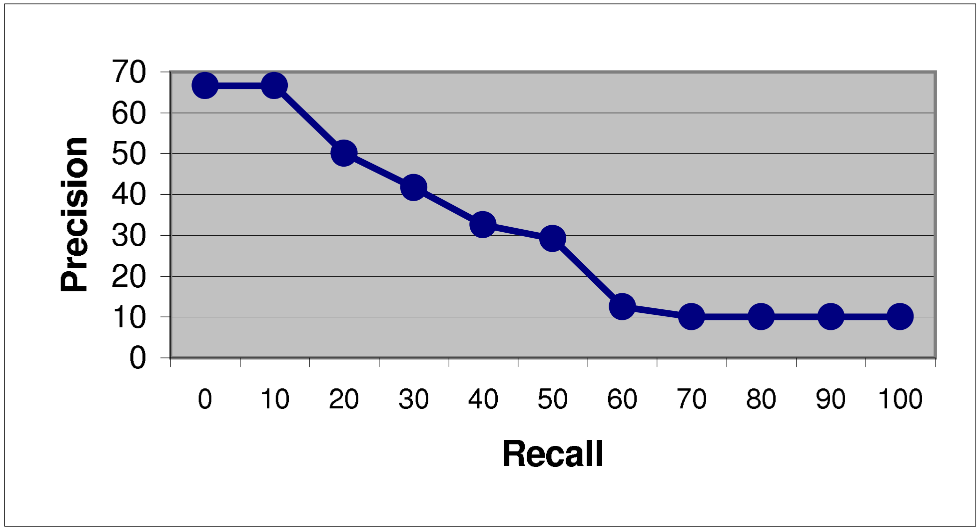
- In the examples above, the precision and recall figures have been computed for single queries
- Usually, however, retrieval algorithms are evaluated by running them for several distinct test queries
- To evaluate the retrieval performance for N_q queries, we average the precision at each recall level as follows

$$\overline{P}(r_j) = \sum_{i=1}^{N_q} \frac{P_i(r_j)}{N_q}$$

- where
  - $\overline{P}(r_j)$ is the average precision at the recall level r_j
  - $P_i(r_j)$ is the precision at recall level r_j for the i-th query

Precision and Recall

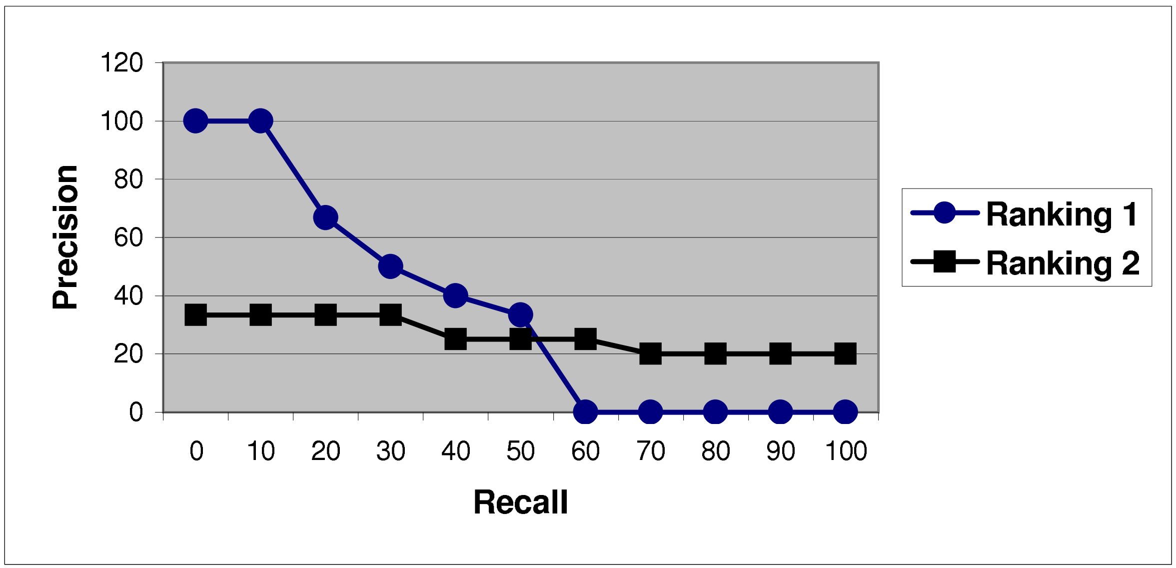
- To illustrate, the figure below illustrates precision-recall figures averaged over queries q_1 and q_2



Recall	Precision
0	66.6
10	66.6
20	49.9
30	41.6
40	32.5
50	29.1
60	12.5
70	10
80	10
90	10
100	10

Precision and Recall

- Average precision-recall curves are normally used to compare the performance of distinct IR algorithms
- The figure below illustrates average precision-recall curves for two distinct retrieval algorithms



Precision-Recall Appropriateness

- Precision and recall have been extensively used to evaluate the retrieval performance of IR algorithms
- However, a more careful reflection reveals problems with these two measures:
  - First, the proper estimation of maximum recall for a query requires detailed knowledge of all the documents in the collection
  - Second, in many situations the use of a single measure could be more appropriate
  - Third, recall and precision measure the effectiveness over a set of queries processed in batch mode
  - Fourth, for systems which require a weak ordering though, recall and precision might be inadequate

Single Value Summaries

- Average precision-recall curves constitute standard evaluation metrics for information retrieval systems
- However, there are situations in which we would like to evaluate retrieval performance over individual queries
- The reasons are twofold:
  - First, averaging precision over many queries might disguise important anomalies in the retrieval algorithms under study
  - Second, we might be interested in investigating whether a algorithm outperforms the other for each query
- In these situations, a single precision value can be used

P@5 and *P@10*

- In the case of Web search engines, the majority of searches does not require high recall
- Higher the number of relevant documents at the top of the ranking, more positive is the impression of the users
- Precision at 5 (*P@5*) and at 10 (*P@10*) measure the precision when 5 or 10 documents have been seen
- These metrics assess whether the users are getting relevant documents at the top of the ranking or not

P@5 and *P@10*

- To exemplify, consider again the ranking for the example query q_1 we have been using:

01. d_{123} $\bullet$	06. d_9 $\bullet$	11. d_{38}
02. d_{84}	07. d_{511}	12. d_{48}
03. d_{56} $\bullet$	08. d_{129}	13. d_{250}
04. d_6	09. d_{187}	14. d_{113}
05. d_8	10. d_{25} $\bullet$	15. d_3 $\bullet$

- For this query, we have *P@5* $= 40\%$ and *P@10* $= 40\%$
- Further, we can compute *P@5* and *P@10* averaged over a sample of 100 queries, for instance
- These metrics provide an early assessment of which algorithm might be preferable in the eyes of the users

MAP: Mean Average Precision

Definition

Let *Ri* refer to the set of relevant documents for query *qi*, and |*Ri*| refer to its size. Let *Ri*[*k*] be a reference to the *k*-th document in *Ri*. Then, *P*(*Ri*[*k*]) is the precision when the *Ri*[*k*] document is observed in the ranking of *qi*. If that document is never retrieved, *P*(*Ri*[*k*]) is taken as zero.

MAPi is the mean value precision for *qi*.

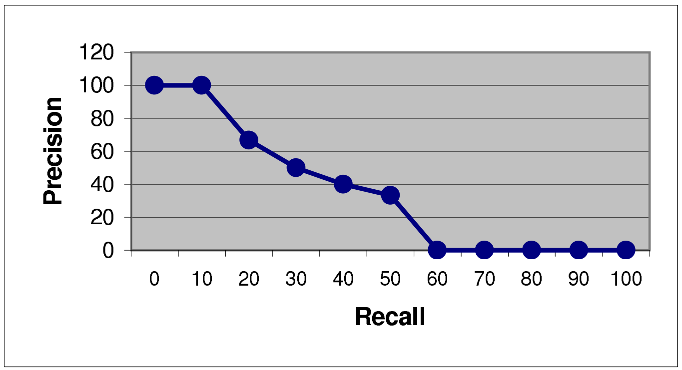
MAP is the mean value precision over a set of queries.

$$R_{q_1} = \{d_3, d_5, d_9, d_{25}, d_{39}, d_{44}, d_{56}, d_{71}, d_{89}, d_{123}\}$$

		(50%,30%)	
(100%,10%)	01. d_{123} •	06. d_9 •	11. d_{38}
	02. d_{84}	07. d_{511}	12. d_{48}
(66%,20%)	03. d_{56} •	08. d_{129}	13. d_{250}
	04. d_6	09. d_{187}	14. d_{113}
	05. d_8	10. d_{25} •	15. d_3 •
		(40%,40%)	(33%,50%)

MAP: Mean Average Precision

- The idea here is to average the precision figures obtained after each new relevant document is observed
  - For relevant documents not retrieved, the precision is set to 0
- To illustrate, consider again the precision-recall curve for the example query q_1



Recall	Precision
0	100
10	100
20	66.6
30	50
40	40
50	33.3
60	0
70	0
80	0
90	0
100	0

- The mean average precision (MAP) for q_1 is given by

$$MAP_1 = \frac{1 + 0.66 + 0.5 + 0.4 + 0.33 + 0 + 0 + 0 + 0 + 0}{10} = 0.28$$

MAP: Mean Average Precision

$R_{q_2} = \{d_3, d_{56}, d_{129}\}$

		(25%,66.6%)	
	01. d_{425}	06. d_{615}	11. d_{193}
	02. d_{87}	07. d_{512}	12. d_{715}
(33.3%,33.3%)	03. d_{56} •	08. d_{129} •	13. d_{810}
	04. d_{32}	09. d_4	14. d_5
	05. d_{124}	10. d_{130}	15. d_3 •
			(20%,100%)

MAP2= (0.33+0.25+0.20)=0.26

MAP=(*MAP*1+*MAP*2)/2=(0.28+0.26)/2=0.27

R-Precision

- Let R be the total number of relevant docs for a given query
- The idea here is to compute the precision at the R-th position in the ranking
- For the query q_1, the R value is 10 and there are 4 relevants among the top 10 documents in the ranking
- Thus, the R-Precision value for this query is 0.4
- The R-precision measure is a useful for observing the behavior of an algorithm for individual queries
- Additionally, one can also compute an average R-precision figure over a set of queries
  - However, using a single number to evaluate a algorithm over several queries might be quite imprecise

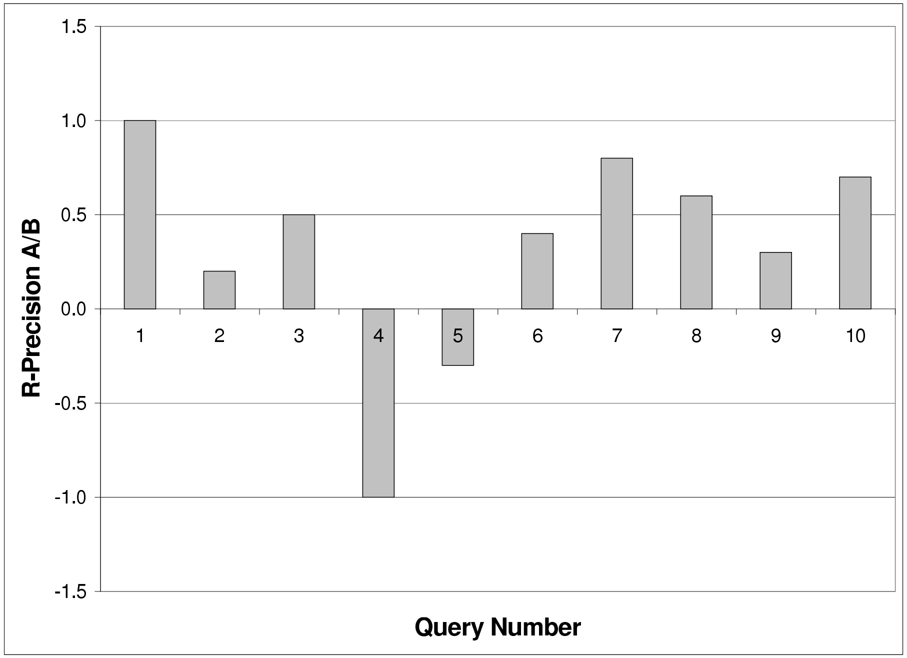
Precision Histograms

- The R-precision computed for several queries can be used to compare two algorithms as follows
- Let,
  - $RP_A(i)$: R-precision for algorithm A for the i-th query
  - $RP_B(i)$: R-precision for algorithm B for the i-th query
- Define, for instance, the difference

$$RP_{A/B}(i) = RP_A(i) - RP_B(i)$$

Precision Histograms

- Figure below illustrates the $RP_{A/B}(i)$ values for two retrieval algorithms over 10 example queries



- The algorithm A performs better for 8 of the queries, while the algorithm B performs better for the other 2 queries

MRR: Mean Reciprocal Rank

- MRR is a good metric for those cases in which we are interested in the first correct answer such as
  - Question-Answering (QA) systems
  - Search engine queries that look for specific sites
    - URL queries
    - Homepage queries

MRR: Mean Reciprocal Rank

- Let,
  - $\mathcal{R}_i$: ranking relative to a query q_i
  - $S_{correct}(\mathcal{R}_i)$: position of the first correct answer in $\mathcal{R}_i$
  - S_h: threshold for ranking position
- Then, the reciprocal rank $RR(\mathcal{R}_i)$ for query q_i is given by

$$RR(\mathcal{R}_i) = \begin{cases} \frac{1}{S_{correct}(\mathcal{R}_i)} & \text{if } S_{correct}(\mathcal{R}_i) \leq S_h \\ 0 & \text{otherwise} \end{cases}$$

- The mean reciprocal rank (MRR) for a set Q of N_q queries is given by

$$\boxed{MRR(Q) = \sum_i^{N_q} RR(\mathcal{R}_i)}$$

The E-Measure

- A measure that combines recall and precision
- The idea is to allow the user to specify whether he is more interested in recall or in precision
- The E measure is defined as follows

$$E(j) = 1 - \frac{1+b^2}{\frac{b^2}{r(j)} + \frac{1}{P(j)}}$$

$$E = 1 - \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}}$$

$$= 1 - \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

For $\alpha = \frac{1}{\beta^2+1}$

- where
  - $r(j)$ is the recall at the j-th position in the ranking
  - $P(j)$ is the precision at the j-th position in the ranking
  - $b \geq 0$ is a user specified parameter
  - $E(j)$ is the E metric at the j-th position in the ranking

The E-Measure

- The parameter b is specified by the user and reflects the relative importance of recall and precision
- If $b = 0$

$$E = 1 - \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = 1 - \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

For $\alpha = \frac{1}{\beta^2+1}$

  - $E(j) = 1 - P(j)$
  - low values of b make $E(j)$ a function of precision
- If $b \to \infty$
  - $lim_{b\to\infty} \; E(j) = 1 - r(j)$
  - high values of b make $E(j)$ a function of recall
- For $b = 1$, the E-measure becomes the F-measure

$$F(j) = 1 - E(j)$$

- Small β increases importance of precision (For β=1 precision and recall are equally weighted)
- Small values of E mean good results

F-Measure: Harmonic Mean

- The F-measure is also a single measure that combines recall and precision

$$F(j) = \frac{2}{\frac{1}{r(j)} + \frac{1}{P(j)}}$$

where

- $r(j)$ is the recall at the j-th position in the ranking
- $P(j)$ is the precision at the j-th position in the ranking
- $F(j)$ is the harmonic mean at the j-th position in the ranking

The harmonic mean H of the positive real numbers $x_1, x_2, \ldots, x_n > 0$ is defined to be

$$H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_n}} = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}} = \frac{n \cdot \prod_{j=1}^{n} x_j}{\sum_{i=1}^{n} \frac{\prod_{j=1}^{n} x_j}{x_i}}.$$

F-Measure: Harmonic Mean

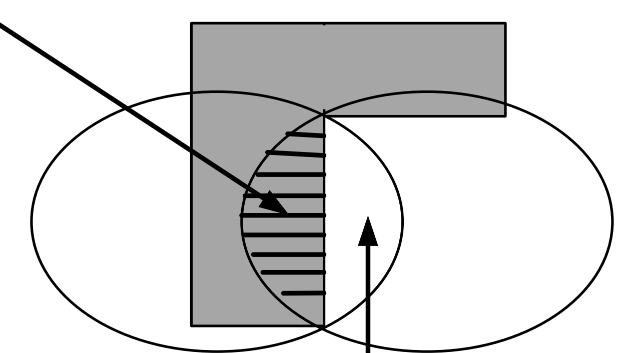
- The function F assumes values in the interval $[0, 1]$
- It is 0 when no relevant documents have been retrieved and is 1 when all ranked documents are relevant
- Further, the harmonic mean F assumes a high value only when both recall and precision are high
- To maximize F requires finding the best possible compromise between recall and precision
- Notice that setting $b = 1$ in the formula of the E-measure yields

$$F(j) = 1 - E(j)$$

User-Oriented Measures

- Recall and precision assume that the set of relevant docs for a query is independent of the users
- However, different users might have different relevance interpretations
- To cope with this problem, user-oriented measures have been proposed
- As before,
  - consider a reference collection, an information request I, and a retrieval algorithm to be evaluated
  - with regard to I, let R be the set of relevant documents and A be the set of answers retrieved

User-Oriented Measures



- K: set of documents known to the user
- $K \cap R \cap A$: set of relevant docs that have been retrieved and are known to the user
- $(R \cap A) - K$: set of relevant docs that have been retrieved but are not known to the user

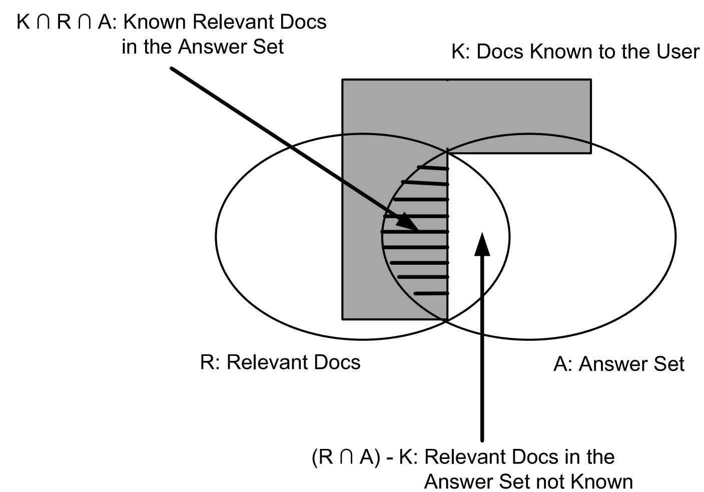
User-Oriented Measures

- The **coverage ratio** is the fraction of the documents known and relevant that are in the answer set, that is

$$coverage = \frac{|K \cap R \cap A|}{|K \cap R|}$$

- The **novelty ratio** is the fraction of the relevant docs in the answer set that are not known to the user

$$novelty = \frac{|(R \cap A) - K|}{|R \cap A|}$$



DCG — Discounted Cumulated Gain

Discounted Cumulated Gain

- Precision and recall allow only binary relevance assessments
- As a result, there is no distinction between highly relevant docs and mildly relevant docs
- These limitations can be overcome by adopting graded relevance assessments and metrics that combine them
- The **discounted cumulated gain** (DCG) is a metric that combines graded relevance assessments effectively

Discounted Cumulated Gain

- When examining the results of a query, two key observations can be made:
  - highly relevant documents are preferable at the top of the ranking than mildly relevant ones
  - relevant documents that appear at the end of the ranking are less valuable

Discounted Cumulated Gain

- Consider that the results of the queries are graded on a scale 0–3 (0 for non-relevant, 3 for strong relevant docs)
- For instance, for queries q_1 and q_2, consider that the graded relevance scores are as follows:

$$\begin{aligned} R_{q_1} &= \{ [d_3, 3], [d_5, 3], [d_9, 3], [d_{25}, 2], [d_{39}, 2], \\ &\quad [d_{44}, 2], [d_{56}, 1], [d_{71}, 1], [d_{89}, 1], [d_{123}, 1] \} \\ R_{q_2} &= \{ [d_3, 3], [d_{56}, 2], [d_{129}, 1] \} \end{aligned}$$

- That is, while document d_3 is highly relevant to query q_1, document d_{56} is just mildly relevant

Discounted Cumulated Gain

- Given these assessments, the results of a new ranking algorithm can be evaluated as follows
- Specialists associate a graded relevance score to the top 10-20 results produced for a given query q
  - This list of relevance scores is referred to as the *gain vector G*
- Considering the top 15 docs in the ranking produced for queries q_1 and q_2, the gain vectors for these queries are:

$$
\begin{aligned}
G_1 &= (1, 0, 1, 0, 0, 3, 0, 0, 0, 2, 0, 0, 0, 0, 3) \\
G_2 &= (0, 0, 2, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 3)
\end{aligned}
$$

Discounted Cumulated Gain

- By summing up the graded scores up to any point in the ranking, we obtain the cumulated gain (CG)
- For query q_1, for instance, the cumulated gain at the first position is 1, at the second position is 1+0, and so on
- Thus, the *cumulated gain vectors* for queries q_1 and q_2 are given by

$$\begin{aligned} CG_1 &= (1, 1, 2, 2, 2, 5, 5, 5, 5, 7, 7, 7, 7, 7, 10) \\ CG_2 &= (0, 0, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 6) \end{aligned}$$

- For instance, the cumulated gain at position 8 of CG_1 is equal to 5

$$\begin{aligned} G_1 &= (1, 0, 1, 0, 0, 3, 0, 0, 0, 2, 0, 0, 0, 0, 3) \\ G_2 &= (0, 0, 2, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 3) \end{aligned}$$

Discounted Cumulated Gain

- In formal terms, we define
  - Given the gain vector G_j for a test query q_j, the CG_j associated with it is defined as

$$CG_j[i] = \begin{cases} G_j[1] & \texttt{if } i = 1; \\ \\ G_j[i] + CG_j[i-1] & \texttt{otherwise} \end{cases}$$

    where $CG_j[i]$ refers to the cumulated gain at the ith position of the ranking for query q_j

Discounted Cumulated Gain

- We also introduce a discount factor that reduces the impact of the gain as we move upper in the ranking
- A simple discount factor is the logarithm of the ranking position
- If we consider logs in base 2, this discount factor will be $\log_2 2$ at position 2, $\log_2 3$ at position 3, and so on
- By dividing a gain by the corresponding discount factor, we obtain the discounted cumulated gain (DCG)

Discounted Cumulated Gain

- More formally,
  - Given the gain vector G_j for a test query q_j, the vector DCG_j associated with it is defined as

$$DCG_j[i] = \begin{cases} G_j[1] & \texttt{if } i = 1; \\ \frac{G_j[i]}{\log_2 i} + DCG_j[i-1] & \texttt{otherwise} \end{cases}$$

where $DCG_j[i]$ refers to the discounted cumulated gain at the ith position of the ranking for query q_j

Discounted Cumulated Gain

- For the example queries q_1 and q_2, the DCG vectors are given by

$$\begin{aligned} DCG_1 &= (1.0, 1.0, 1.6, 1.6, 1.6, 2.8, 2.8, 2.8, 2.8, 3.4, 3.4, 3.4, 3.4, 3.4, 4.2) \\ DCG_2 &= (0.0, 0.0, 1.3, 1.3, 1.3, 1.3, 1.3, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 2.4) \end{aligned}$$

- Discounted cumulated gains are much less affected by relevant documents at the end of the ranking
- By adopting logs in higher bases the discount factor can be accentuated

$$\begin{aligned} G_1 &= (1, 0, 1, 0, 0, 3, 0, 0, 0, 2, 0, 0, 0, 0, 3) \\ G_2 &= (0, 0, 2, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 3) \end{aligned}$$

$$\begin{aligned} CG_1 &= (1, 1, 2, 2, 2, 5, 5, 5, 5, 7, 7, 7, 7, 7, 10) \\ CG_2 &= (0, 0, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 6) \end{aligned}$$

DCG Curves

- To produce CG and DCG curves over a set of test queries, we need to average them over all queries
- Given a set of N_q queries, average $\overline{CG}[i]$ and $\overline{DCG}[i]$ over all queries are computed as follows

$$\boxed{\overline{CG}[i] = \sum_{j=1}^{N_q} \frac{CG_j[i]}{N_q}; \qquad \overline{DCG}[i] = \sum_{j=1}^{N_q} \frac{DCG_j[i]}{N_q}}$$

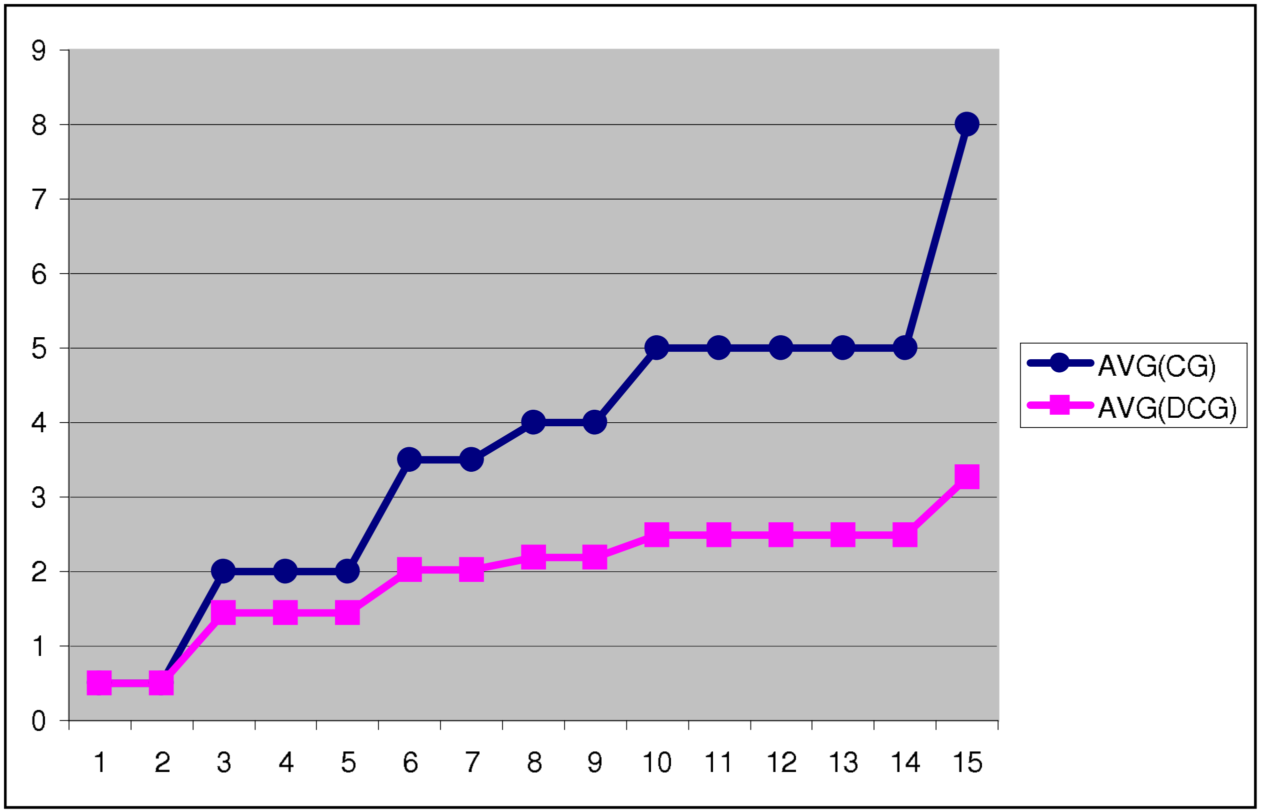
- For instance, for the example queries q_1 and q_2, these averages are given by

$$\begin{aligned}
\overline{CG} &= (0.5, 0.5, 2.0, 2.0, 2.0, 3.5, 3.5, 4.0, 4.0, 5.0, 5.0, 5.0, 5.0, 5.0, 8.0)\\
\overline{DCG} &= (0.5, 0.5, 1.5, 1.5, 1.5, 2.1, 2.1, 2.2, 2.2, 2.5, 2.5, 2.5, 2.5, 2.5, 3.3)
\end{aligned}$$

$$\begin{aligned}
CG_1 &= (1, 1, 2, 2, 2, 5, 5, 5, 5, 7, 7, 7, 7, 7, 10) & DCG_1 &= (1.0, 1.0, 1.6, 1.6, 1.6, 2.8, 2.8, 2.8, 2.8, 3.4, 3.4, 3.4, 3.4, 3.4, 4.2)\\
CG_2 &= (0, 0, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 6) & DCG_2 &= (0.0, 0.0, 1.3, 1.3, 1.3, 1.3, 1.3, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 2.4)
\end{aligned}$$

DCG Curves

- Then, average curves can be drawn by varying the rank positions from 1 to a pre-established threshold



Ideal CG and DCG Metrics

- Recall and precision figures are computed relatively to the set of relevant documents
- CG and DCG scores, as defined above, are not computed relatively to any baseline
- This implies that it might be confusing to use them directly to compare two distinct retrieval algorithms
- One solution to this problem is to define a baseline to be used for normalization
- This baseline are the ideal CG and DCG metrics, as we now discuss

Ideal CG and DCG Metrics

- For a given test query q, assume that the relevance assessments made by the specialists produced:
  - n_3 documents evaluated with a relevance score of 3
  - n_2 documents evaluated with a relevance score of 2
  - n_1 documents evaluated with a score of 1
  - n_0 documents evaluated with a score of 0
- The ideal gain vector IG is created by sorting all relevance scores in decreasing order, as follows:

$$IG = (3, \ \ldots, \ 3, \ 2, \ \ldots, \ 2, \ 1, \ \ldots, \ 1, \ 0, \ \ldots, \ 0)$$

- For instance, for the example queries q_1 and q_2, we have

$$\begin{aligned} IG_1 &= (3,3,3,2,2,2,1,1,1,1,0,0,0,0,0) \\ IG_2 &= (3,2,1,0,0,0,0,0,0,0,0,0,0,0,0) \end{aligned}$$

Ideal CG and DCG Metrics

- Ideal CG and ideal DCG vectors can be computed analogously to the computations of CG and DCG
- For the example queries q_1 and q_2, we have

$$\begin{aligned} ICG_1 &= (3, 6, 9, 11, 13, 15, 16, 17, 18, 19, 19, 19, 19, 19, 19) \\ ICG_2 &= (3, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6) \end{aligned}$$

- The ideal DCG vectors are given by

$$\begin{aligned} IDCG_1 &= (3.0, 6.0, 7.9, 8.9, 9.8, 10.5, 10.9, 11.2, 11.5, 11.8, 11.8, 11.8, 11.8, 11.8, 11.8) \\ IDCG_2 &= (3.0, 5.0, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6) \end{aligned}$$

$$\begin{aligned} IG_1 &= (3, 3, 3, 2, 2, 2, 1, 1, 1, 1, 0, 0, 0, 0, 0) \\ IG_2 &= (3, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) \end{aligned}$$

Ideal CG and DCG Metrics

- Further, average $\overline{ICG}$ and average $\overline{IDCG}$ scores can be computed as follows

$$\boxed{\overline{ICG}[i] = \sum_{j=1}^{N_q} \frac{ICG_j[i]}{N_q}; \qquad \overline{IDCG}[i] = \sum_{j=1}^{N_q} \frac{IDCG_j[i]}{N_q}}$$

- For instance, for the example queries q_1 and q_2, $\overline{ICG}$ and $\overline{IDCG}$ vectors are given by

$$\begin{aligned}
\overline{ICG} &= (3.0, 5.5, 7.5, 8.5, 9.5, 10.5, 11.0, 11.5, 12.0, 12.5, 12.5, 12.5, 12.5, 12.5, 12.5) \\
\overline{IDCG} &= (3.0, 5.5, 6.8, 7.3, 7.7, 8.1, 8.3, 8.4, 8.6, 8.7, 8.7, 8.7, 8.7, 8.7, 8.7)
\end{aligned}$$

- By comparing the average CG and DCG curves for an algorithm with the average ideal curves, we gain insight on how much room for improvement there is

$$\begin{aligned}
ICG_1 &= (3, 6, 9, 11, 13, 15, 16, 17, 18, 19, 19, 19, 19, 19, 19) \\
ICG_2 &= (3, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6)
\end{aligned}$$

$$\begin{aligned}
IDCG_1 &= (3.0, 6.0, 7.9, 8.9, 9.8, 10.5, 10.9, 11.2, 11.5, 11.8, 11.8, 11.8, 11.8, 11.8, 11.8) \\
IDCG_2 &= (3.0, 5.0, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6)
\end{aligned}$$

Normalized DCG

- Precision and recall figures can be directly compared to the ideal curve of 100% precision at all recall levels
- DCG figures, however, are not build relative to any ideal curve, which makes it difficult to compare directly DCG curves for two distinct ranking algorithms
- This can be corrected by normalizing the DCG metric
- Given a set of N_q test queries, normalized CG and DCG metrics are given by

$$\boxed{NCG[i] = \frac{\overline{CG}[i]}{\overline{ICG}[i]}; \qquad NDCG[i] = \frac{\overline{DCG}[i]}{\overline{IDCG}[i]}}$$

Normalized DCG

- For instance, for the example queries q_1 and q_2, NCG and NDCG vectors are given by

$$
\begin{aligned}
NCG &= (0.17, 0.09, 0.27, 0.24, 0.21, 0.33, 0.32, \\
&\quad 0.35, 0.33, 0.40, 0.40, 0.40, 0.40, 0.40, 0.64) \\
NDCG &= (0.17, 0.09, 0.21, 0.20, 0.19, 0.25, 0.25, \\
&\quad 0.26, 0.26, 0.29, 0.29, 0.29, 0.29, 0.29, 0.38)
\end{aligned}
$$

- The area under the NCG and NDCG curves represent the quality of the ranking algorithm
- Higher the area, better the results are considered to be
- Thus, normalized figures can be used to compare two distinct ranking algorithms

Discussion on DCG Metrics

- CG and DCG metrics aim at taking into account multiple level relevance assessments
- This has the advantage of distinguishing highly relevant documents from mildly relevant ones
- The inherent disadvantages are that multiple level relevance assessments are harder and more time consuming to generate

Discussion on DCG Metrics

- Despite these inherent difficulties, the CG and DCG metrics present benefits:
  - They allow systematically combining document ranks and relevance scores
  - Cumulated gain provides a single metric of retrieval performance at any position in the ranking
  - It also stresses the gain produced by relevant docs up to a position in the ranking, which makes the metrics more imune to outliers
  - Further, discounted cumulated gain allows down weighting the impact of relevant documents found late in the ranking

Rank Correlation Metrics

Rank Correlation Metrics

- Precision and recall allow comparing the relevance of the results produced by two ranking functions
- However, there are situations in which
  - we cannot directly measure relevance
  - we are more interested in determining how differently a ranking function varies from a second one that we know well
- In these cases, we are interested in comparing the relative ordering produced by the two rankings
- This can be accomplished by using statistical functions called **rank correlation metrics**

Rank Correlation Metrics

- Let rankings $\mathcal{R}_1$ and $\mathcal{R}_2$
- A rank correlation metric yields a correlation coefficient $C(\mathcal{R}_1, \mathcal{R}_2)$ with the following properties:
  - $-1 \leq C(\mathcal{R}_1, \mathcal{R}_2) \leq 1$
  - if $C(\mathcal{R}_1, \mathcal{R}_2) = 1$, the agreement between the two rankings is perfect i.e., they are the same.
  - if $C(\mathcal{R}_1, \mathcal{R}_2) = -1$, the disagreement between the two rankings is perfect i.e., they are the reverse of each other.
  - if $C(\mathcal{R}_1, \mathcal{R}_2) = 0$, the two rankings are completely independent.
  - increasing values of $C(\mathcal{R}_1, \mathcal{R}_2)$ imply increasing agreement between the two rankings.

The Spearman Coefficient

The Spearman Coefficient

- The Spearman coefficient is likely the mostly used rank correlation metric
- It is based on the differences between the positions of a same document in two rankings
- Let
  - $s_{1,j}$ be the position of a document d_j in ranking $\mathcal{R}_1$ and
  - $s_{2,j}$ be the position of d_j in ranking $\mathcal{R}_2$

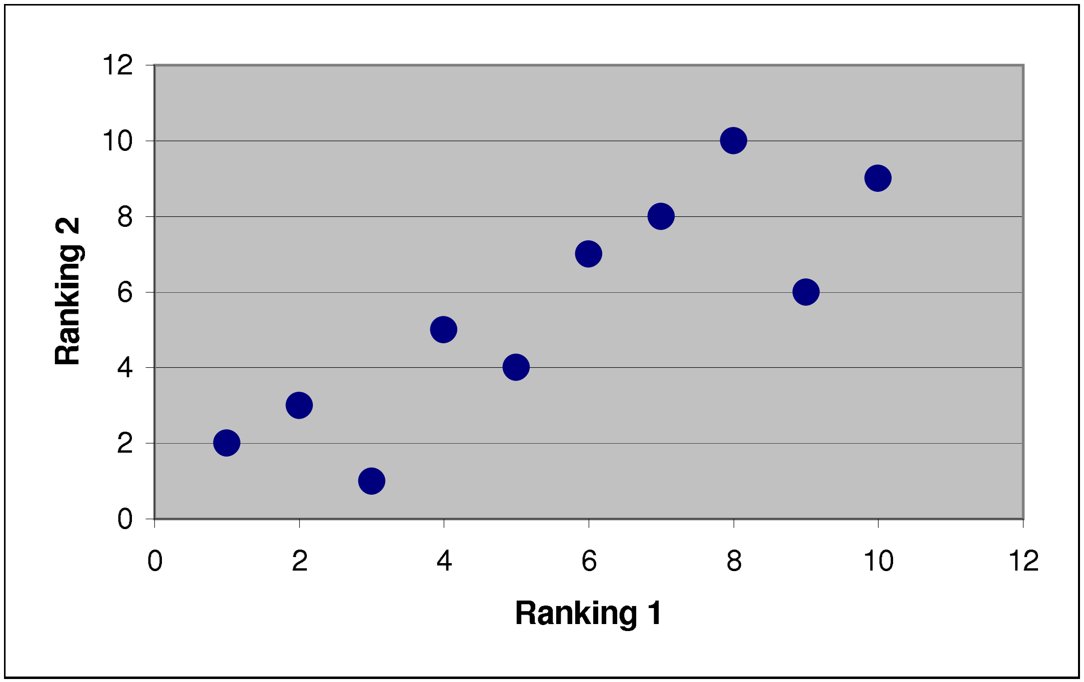
The Spearman Coefficient

- Consider 10 example documents retrieved by two distinct rankings $\mathcal{R}_1$ and $\mathcal{R}_2$. Let $s_{1,j}$ and $s_{2,j}$ be the document position in these two rankings, as follows:

documents	$s_{1,j}$	$s_{2,j}$	$s_{i,j} - s_{2,j}$	$(s_{1,j} - s_{2,j})^2$
d_{123}	1	2	-1	1
d_{84}	2	3	-1	1
d_{56}	3	1	+2	4
d_6	4	5	-1	1
d_8	5	4	+1	1
d_9	6	7	-1	1
d_{511}	7	8	-1	1
d_{129}	8	10	-2	4
d_{187}	9	6	+3	9
d_{25}	10	9	+1	1
Sum of Square Distances				24

The Spearman Coefficient

- By plotting the rank positions for $\mathcal{R}_1$ and $\mathcal{R}_2$ in a 2-dimensional coordinate system, we observe that there is a strong correlation between the two rankings



The Spearman Coefficient

- To produce a quantitative assessment of this correlation, we sum the squares of the differences for each pair of rankings
- If there are K documents ranked, the maximum value for the sum of squares of ranking differences is given by

$$\frac{K \times (K^2 - 1)}{3}$$

- Let $K = 10$
  - If the two rankings were in perfect disagreement, then this value is $(10 \times (10^2 - 1))/3$, or 330
  - On the other hand, if we have a complete agreement the sum is 0

The Spearman Coefficient

- Let us consider the fraction

$$\frac{\sum_{j=1}^{K}(s_{1,j} - s_{2,j})^2}{\frac{K \times (K^2-1)}{3}}$$

- Its value is
  - 0 when the two rankings are in perfect agreement
  - +1 when they are in perfect disagreement
- If we multiply the fraction by 2, its value shifts to the range $[0, +2]$
- If we now subtract the result from 1, the resultant value shifts to the range $[-1, +1]$

The Spearman Coefficient

- This reasoning suggests defining the correlation between the two rankings as follows
- Let $s_{1,j}$ and $s_{2,j}$ be the positions of a document d_j in two rankings $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively
- Define

$$\boxed{S(\mathcal{R}_1, \mathcal{R}_2) = 1 - \frac{6 \times \sum_{j=1}^{K} (s_{1,j} - s_{2,j})^2}{K \times (K^2 - 1)}}$$

where

  - $S(\mathcal{R}_1, \mathcal{R}_2)$ is the *Spearman rank correlation coefficient*
  - K indicates the size of the ranked sets

The Spearman Coefficient

- For the rankings in Figure below, we have

$$S(\mathcal{R}_1, \mathcal{R}_2) = 1 - \frac{6 \times 24}{10 \times (10^2 - 1)} = 1 - \frac{144}{990} = 0.854$$

documents	$s_{1,j}$	$s_{2,j}$	$s_{i,j} - s_{2,j}$	$(s_{1,j} - s_{2,j})^2$
d_{123}	1	2	-1	1
d_{84}	2	3	-1	1
d_{56}	3	1	+2	4
d_{6}	4	5	-1	1
d_{8}	5	4	+1	1
d_{9}	6	7	-1	1
d_{511}	7	8	-1	1
d_{129}	8	10	-2	4
d_{187}	9	6	+3	9
d_{25}	10	9	+1	1
Sum of Square Distances				24

The numbers of documents in R_1 and R_2 are different

- Compute the set S_{1+2} of all distinct documents included in either of the rankings whose size is the augmented rankings, i.e., $K=|S_{1+2}|$.
- Augment ranking R_1 with all documents in S_{1+2} that are not already in R_1.
  - These documents are added at the bottom of R_1 but using the ordering dictated by R_2.
- Augment ranking R_2 with all documents in S_{1+2} that are not already in R_2.
  - These documents are added at the bottom of R_2 but using the ordering dictated by R_1.

$R_1=\{d_{123},d_{84},d_{56},d_6,d_8,d_9,d_{511},d_{129},d_{187},d_{25}\}$
$R_2=\{d_{56},d_{123},d_{84},d_8,d_6,d_{38},d_{48},d_{250},d_{113},d_3\}$
$R_{+1}=\{d_{123},d_{84},d_{56},d_6,d_8,d_9,d_{511},d_{129},d_{187},d_{25}, d_{38},d_{48},d_{250},d_{113},d_3\}$
$R_{+2}=\{d_{56},d_{123},d_{84},d_8,d_6,d_{38},d_{48},d_{250},d_{113},d_3, d_9,d_{511},d_{129},d_{187},d_{25}\}$

$K=|\ S_{1+2}\ |=15$

$\sum_{j=1}^{15}(s_{1,j}-s_{2,j})^2=258$

$S(R_1\text{-}R_2)=1-\frac{6\times 258}{15(15^2-1)}=0.539$

The Kendall Tau Coefficient

The Kendall Tau Coefficient

- It is difficult to assign an operational interpretation to Spearman coefficient
- One alternative is to use a coefficient that has a natural and intuitive interpretation, as the Kendall Tau coefficient

The Kendall Tau Coefficient

- When we think of rank correlations, we think of how two rankings tend to vary in similar ways
- To illustrate, consider two documents d_j and d_k and their positions in the rankings $\mathcal{R}_1$ and $\mathcal{R}_2$
- Further, consider the differences in rank positions for these two documents in each ranking, i.e.,

$$\begin{array}{ccc} s_{1,k} & - & s_{1,j} \\ s_{2,k} & - & s_{2,j} \end{array}$$

- If these differences have the same sign, we say that the document pair $[d_k, d_j]$ is **concordant** in both rankings
- If they have different signs, we say that the document pair is **discordant** in the two rankings

The Kendall Tau Coefficient

- Consider the top 5 documents in rankings $\mathcal{R}_1$ and $\mathcal{R}_2$

documents	$s_{1,j}$	$s_{2,j}$	$s_{i,j} - s_{2,j}$
d_{123}	1	2	-1
d_{84}	2	3	-1
d_{56}	3	1	+2
d_{6}	4	5	-1
d_{8}	5	4	+1

- The ordered document pairs in ranking $\mathcal{R}_1$ are

$$[d_{123}, d_{84}], \quad [d_{123}, d_{56}], \quad [d_{123}, d_{6}], \quad [d_{123}, d_{8}],$$
$$[d_{84}, d_{56}], \quad [d_{84}, d_{6}], \quad [d_{84}, d_{8}],$$
$$[d_{56}, d_{6}], \quad [d_{56}, d_{8}],$$
$$[d_{6}, d_{8}]$$

for a total of $\frac{1}{2} \times 5 \times 4$, or 10 ordered pairs

The Kendall Tau Coefficient

- Repeating the same exercise for the top 5 documents in ranking $\mathcal{R}_2$, we obtain

$$
\begin{array}{l}
[d_{56}, d_{123}], \quad [d_{56}, d_{84}], \quad [d_{56}, d_8], \quad [d_{56}, d_6], \\
[d_{123}, d_{84}], \quad [d_{123}, d_8], \quad [d_{123}, d_6], \\
[d_{84}, d_8], \quad [d_{84}, d_6], \\
[d_8, d_6]
\end{array}
$$

- We compare these two sets of ordered pairs looking for concordant and discordant pairs

documents	$s_{1,j}$	$s_{2,j}$	$s_{i,j} - s_{2,j}$
d_{123}	1	2	-1
d_{84}	2	3	-1
d_{56}	3	1	+2
d_6	4	5	-1
d_8	5	4	+1

The Kendall Tau Coefficient

- Let us mark with a C the concordant pairs and with a D the discordant pairs
- For ranking $\mathcal{R}_1$, we have

$$
\begin{array}{ll}
[d_{123}, d_{84}],\ [d_{123}, d_{56}],\ [d_{123}, d_6],\ [d_{123}, d_8], & C, D, C, C, \\
[d_{84}, d_{56}],\ [d_{84}, d_6],\ [d_{84}, d_8], & D, C, C, \\
[d_{56}, d_6],\ [d_{56}, d_8], & C, C, \\
[d_6, d_8] & D
\end{array}
$$

- For ranking $\mathcal{R}_2$, we have

$$
\begin{array}{ll}
[d_{56}, d_{123}],\ [d_{56}, d_{84}],\ [d_{56}, d_8],\ [d_{56}, d_6], & D, D, C, C, \\
[d_{123}, d_{84}],\ [d_{123}, d_8],\ [d_{123}, d_6], & C, C, C, \\
[d_{84}, d_8],\ [d_{84}, d_6], & C, C, \\
[d_8, d_6] & D
\end{array}
$$

The Kendall Tau Coefficient

- That is, a total of 20, i.e., $K(K-1)$, ordered pairs are produced jointly by the two rankings
- Among these, 14 pairs are concordant and 6 pairs are discordant
- The Kendall Tau coefficient is defined as

$$\tau(\mathcal{R}_1, \mathcal{R}_2) = P(\mathcal{R}_1 = \mathcal{R}_2) - P(\mathcal{R}_1 \neq \mathcal{R}_2)$$

- In our example

$$\begin{aligned} \tau(\mathcal{R}_1, \mathcal{R}_2) &= \frac{14}{20} - \frac{6}{20} \\ &= 0.4 \end{aligned}$$

The Kendall Tau Coefficient

- Let,
  - $\Delta(\mathcal{R}_1, \mathcal{R}_2)$: number of discordant document pairs in $\mathcal{R}_1$ and $\mathcal{R}_2$
  - $K(K-1) - \Delta(\mathcal{R}_1, \mathcal{R}_2)$: number of concordant document pairs in $\mathcal{R}_1$ and $\mathcal{R}_2$
- Then,

$$
\begin{aligned}
P(\mathcal{R}_1 = \mathcal{R}_2) &= \frac{K(K-1) - \Delta(\mathcal{R}_1, \mathcal{R}_2)}{K(K-1)} \\
P(\mathcal{R}_1 \neq \mathcal{R}_2) &= \frac{\Delta(\mathcal{R}_1, \mathcal{R}_2)}{K(K-1)}
\end{aligned}
$$

which yields $\boxed{\tau(\mathcal{R}_1, \mathcal{R}_2) = 1 - \frac{2 \times \Delta(\mathcal{R}_1, \mathcal{R}_2)}{K(K-1)}}$

The Kendall Tau Coefficient

- For the case of our previous example, we have
  - $\Delta(\mathcal{R}_1, \mathcal{R}_2) = 6$
  - $K = 5$
  - Thus,

$$\tau(\mathcal{R}_1, \mathcal{R}_2) = 1 - \frac{2 \times 6}{5(5-1)} = 0.4$$

  as before

- The Kendall Tau coefficient is defined only for rankings over a same set of elements
- Most important, it has a simpler algebraic structure than the Spearman coefficient

Reference Collections

Reference Collections

- With small collections one can apply the Cranfield evaluation paradigm to provide relevance assessments
- With large collections, however, not all documents can be evaluated relatively to a given information need
- The alternative is consider only the top k documents produced by various ranking algorithms for a given information need
  - This is called the **pooling method**
- The method works for reference collections of a few million documents, such as the **TREC collections**

The TREC Collections

The TREC Conferences

- TREC is an yearly promoted conference dedicated to experimentation with large test collections
- For each TREC conference, a set of experiments is designed
- The research groups that participate in the conference use these experiments to compare their retrieval systems
- As with most test collections, a TREC collection is composed of three parts:
  - the documents
  - the example information requests (called **topics**)
  - a set of relevant documents for each example information request

The Document Collections

- The main TREC collection has been growing steadily over the years
- The TREC-3 collection has roughly 2 gigabytes
- The TREC-6 collection has roughly 5.8 gigabytes
  - It is distributed in 5 CD-ROM disks of roughly 1 gigabyte of compressed text each
  - Its 5 disks were also used at the TREC-7 and TREC-8 conferences
- The *Terabyte test collection* introduced at TREC-15, also referred to as GOV2, includes 25 million Web documents crawled from sites in the ".gov" domain

The Document Collections

- TREC documents come from the following sources:

WSJ	$\longrightarrow$	*Wall Street Journal*
AP	$\longrightarrow$	Associated Press (news wire)
ZIFF	$\longrightarrow$	Computer Selects (articles), Ziff-Davis
FR	$\longrightarrow$	Federal Register
DOE	$\longrightarrow$	US DOE Publications (abstracts)
SJMN	$\longrightarrow$	*San Jose Mercury News*
PAT	$\longrightarrow$	US Patents
FT	$\longrightarrow$	*Financial Times*
CR	$\longrightarrow$	Congressional Record
FBIS	$\longrightarrow$	Foreign Broadcast Information Service
LAT	$\longrightarrow$	*LA Times*

The Document Collections

- Contents of TREC-6 disks 1 and 2

Disk	Contents	Size **Mb**	Number **Docs**	Words/Doc. **(median)**	Words/Doc. **(mean)**
1	WSJ, 1987-1989	267	98,732	245	434.0
	AP, 1989	254	84,678	446	473.9
	ZIFF	242	75,180	200	473.0
	FR, 1989	260	25,960	391	1315.9
	DOE	184	226,087	111	120.4
2	WSJ, 1990-1992	242	74,520	301	508.4
	AP, 1988	237	79,919	438	468.7
	ZIFF	175	56,920	182	451.9
	FR, 1988	209	19,860	396	1378.1

The Document Collections

- Contents of TREC-6 disks 3-6

Disk	Contents	Size **Mb**	Number **Docs**	Words/Doc. **(median)**	Words/Doc. **(mean)**
3	SJMN, 1991	287	90,257	379	453.0
	AP, 1990	237	78,321	451	478.4
	ZIFF	345	161,021	122	295.4
	PAT, 1993	243	6,711	4,445	5391.0
4	FT, 1991-1994	564	210,158	316	412.7
	FR, 1994	395	55,630	588	644.7
	CR, 1993	235	27,922	288	1373.5
5	FBIS	470	130,471	322	543.6
	LAT	475	131,896	351	526.5
6	FBIS	490	120,653	348	581.3

The Document Collections

- Documents from all subcollections are tagged with SGML to allow easy parsing
- Some structures are common to all documents:
  - The document number, identified by $<$DOCNO$>$
  - The field for the document text, identified by $<$TEXT$>$
- Minor structures might be different across subcollections

The Document Collections

- An example of a TREC document in the **Wall Street Journal** subcollection

```
<doc>

<docno> WSJ880406-0090 </docno>
<hl> AT&T Unveils Services to Upgrade Phone Networks
Under Global Plan </hl>
<author> Janet Guyon (WSJ Staff) </author>
<dateline> New York </dateline>

<text>
American Telephone & Telegraph Co introduced the first
of a new generation of phone services with broad ...
</text>

</doc>
```

The TREC Web Collections

- A Web Retrieval track was introduced at TREC-9
  - The VLC2 collection is from an Internet Archive crawl of 1997
  - WT2g and WT10g are subsets of the VLC2 collection
  - .GOV is from a crawl of the .gov Internet done in 2002
  - .GOV2 is the result of a joint NIST/UWaterloo effort in 2004

Collection	# Docs	Avg Doc Size	Collection Size
VLC2 (WT100g)	18,571,671	5.7 KBytes	100 GBytes
WT2g	247,491	8.9 KBytes	2.1 GBytes
WT10g	1,692,096	6.2 KBytes	10 GBytes
.GOV	1,247,753	15.2 KBytes	18 GBytes
.GOV2	27 million	15 KBytes	400 GBytes

Information Requests Topics

- Each TREC collection includes a set of example **information requests**
- Each request is a description of an information need in natural language
- In the TREC nomenclature, each test information request is referred to as a **topic**

Information Requests Topics

- An example of an information request is the topic numbered 168 used in TREC-3:

```
<top>

<num> Number:  168
<title> Topic:  Financing AMTRAK

<desc> Description:
A document will address the role of the Federal Government in
financing the operation of the National Railroad Transportation
Corporation (AMTRAK)
<narr> Narrative:  A relevant document must provide information on
the government’s responsibility to make AMTRAK an economically viable
entity.  It could also discuss the privatization of AMTRAK as an
alternative to continuing government subsidies.  Documents comparing
government subsidies given to air and bus transportation with those
provided to AMTRAK would also be relevant

</top>
```

Information Requests Topics

- The task of converting a topic into a system query is considered to be a part of the evaluation procedure
- The number of topics prepared for the first eight TREC conferences is 450

The Relevant Documents

- The set of relevant documents for each topic is obtained from a pool of possible relevant documents
  - This pool is created by taking the top K documents (usually, $K = 100$) in the rankings generated by various retrieval systems
  - The documents in the pool are then shown to human assessors who ultimately decide on the relevance of each document
- This technique of assessing relevance is called the pooling method and is based on two assumptions:
  - First, that the vast majority of the relevant documents is collected in the assembled pool
  - Second, that the documents which are not in the pool can be considered to be not relevant

The Benchmark Tasks

- The TREC conferences include two main information retrieval tasks
  - **Ad hoc task**: a set of new requests are run against a fixed document database
  - **routing task**: a set of fixed requests are run against a database whose documents are continually changing
- For the ad hoc task, the participant systems execute the topics on a pre-specified document collection
- For the routing task, they receive the test information requests and two distinct document collections
  - The first collection is used for training and allows the tuning of the retrieval algorithm
  - The second is used for testing the tuned retrieval algorithm

The Benchmark Tasks

- Starting at the TREC-4 conference, new secondary tasks were introduced
- At TREC-6, secondary tasks were added in as follows:
  - **Chinese** — ad hoc task in which both the documents and the topics are in Chinese
  - **Filtering** — routing task in which the retrieval algorithms has only to decide whether a document is relevant or not
  - **Interactive** — task in which a human searcher interacts with the retrieval system to determine the relevant documents
  - **NLP** — task aimed at verifying whether retrieval algorithms based on natural language processing offer advantages when compared to the more traditional retrieval algorithms based on index terms

The Benchmark Tasks

- Other tasks added in TREC-6:
  - **Cross languages** — ad hoc task in which the documents are in one language but the topics are in a different language
  - **High precision** — task in which the user of a retrieval system is asked to retrieve ten documents that answer a given information request within five minutes
  - **Spoken document retrieval** — intended to stimulate research on retrieval techniques for spoken documents
  - **Very large corpus** — ad hoc task in which the retrieval systems have to deal with collections of size 20 gigabytes

The Benchmark Tasks

- The more recent TREC conferences have focused on new tracks that are not well established yet
  - The motivation is to use the experience at these tracks to develop new reference collections that can be used for further research
- At TREC-15, the main tracks were question answering, genomics, terabyte, enterprise, spam, legal, and blog

2014 TREC Tracks

- **Clinical Decision Support Track**: Investigate techniques for linking medical cases to information relevant for patient care.
- **Contextual Suggestion Track**: Investigate search techniques for complex information needs that are highly dependent on context and user interests.
- **Federated Web Search Track**: Investigate techniques for the selection and combination of search results from a large number of real on-line web search services.
- **Knowledge Base Acceleration Track**: Develop techniques to dramatically improve the efficiency of (human) knowledge base curators by having the system suggest modifications/extensions to the KB based on its monitoring of the data streams.
- **Microblog Track**: Examine the nature of real-time information needs and their satisfaction in the context of microblogging environments such as Twitter.
- **Session Track**: Provide the necessary resources in the form of test collections to simulate user interaction and help evaluate the utility of an IR system over a sequence of queries and user interactions, rather than for a single "one-shot" query.
- **Temporal Summarization Track**: Develop systems that allow users to efficiently monitor the information associated with an event over time.
- **Web Track**: Explore Web-specific retrieval tasks, including diversity and efficiency tasks, over collections of up to one billion Web pages.

Evaluation Measures at TREC

- At the TREC conferences, four basic types of evaluation measures are used:
  - **Summary table statistics** — this is a table that summarizes statistics relative to a given task
  - **Recall-precision averages** — these are a table or graph with average precision (over all topics) at 11 standard recall levels
  - **Document level averages** — these are average precision figures computed at specified document cutoff values
  - **Average precision histogram** — this is a graph that includes a single measure for each separate topic

Other Reference Collections

- Inex
- Reuters
- OHSUMED
- NewsGroups
- NTCIR
- CLEF
- Small collections
  - ADI, CACM, ISI, CRAN, LISA, MED, NLM, NPL, TIME
  - CF (Cystic Fibrosis)

INEX Collection

- INitiative for the Evaluation of XML Retrieval
- It is a test collection designed specifically for evaluating XML retrieval effectiveness
- It is of central importance for the XML community

Reuters, OHSUMED, NewsGroups

- Reuters
  - A reference collection composed of news articles published by Reuters
  - It contains more than 800 thousand documents organized in 103 topical categories.
- OHSUMED
  - A reference collection composed of medical references from the Medline database
  - It is composed of roughly 348 thousand medical references, selected from 270 journals published in the years 1987-1991

Reuters, OHSUMED, NewsGroups

- NewsGroups
  - Composed of thousands of newsgroup messages organized according to 20 groups
- These three collections contain information on categories (classes) associated with each document
- Thus, they are particularly suitable for the evaluation of text classification algorithms

NTCIR Collections

- NII Test Collection for IR Systems
- It promotes yearly workshops code-named NTCIR Workshops
  - For these workshops, various reference collections composed of patents in Japanese and English have been assembled
- To illustrate, the NTCIR-7 PATMT (Patent Translation Test) collection includes:
  - 1.8 million translated sentence pairs (Japanese-English)
  - 5,200 test sentence pairs
  - 124 queries
  - human judgements for the translation results

NTCIR-12

- **Search Intent and Task Mining**: explore and evaluate the technologies of understanding user intents behind the query and satisfying different user intents
- **Medical Natural Language Processing for Clinical Document**: Assign a suitable diagnosis and the corresponding disease code to a clinical case in Japanese
- **Mobile Information Access**: generate a two-layered summary in response to a given query that fits screens of mobile devices instead of ten blue links
- **Spoken Query and Spoken Document Retrieval**
- **Temporal Information Access**
- **Mathematical Information Retrieval**: Retrieval using queries comprised of keywords and formulae
- **Lifelog**
- **QA Lab for Entrance Exam**
- **Short Text Conversation**

CLEF Collections

- CLEF is an annual conference focused on Cross-Language IR (CLIR) research and related issues
- For supporting experimentation, distinct CLEF reference collections have been assembled over the years

Other Small Test Collections

- Many small test collections have been used by the IR community over the years
- They are no longer considered as state of the art test collections, due to their small sizes

Collection	Subject	Num Docs	Num Queries
ADI	Information Science	82	35
CACM	Computer Science	3200	64
ISI	Library Science	1460	76
CRAN	Aeronautics	1400	225
LISA	Library Science	6004	35
MED	Medicine	1033	30
NLM	Medicine	3078	155
NPL	Elec Engineering	11,429	100
TIME	General Articles	423	83

Other Small Test Collections

- Another small test collection of interest is the Cystic Fibrosis (CF) collection
- It is composed of:
  - 1,239 documents indexed with the term ‘cystic fibrosis’ in the MEDLINE database
  - 100 information requests, which have been generated by an expert with research experience with cystic fibrosis
- Distinctively, the collection includes 4 separate relevance scores for each relevant document

Evaluation using Clickthrough Data

Evaluation w/ Clickthrough Data

- Reference collections provide an effective means of evaluating the relevance of the results set
- However, they can only be applied to a relatively small number of queries
- On the other side, the query log of a Web search engine is typically composed of billions of queries
  - Thus, evaluation of a Web search engine using reference collections has its limitations

Evaluation w/ Clickthrough Data

- One very promising alternative is evaluation based on the analysis of clickthrough data
- It can be obtained by observing how frequently the users click on a given document, when it is shown in the answer set for a given query
- This is particularly attractive because the data can be collected at a low cost without overhead for the user

Biased Clickthrough Data

- To compare two search engines A and B , we can measure the clickthrough rates in rankings $\mathcal{R}_A$ and $\mathcal{R}_B$
- To illustrate, consider that a same query is specified by various users in distinct moments in time
- We select one of the two search engines randomly and show the results for this query to the user
- By comparing clickthrough data over millions of queries, we can infer which search engine is preferable

Biased Clickthrough Data

- However, clickthrough data is difficult to interpret
- To illustrate, consider a query q and assume that the users have clicked
  - on the answers 2, 3, and 4 in the ranking $\mathcal{R}_A$, and
  - on the answers 1 and 5 in the ranking $\mathcal{R}_B$
- In the first case, the average clickthrough rank position is (2+3+4)/3, which is equal to 3
- In the second case, it is (1+5)/2, which is also equal to 3
- The example shows that clickthrough data is difficult to analyze

Biased Clickthrough Data

- Further, clickthrough data is **not** an absolute indicator of relevance
- That is, a document that is highly clicked is not necessarily relevant
- Instead, it is preferable with regard to the other documents in the answer
- Further, since the results produced by one search engine are not relative to the other, it is difficult to use them to compare two distinct ranking algorithms directly
- The alternative is to mix the two rankings to collect unbiased clickthrough data, as follows

Unbiased Clickthrough Data

- To collect unbiased clickthrough data from the users, we mix the result sets of the two ranking algorithms
- This way we can compare clickthrough data for the two rankings
- To mix the results of the two rankings, we look at the top results from each ranking and mix them

Unbiased Clickthrough Data

- The algorithm below achieves the effect of mixing rankings $\mathcal{R}_A$ and $\mathcal{R}_B$

```
Input:   R_A = (a_1, a_2, ...),  R_B = (b_1, b_2, ...).
Output:  a combined ranking R.
combine_ranking(R_A, R_B, k_a, k_b, R) {
    if (k_a = k_b) {
        if (R_A[k_a + 1] ∉ R) { R := R + R_A[k_a + 1] }
        combine_ranking(R_A, R_B, k_a + 1, k_b, R)
    } else {
        if (R_B[k_b + 1] ∉ R) { R := R + R_B[k_b + 1] }
        combine_ranking(R_A, R_B, k_a, k_b + 1, R)
    }
}
```

Unbiased Clickthrough Data

- Notice that, among any set of top r ranked answers, the number of answers originary from each ranking differs by no more than 1
- By collecting clickthrough data for the combined ranking, we further ensure that the data is unbiased and reflects the user preferences

Unbiased Clickthrough Data

- Under mild conditions, it can be shown that

  Ranking $\mathcal{R}_A$ contains more relevant documents than ranking $\mathcal{R}_B$ only if the clickthrough rate for $\mathcal{R}_A$ is higher than the clickthrough rate for $\mathcal{R}_B$. Most important, under mild assumptions, the comparison of two ranking algorithms with basis on the combined ranking clickthrough data is consistent with a comparison of them based on relevance judgements collected from human assessors.

- This is a striking result that shows the correlation between clicks and the relevance of results