Chapter 4

Retrieval Evaluation

- The Cranfield Paradigm
- Retrieval Performance Evaluation
- Evaluation Using Reference Collections
- Interactive Systems Evaluation
- Search Log Analysis using Clickthrough Data
- Trends and Research Issues

Introdution

- 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
- Some metrics have been defined that, on average, have a correlation with the preferences of a group of users
- Without proper evaluation,
  - we have no way to establish how well an IR system is performing
  - we cannot compare its retrieval performance with that of other systems objectively

Introdution

- Systematic evaluation of the IR system allows answering:
  - 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

Introdution

- Retrieval performance evaluation
  - To associate a quantitative metric to the results produced by an IR system
  - This metric should be directly associated with the relevance of the results
  - It compares the results produced by the system with results suggested by humans for a same set of queries

The Cranfield Paradigm

The Cranfield Paradigm

- Evaluation of IR systems is the result of early experimentation initiated in the 50's by Cyril Cleverdon
- The insights derived from these experiments provide a foundation for the evaluation of IR systems
- Back in 1952, Cleverdon took notice of a new indexing system called **Uniterm**, proposed by Mortimer Taube
- Cleverdon thought it appealing and with Bob Thorne, a colleague, did a small test
- Cleverdon manually indexed 200 documents using Uniterm and asked Thorne to run some queries
- This put Cleverdon on a life trajectory of reliance on experimentation for evaluating indexing systems

The Cranfield Paradigm

- Cleverdon obtained a grant from the National Science Foundation to compare distinct indexing systems
- These experiments provided interesting insights, that culminated in the modern metrics of precision and recall
  - **Recall ratio:** the fraction of relevant documents retrieved
  - **Precision ration:** the fraction of documents retrieved that are relevant
- For instance, it became clear that, in practical situations, the majority of searches does not require high recall
- Instead, the vast majority of the users require just a few relevant answers

The Cranfield Paradigm

- The next step was to devise a set of experiments that would allow evaluating each indexing system in isolation more thoroughly
- The result was a **test reference collection** composed of documents, queries, and relevance judgements
  - It became known as the *Cranfield-2* collection
- The same set of documents and queries can be used to evaluate different ranking systems
- The uniformity of this setup allows quick evaluation once relevance judgements have been produced

Reference Collections

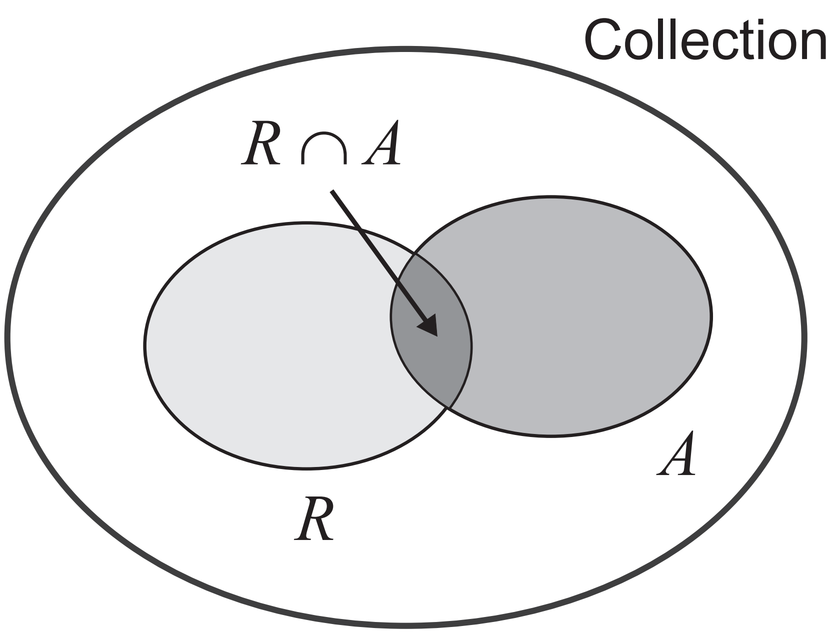
- Reference collections are based on the foundations established by the Cranfield experiments
- They constitute the most used evaluation method in IR
- A reference collection is composed of:
  - A set $\mathcal{D}$ of pre-selected documents
  - A set I of information need descriptions used for testing
  - 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

Retrieval Performance Evaluation

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



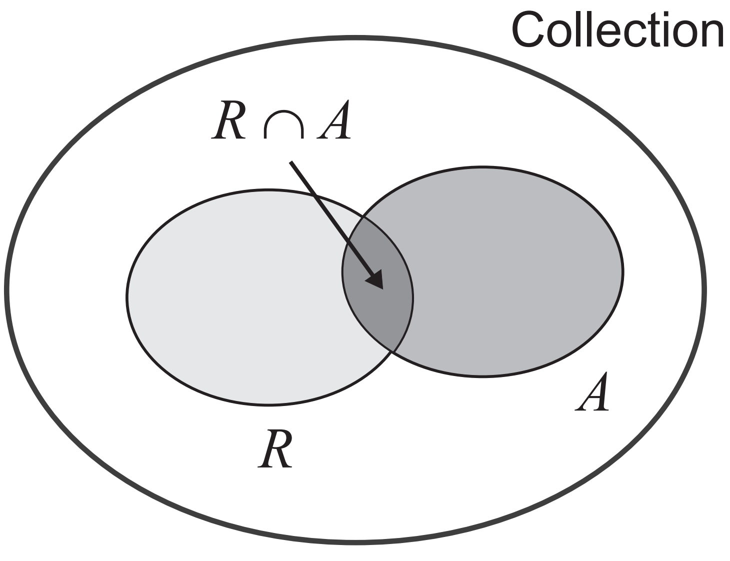
Precision and Recall

- Then, 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 viewpoint using the sets R, A, and $R \cap A$ assume 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 his 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):

01. d_{123} •	06. d_9 •	11. d_{38}
02. d_{84}	07. d_{511}	12. d_{48}
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 •

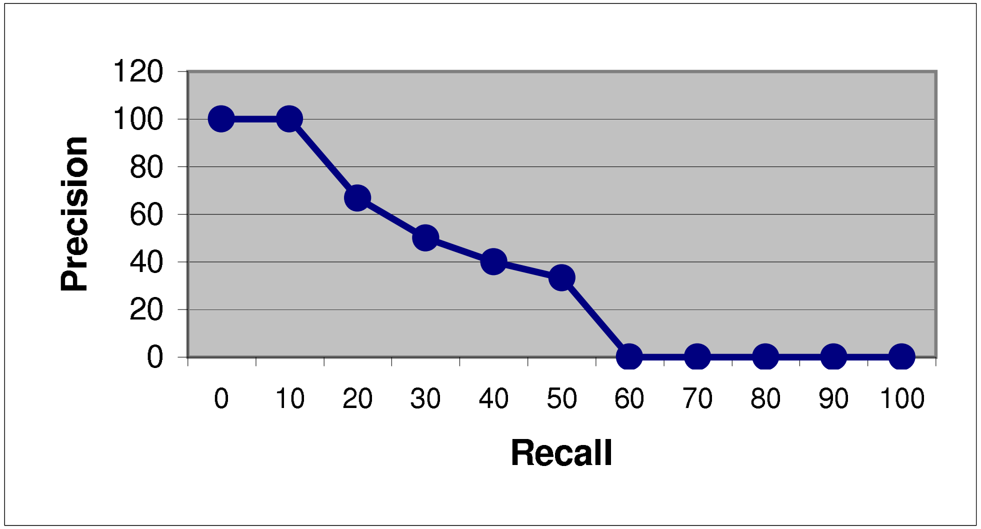
Precision and Recall

- If we examine this ranking, we observe
  - 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



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

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} •

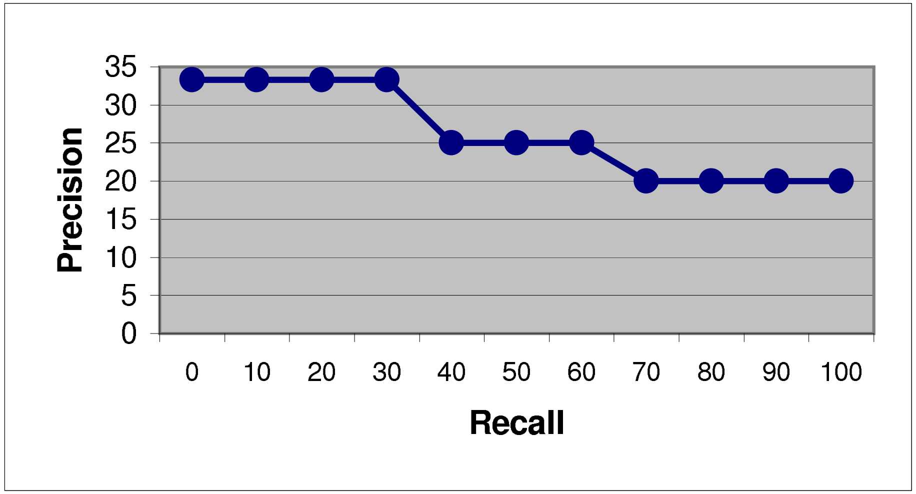
Precision and Recall

- 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, $P(r_j) = max_{\ r_j \leq r \leq r_{j+1}} \ P(r)$
- 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

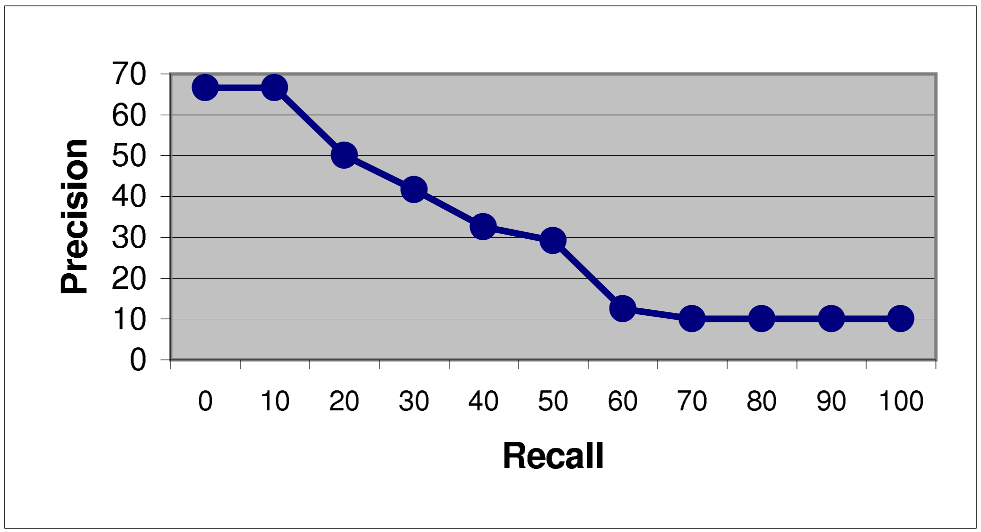
- In the examples above, the precision and recall figures have been computed for single queries
- Usually, however, retrieval algorithms are evaluated by running them for several distinct test queries
- To evaluate the retrieval performance for N_q queries, we average the precision at each recall level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

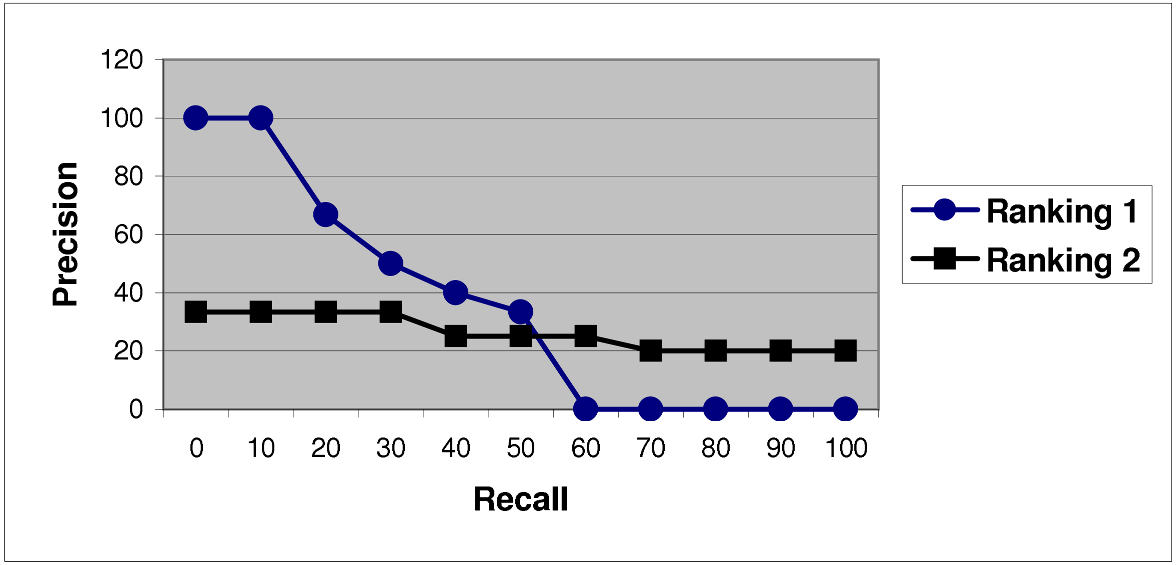
- To illustrate, figure below illustrates precision-recall figures averaged over queries q_1 and q_2



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
- Figure below illustrates average precision-recall curves for two distinct retrieval algorithms



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*

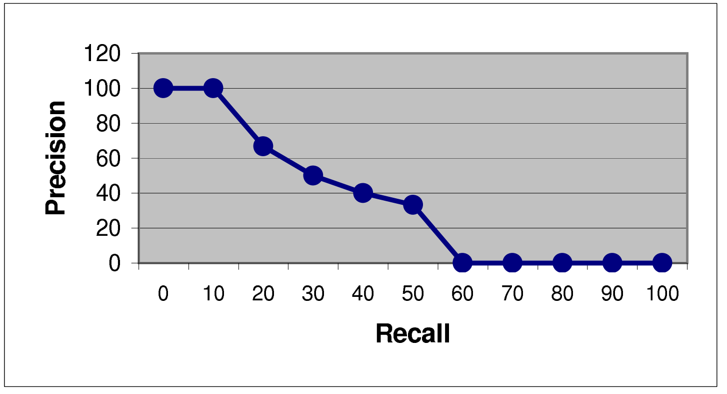
- To exemplify, consider again the ranking for the example query q_1 we have been using:

01. d_{123} •	06. d_9 •	11. d_{38}
02. d_{84}	07. d_{511}	12. d_{48}
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 •

- 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
- This metrics get an early assessment of which algorithm might be preferable in the eyes of the users

Average Precision

- The idea here is to average the precision figures obtained after each new relevant document is observed
- To illustrate, consider again the precision-recall curve below



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 precision figures after each new relevant document is observed are 1, 0.66, 0.5, 0.4, and 0.33
- Thus, the *average precision at seen relevant documents* is given by (1+0.66+0.5+0.4+0.33)/5 or 0.57

R-Precision

- Let R 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

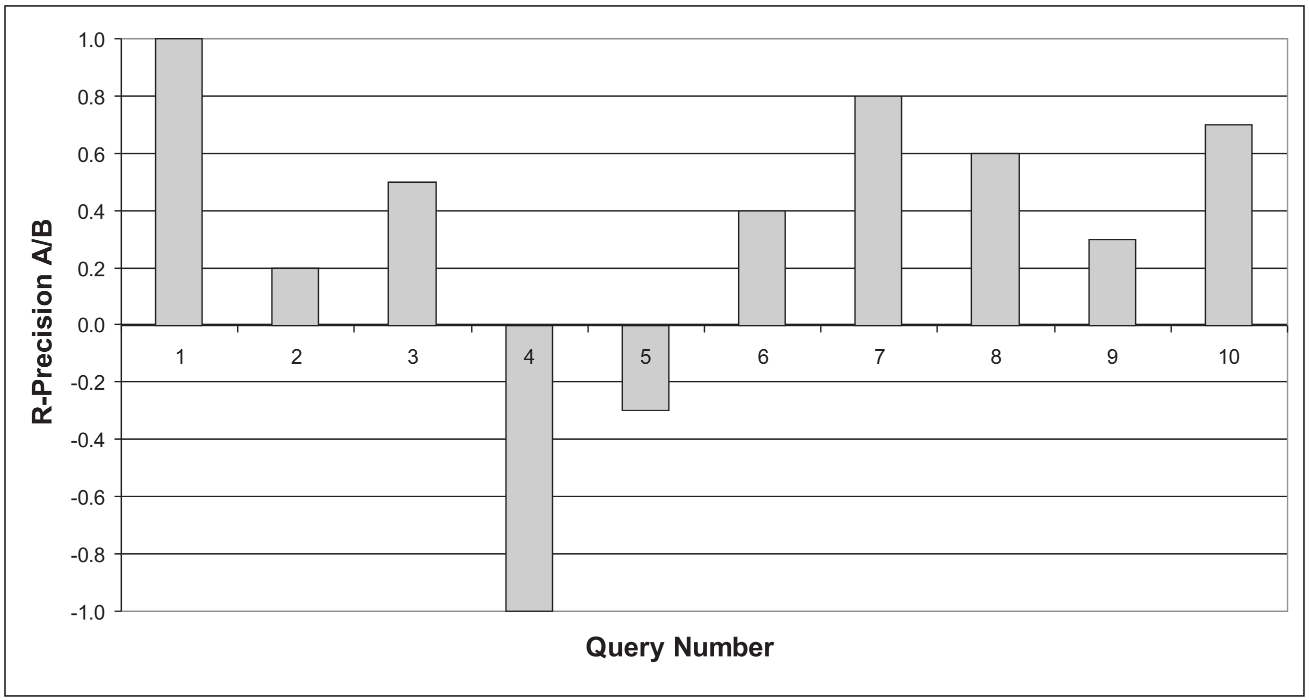
Precision Histograms

- 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 is superior for eight queries while the algorithm B performs better for the two other queries

Summary Table Statistics

- Single value measures can also be stored in a table to provide a statistical summary
- For instance, these summary table statistics could include
  - the number of queries used in the task
  - the total number of documents retrieved by all queries
  - the total number of relevant docs retrieved by all queries
  - the total number of relevant docs for all queries as judged by the specialists

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

Retrieval Performance Evaluation

Harmonic Mean and the E Measure

The Harmonic Mean

- As discussed above, a single measure which combines recall and precision might be of interest
- One such measure is the harmonic mean F of recall and precision, which is computed as

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

- 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

- 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 maximizes F is an attempt to find the best possible compromise between recall and precision

The E Measure

- Another 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

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

- 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
  - $E(j)$ is the E metric at the j-th position in the ranking

The E Measure

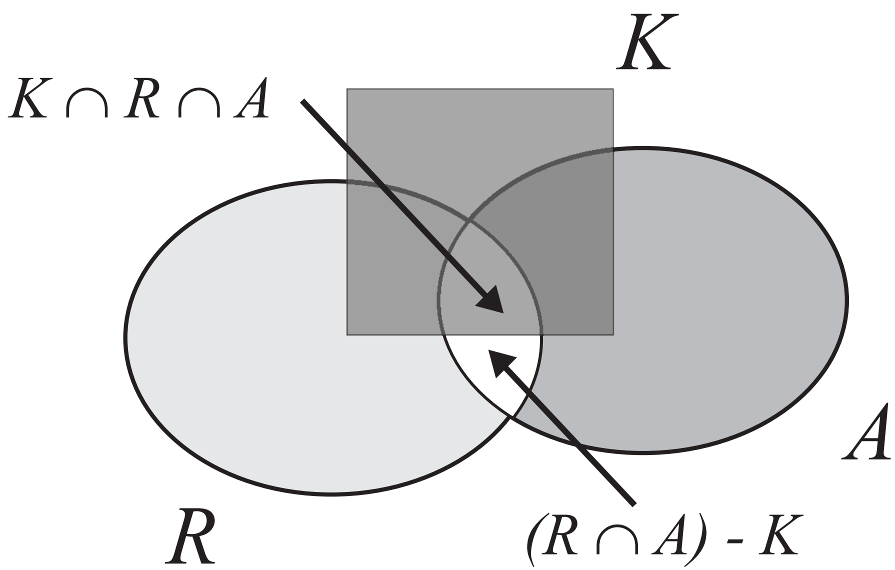
- The parameter b is specified by the user and reflects the relative importance of recall and precision
  - For $b = 1$, $E(j)$ works as the complement of the harmonic mean $F(j)$
  - Values of b greater than 1 indicate that the user is more interested in precision
  - Values of b smaller than 1 indicate that the user is more interested in recall

User-Oriented Measures

- Recall and precision assumes 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

- Also, let K be the set of documents of the collection known to the user
- The set $K \cap R \cap A$ is composed of the relevant docs known to the user that have been retrieved
- The set $(R \cap A) - K$ is composed of relevant docs that have been retrieved by are not known to the user



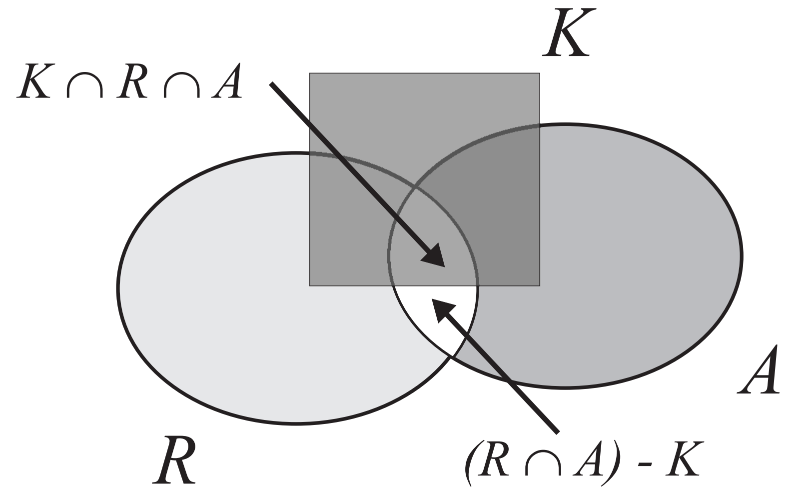
User-Oriented Measures

- 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

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



User-Oriented Measures

- A high coverage indicates that the system is finding most of the relevant docs the user expected to see
- A high novelty indicates that the system is revealing many new relevant docs which were unknown
- Additionally, two other measures can be defined
  - The **relative recall** is the ratio between the number of relevant docs found and the number of relevant docs the user expected to find
  - The **recall effort** is the ratio between the number of relevant docs the user expected to find and the number of documents examined in an attempt to find the expected relevant documents

Retrieval Performance Evaluation

DCG – Retrieval Evaluation with Graded Relevance Information

Discounted Cumulated Gain

- Precision and recall allow only binary relevance assessments
- As a result, there is no distinction 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 metric that combine 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:

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

- 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 of the IR algorithm 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

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 logarithms 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

$$
\begin{aligned}
DCG_1 &= (1, 1, 1.6, 1.6, 1.6, 2.7, 2.7, 2.7, 2.7, 3.3, 3.3, 3.3, 3.3, 3.3, 4.1) \\
DCG_2 &= (0, 0, 1.2, 1.2, 1.2, 1.2, 1.2, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 2.3)
\end{aligned}
$$

- 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

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

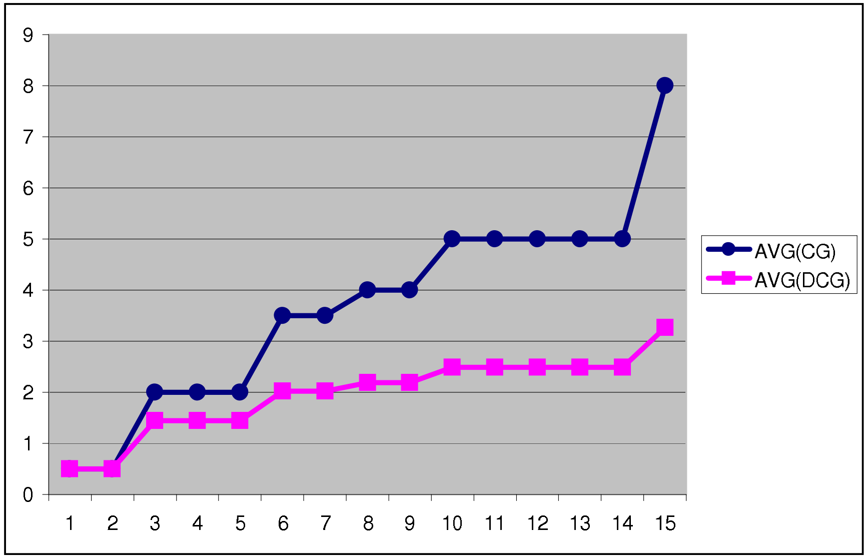
$$\overline{CG}[i] = \sum_{j=1}^{N_q} \frac{CG_j[i]}{N_q}; \qquad \overline{DCG}[i] = \sum_{j=1}^{N_q} \frac{DCG_j[i]}{N_q}$$

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

$$\begin{aligned} \overline{CG} &= (0.5, 0.5, 2.0, 2.0, 2.0, 3.5, 3.5, 4.0, 4.0, 5.0, 5.0, 5.0, 5.0, 5.0, 8.0) \\ \overline{DCG} &= (0.5, 0.5, 1.4, 1.4, 1.4, 2.0, 2.0, 2.1, 2.1, 2.4, 2.4, 2.4, 2.4, 2.4, 3.2) \end{aligned}$$

DCG Curves

- Then, average curves can be drawn by varying the rank positions from 1 to a pre-established threshold
- In the example above, this threshold is set at 15, in the Web it is normally set at 10
- Figure below shows CG and DCG curves corresponding to the $\overline{CG}$ and $\overline{DCG}$ vectors



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

$$IG_1 = (3,3,2,1,1,0,0,0,0,0,0,0,0,0,0)$$
$$IG_2 = (3,2,1,0,0,0,0,0,0,0,0,0,0,0,0)$$

Ideal CG and DCG Metrics

- Ideal CG and ideal DCG vectors can be computed analogously to the computations of CG and DCG
- For instance, for the example queries q_1 and q_2, the ideal CG vectors are given by

$$\begin{aligned} ICG_1 &= (3, 6, 8, 9, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10) \\ ICG_2 &= (3, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6) \end{aligned}$$

- The ideal DCG vectors are given by

$$\begin{aligned} IDCG_1 &= (3, 6, 7.2, 7.7, 8.1, 8.1, 8.1, 8.1, 8.1, 8.1, 8.1, 8.1, 8.1, 8.1, 8.1) \\ IDCG_2 &= (3, 5, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6, 5.6) \end{aligned}$$

Ideal CG and DCG Metrics

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

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

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

$$\begin{aligned} \overline{ICG} &= (3.0, 5.5, 7.0, 7.5, 8.0, 8.0, 8.0, 8.0, 8.0, 8.0, 8.0, 8.0, 8.0, 8.0, 8.0) \\ \overline{IDCG} &= (3.0, 5.5, 6.4, 6.7, 6.9, 6.9, 6.9, 6.9, 6.9, 6.9, 6.9, 6.9, 6.9, 6.9, 6.9) \end{aligned}$$

- 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

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
- The consequence is that it is 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

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

Normalized DCG

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

$$
\begin{aligned}
NCG &= (0.17, 0.09, 0.29, 0.27, 0.25, 0.44, 0.44, \\
&\quad 0.50, 0.50, 0.63, 0.63, 0.63, 0.63, 0.63, 1.00) \\
NDCG &= (0.17, 0.09, 0.22, 0.22, 0.21, 0.29, 0.29, \\
&\quad 0.32, 0.32, 0.36, 0.36, 0.36, 0.36, 0.36, 0.47)
\end{aligned}
$$

- 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 distinguish 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 ranking position which makes the metrics more imune to outliers
  - Discounted cumulated gain allows down weighting the impact of relevant documents found late in the ranking

Retrieval Performance Evaluation

BPREF – Retrieval Evaluation with Incomplete Information

BPREF

- The Cranfield evaluation paradigm assumes that all documents in the collection are evaluated to each query
- This works well with small collections but is not practical with larger collections
- The solution for large collections is the pooling method
  - This method compiles in a pool the top results produced by various retrieval algorithms
  - Then, only the documents in the pool are evaluated
- This method is reliable and can be used to effectively compare the retrieval performance of distinct systems

BPREF

- A different situation is observed, for instance, in the Web, that is composed of billions of documents
- There is no guarantee that the pooling method allows reliably comparing distinct Web retrieval systems
- The key underlying problem is that too many unseen docs would be regarded as non-relevant
- In these cases is desirable a distinct metric designed for the evaluation of results with incomplete information
- This is the motivation for the proposal of the BPREF metric, as we now discuss

BPREF

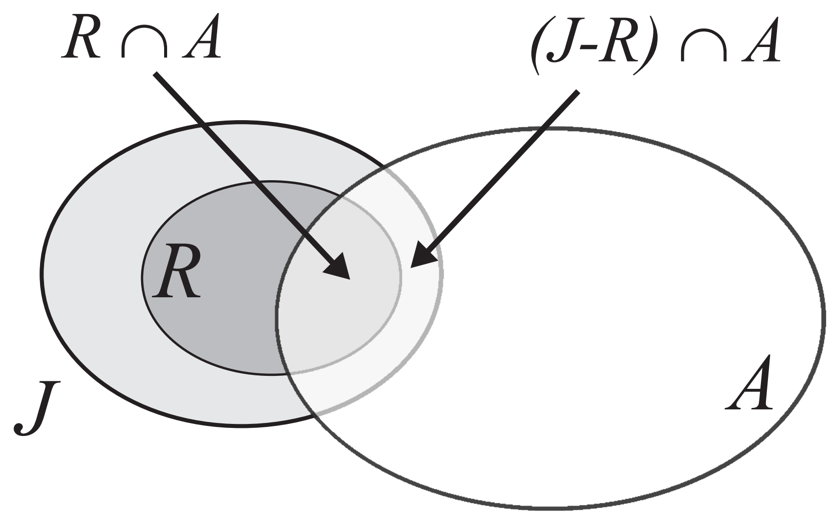
- Metrics such as precision-recall and *P@10* consider all documents that were not retrieved as non-relevant
- For very large collections this is a problem because too many documents are not retrieved for any single query
- One approach to circumvent this problem is to use preference relations
  - i.e., relations of preference between any two documents retrieved, instead of using the rank positions directly
- This is the basic idea used to derive the BPREF metric

BPREF

- Bpref measures the number of docs that are known to be non-relevant are retrieved before relevant docs
  - The measure is called Bpref because the preference relations are binary
- The assessment is simply whether document d_j is preferable to document d_k to a given information need
- To illustrate, any relevant document is preferred over any non-relevant document for a given information need

BPREF

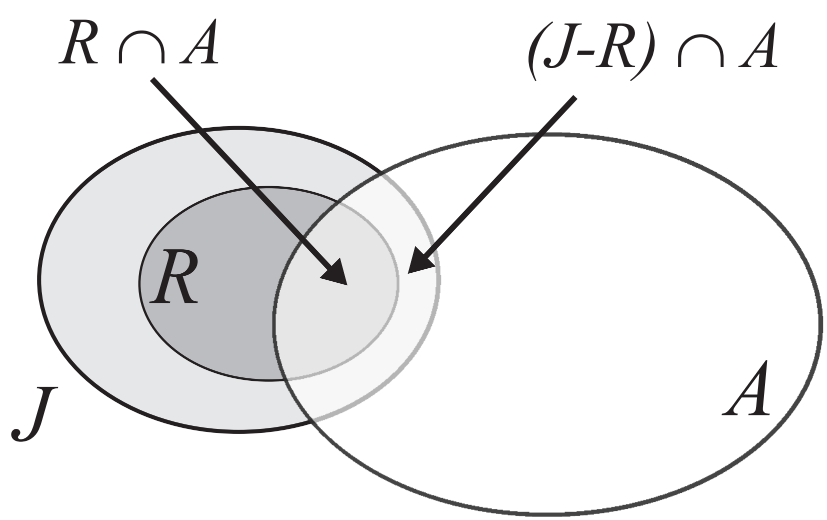
- Figure below depicts the sets considered in a Bpref computation



- J it the set of all documents *judged* by the specialists with regard to a given information need
- The set J contains the sets
  - R, composed by docs that were found to be relevant
  - $J - R$, composed by docs that were found to be non-relevant

BPREF

- Given an information need I, let
  - $\mathcal{R}_A$ be a ranking computed by an IR system A relatively to I
  - $s_{A,j}$ refer to the position of document d_j in $\mathcal{R}_A$
  - $(J - R)_{|R|}$ be the set composed of the first $|R|$ documents from $J - R$ in the ranking $\mathcal{R}_A$



BPREF

- Let N be a function given by

$$N(\mathcal{R}_A, J, R, d_j) \quad = \sum_{d_k \in (J-R)_{|R|} \;\wedge\; s_{A,k} < s_{A,j}} 1$$

- That is, N counts the number of docs from $(J - R)_{|R|}$ that appear before d_j in the ranking
- We define

$$Bpref(\mathcal{R}_A) \; = \; \frac{1}{|R|} \sum_{d_j \in R} \left[1 - \frac{N(\mathcal{R}_A, J, R, d_j)}{|R|} \right]$$

BPREF

- For each relevant document d_j in the ranking, Bpref accumulates a weight
  - This weight varies inversely with the number of judged non-relevant docs that precede each relevant doc d_j
- For instance, if all $|R|$ documents from $(J - R)_{|R|}$ precede d_j in the ranking, the weight accumulated is 0
- If just one document from $(J - R)_{|R|}$ precedes d_j in the ranking, the weight accumulated is $1 - 1/|R|$
- If no documents from $(J - R)_{|R|}$ precede d_j in the ranking, the weight accumulated is 1
- After all weights have been accumulated, the sum is normalized by $|R|$

BPREF

- Bpref is a stable metric and can be used to compare distinct algorithms in the context of large collections, because
  - The weights associated with relevant docs are normalized
  - The number of judged non-relevant docs considered is equal to the maximum number of relevant docs

BPREF-10

- Bpref is intended to be used in the presence of incomplete information
- Because that, it might just be that the number of known relevant documents is small, even as small as 1 or 2
- In this case, the metric might become unstable
  - Particularly if the number of preference relations available to define $N(\mathcal{R}_A, J, R, d_j)$ is too small
- Bpref-10 is a variation of Bpref that aims at correcting this problem

BPREF-10

- This metric ensures that a minimum of 10 preference relations are available, as follows
- Let $(J-R)_{|R|+10}$ be the set composed of the first $|R|+10$ documents from $J-R$ in the ranking
- Further, let $N10$ be a function given by

$$N10(\mathcal{R}_A, J, R, d_j) = \sum_{d_k \in (J-R)_{|R|+10} \ \wedge \ s_{A,k} < s_{A,j}} 1$$

- Then,

$$Bpref\text{-}10(\mathcal{R}_A) = \frac{1}{|R|} \sum_{d_j \in R} \left[1 - \frac{N10(\mathcal{R}_A, J, R, d_j)}{|R|+10} \right]$$