Evaluation in Document Retrieval
文档检索系统评价

References:

James Allan, University of Massachusetts Amherst
Pandu Nayak and Prabhakar Raghavan, Stanford University

Evaluation in document retrieval: outline

- Relevance (相关性) and test collections
- Effectiveness measures (有效性度量)
  - Recall and precision (召回率与精度)
  - E and F
  - Expected search length (期望搜索长度)
- TREC Conference
- Other issues and problems

Relevance

- How do you measure relevance?
- Relevance measurement requires 3 elements:
  - A benchmark document collection
  - A benchmark suite of queries
  - A usually binary assessment of either Relevant or Nonrelevant for each query and each document
    - Some work on more-than-binary, but not the standard
- What's the main challenges of relevance measurement?

3

Relevance

- Relevance is difficult to define satisfactorily
- Note: the **information need** is translated into a **query**
  - Relevance is assessed relative to the **information need** *not* the **query**
  - Input “深圳社会保险”, get 2 result sets
  - Which one is more relevant to the user’s information need?
- A relevant document is one judged useful in the context of a query
  - Who judges? What is useful?
  - Humans not very consistent
  - Judgments depend on more than document and query

Test Collections

- With real collections, never know full set of relevant documents
- A test collection usually consists of
  - set of documents
  - set of queries
  - set of relevance judgments (which docs relevant to each query)
- To compare the performance of two techniques:
  - each technique used to evaluate test queries
  - results (set or ranked list) compared using some performance measure
  - most common measures - *precision* and *recall*
- Usually use multiple measures to get different views of performance
- Usually test with multiple collections - performance is collection dependent

Chinese Web Corpus

- Data from Sogou
  - SogouT (collected in 2008)
    - http://www.sogou.com/labs/dl/t.html
    - 0.13 billion Webpages (5TB).
  - SogouQ
    - About 1 month of user query logs with user clicked URLs

The Way of Finding Relevant Documents

- Question: did system find *all* relevant material?
- To answer accurately, collection needs complete judgments
  - i.e., "yes," "no," or some score for *every* query-document pair
- For small test collections, can review all documents for all queries
- Not practical for large or medium-sized collections
  - TREC collections have millions of documents
- Other approaches that can be used
  - Pooling
  - Sampling
  - Search-based

Finding relevant documents (2)

- Search-based
  - Rather than read every document, use manually-guided search
  - Read retrieved documents until convinced all relevance found
- Sampling
  - Possible to estimate size of true relevant set by sampling
- Pooling
  - Retrieve documents using several (usually automatic) techniques
  - Judge top *n* documents for each technique
  - Relevant set is union
  - Subset of true relevant set
- All are incomplete, so when testing:
  - How should unjudged documents be treated?
  - How might this affect results?

Evaluation in document retrieval: outline

- *Relevance and test collections*
- Effectiveness measures(有效性度量)
  - Recall and precision (召回率和精度)
  - E and F
  - Expected search length （期望搜索长度）
- Significance tests
- Other issues and problems

Precision and Recall

- Precision(精度)
  - Proportion of a retrieved set that is relevant
  - Precision = |relevant ∩ retrieved| ÷ |retrieved|
    = P(relevant | retrieved)
- Recall(召回率)
  - proportion of all relevant documents in the collection included in the retrieved set
  - Recall = |relevant ∩ retrieved| ÷ |relevant|
    = P(retrieved | relevant)

Another common representation

	Relevant	Not relevant
Retrieved	A	B
Not retrieved	C	D

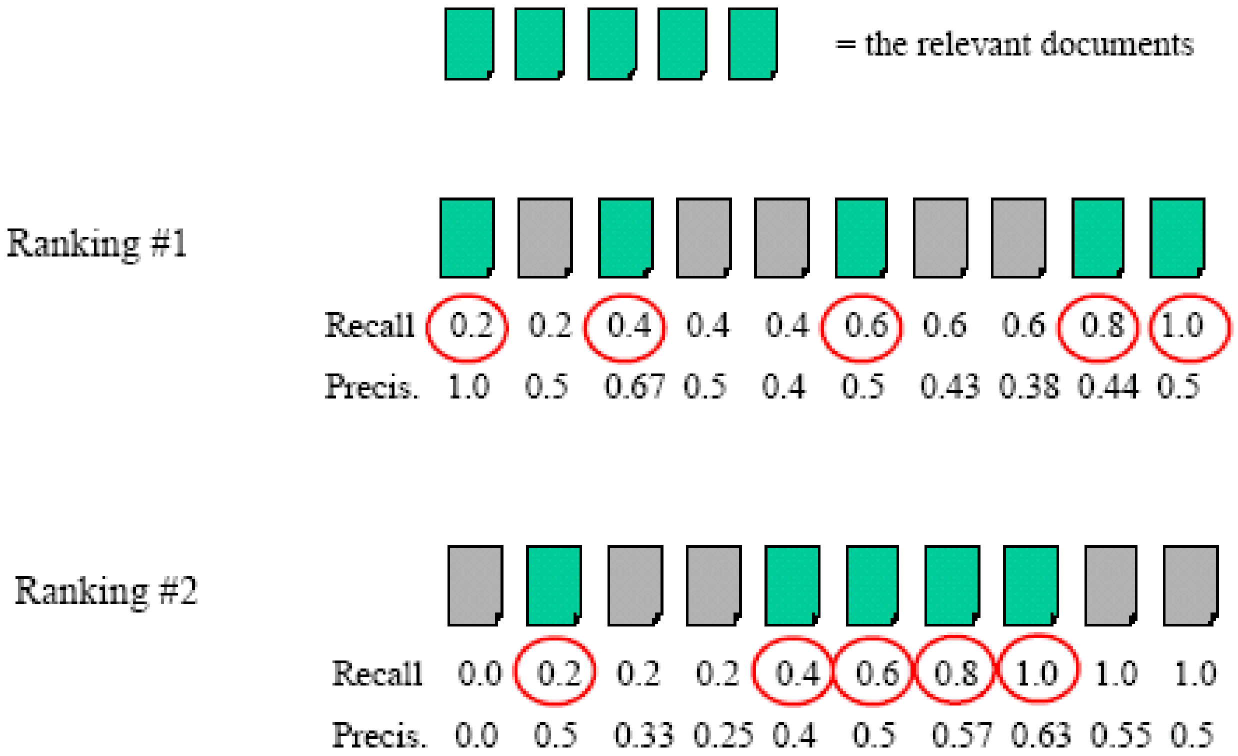
- Relevant = A+C
- Retrieved = A+B
- Collection size = A+B+C+D
- Precision = A ÷ (A+B)
- Recall = A ÷ (A+C)
- Miss = C ÷ (A+C) （漏识)
- False alarm (fallout) = B ÷ (B+D)（误报）

Precision and Recall

- Precision and recall are well-defined for sets (for unranked collection)
- For ranked retrieval, how to compute P/R values?
  - Compute a P/R point for each relevant document
  - Compute value at fixed recall points (e.g., precision at 20% recall)
  - Compute value at fixed rank cutoffs (e.g., precision at rank 20)

Precision and Recall for Ranked List

- Computing the precision and recall based on ranking



Average precision of a query

- Often want a single-number effectiveness measure
  - E.g., for a machine-learning algorithm to detect improvement
- Average precision is widely used in IR
- Calculate by averaging precision when recall increases

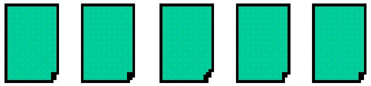
Recall	0.2	0.2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	1.0
Precis.	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5

AvgPrec= 62.2%

Recall	0.0	0.2	0.2	0.2	0.4	0.6	0.8	1.0	1.0	1.0
Precis.	0.0	0.5	0.33	0.25	0.4	0.5	0.57	0.63	0.55	0.5

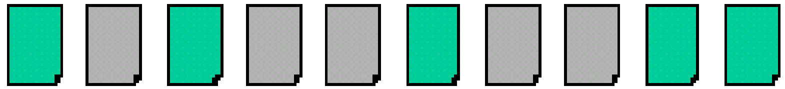
AvgPrec= 52.0%

Precision and Recall example 2



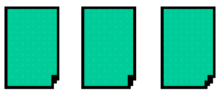
= the relevant documents

Ranking #1



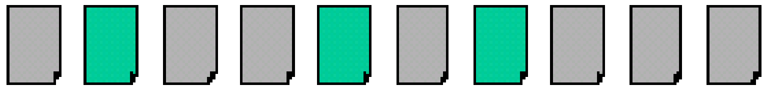
Recall	0.2	0.2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	1.0
Precis.	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5

AvgPrec= 62.2%



= different query's relevant documents

Ranking #3



Recall	0.0	0.33	0.33	0.33	0.67	0.67	1.0	1.0	1.0	1.0
Precis.	0.0	0.5	0.33	0.25	0.4	0.33	0.43	0.38	0.33	0.3

AvgPrec= 44.3%

Back to Averaging graphs

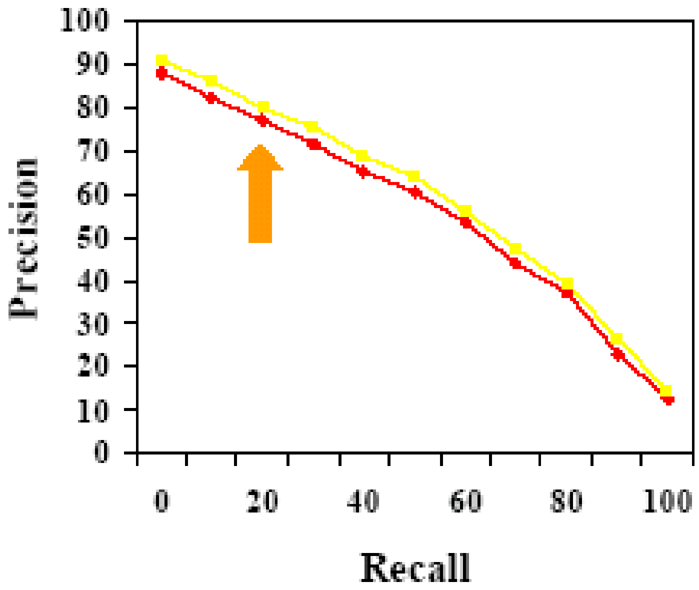
Averaging *across* queries

- It's very hard to compare P/R graphs or tables for individual queries (too much data)
  - Need to average over many queries
- Two main types of averaging
  - Micro-average - each relevant document is a point in the average
  - Macro-average - each *query* is a point in the average (Most Common)
  - What does each tell someone evaluating a system?
    - Why use one over the other?
- MAP
  - Average of many queries' average precision values
  - Called *mean* average precision (MAP)
    - "Average average precision" sounds weird

Recall/precision graphs

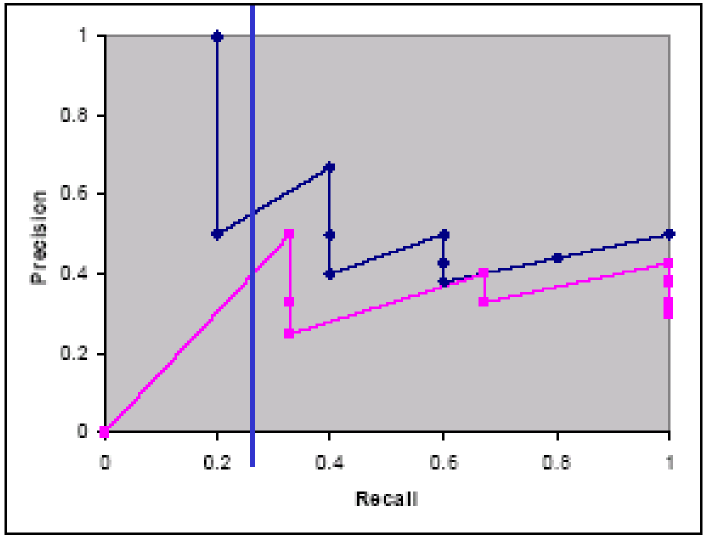
- Average precision hides information
- Sometimes better to show tradeoff in table or graph

	Precision – 44 queries	
Recall	Terms	Phrases
0	88.2	90.8 (+2.9)
10	82.4	86.1 (+4.5)
20	77.0	79.8 (+3.6)
30	71.1	75.6 (+5.4)
40	65.1	68.7 (+5.4)
50	60.3	64.1 (+6.2)
60	53.3	55.6 (+4.4)
70	44.0	47.3 (+7.5)
80	37.2	39.0 (+4.6)
90	23.1	26.6 (+15.1)
100	12.7	14.2 (+11.4)
average	55.9	58.9 (+5.3)



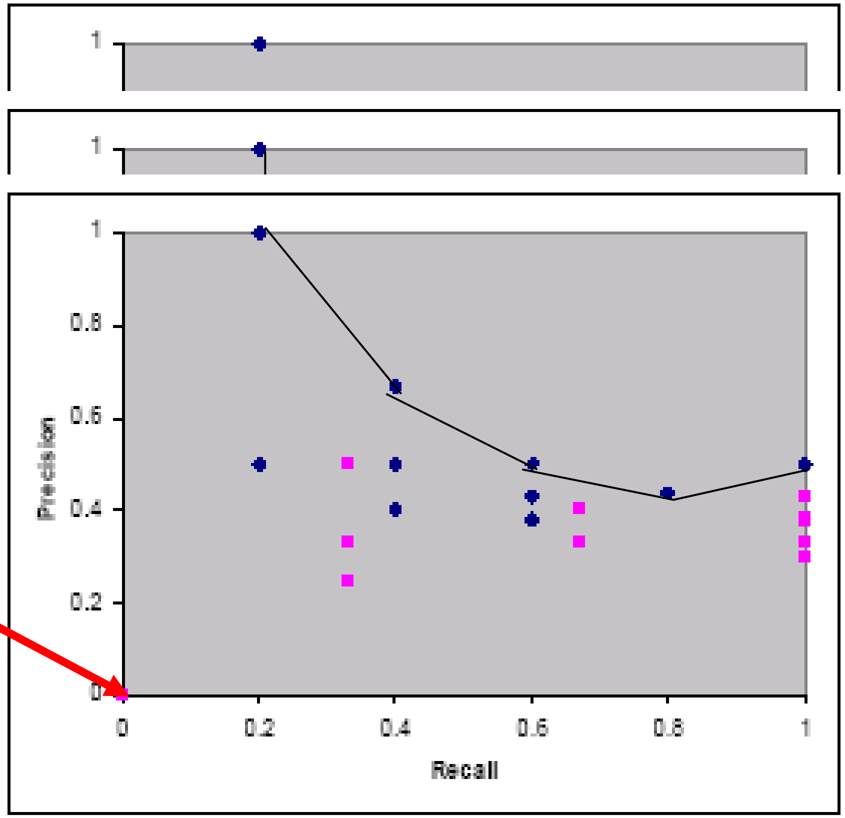
Averaging graphs: a false start

- How can graphs be averaged?
  - Different queries have different meaningful recall values
- Recall/precision graph also has odd saw-shape (锯齿状) if done directly
- Sample graphs (In example 2)
  - What is precision at 25% recall?
  - Need to interpolate
    - But how?



Possible interpolation approaches

- No interpolation
  - Not very useful
- Connect the dots
- Connect max
- Connect min
- Connect average
- …
- How to deal with 0% recall?
  - Assume 0?
  - Assume best?
  - Constant start?



How to choose?

- It is an empirical fact that *on average* as recall increases, precision decreases
  - Verified time and time again
  - *On average*
- Seems reasonable to aim for an interpolation that makes function monotonically decreasing (单调递减)
- One approach:

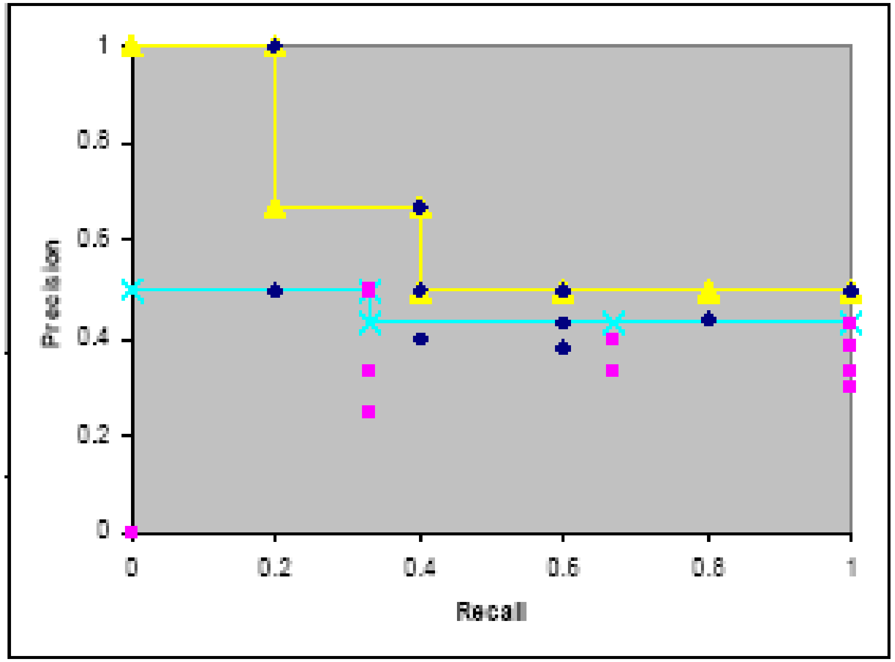
$$P(R) = \max\{P' \;:\; R' \geq R \wedge (R', P') \in S\}$$

-

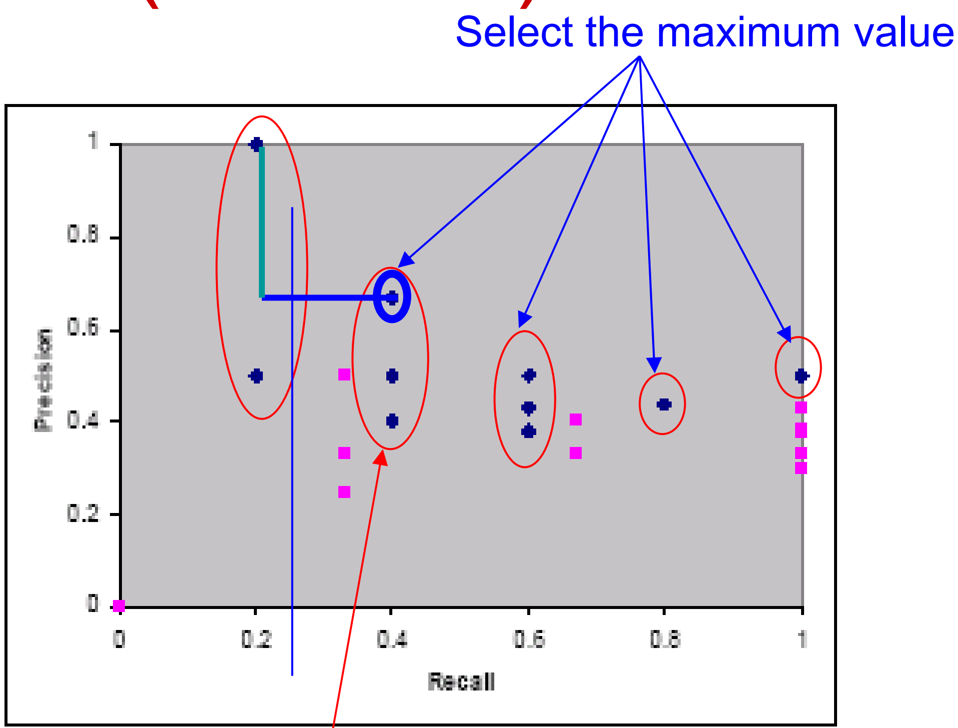
  where S is the set of observed (R,P) points
- Results in a step function

Our example, interpolated this way

- Monotonically drops
- Average will also fall monotonically
- Note R=0.67 and R=0.8
- Handles 0% recall smoothly

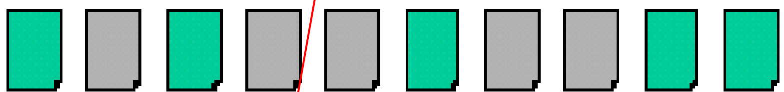


Our example (Cont’ d)



- Given the data by Ranking #1
- What’ s the precision at 0.25 recall?

Ranking #1

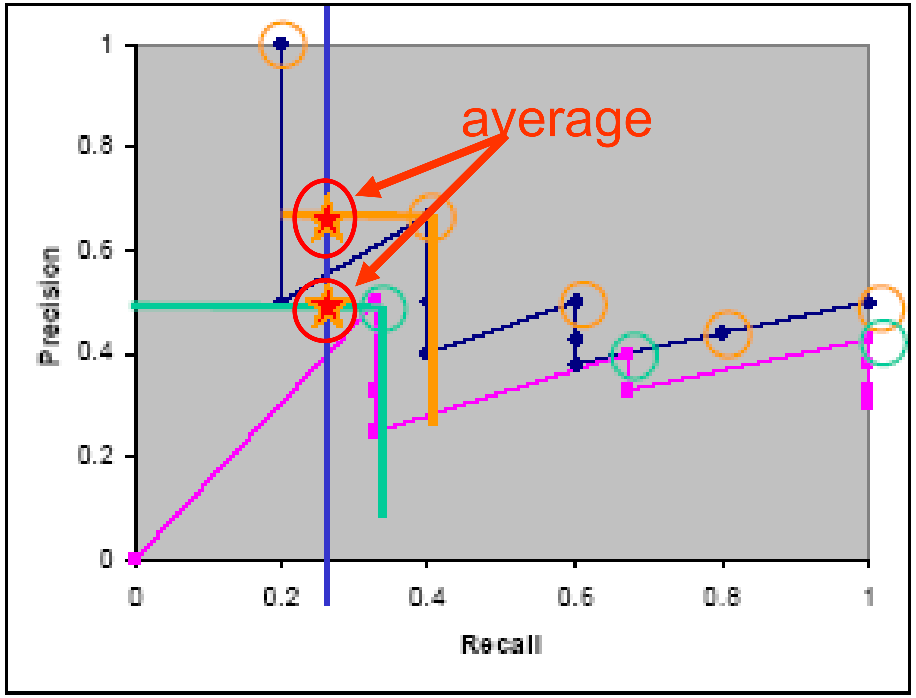


Recall	0.2	0.2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	1.0
Precis.	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5

AvgPrec= 62.2%

Averaging graphs: using interpolation

- How can graphs be averaged?
  - Different queries have different meaningful recall values
- Recall/precision graph also has odd saw-shape if done directly
- Sample graphs (example 2)
- What is precision at 25% recall?
- Interpolate values



Interpolation and averaging

[van Rijsbergen, p. 118 (1979)]

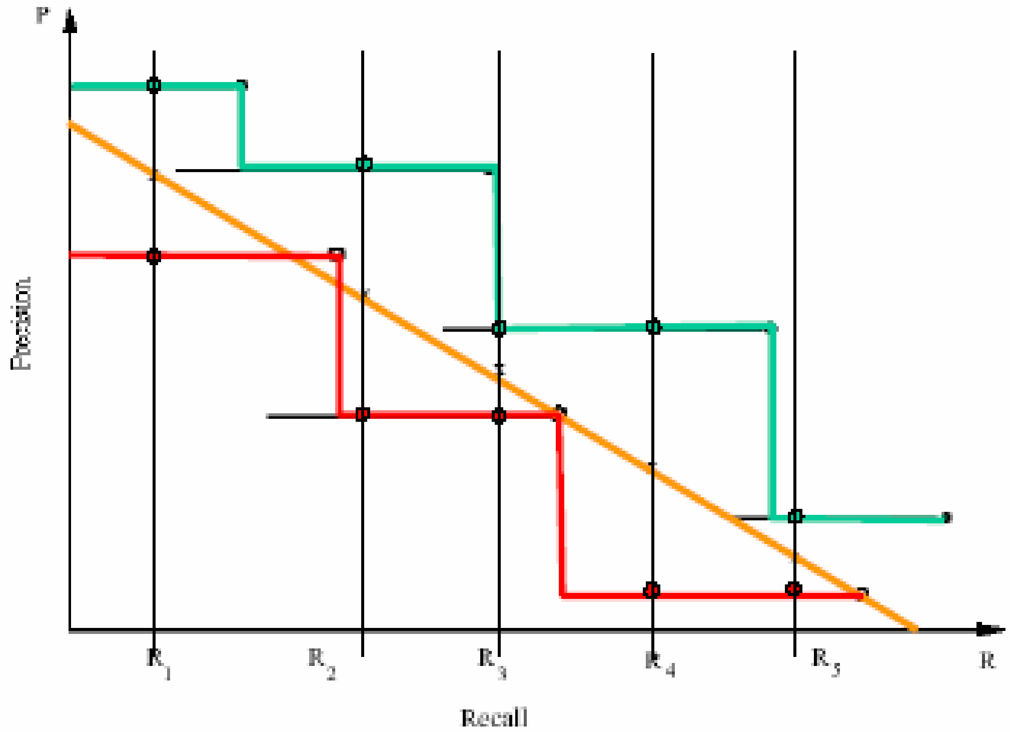


Figure 7.4. An example of macro-evaluation. The points indicated by crosses lie midway between two enclosing horizontal bars and their abscissæ are given by the standard recall values R_i

Interpolated average precision

- Average precision at standard recall points
- For a given query, compute P/R point for every relevant doc.
- Interpolate precision at standard recall levels
  – 11-pt is usually 100%, 90, 80, …, 10, 0% (yes, 0% recall)
  – 3-pt is usually 75%, 50%, 25%
- Average over all queries to get average precision at each recall level
- Average interpolated recall levels to get single result
  – Called “interpolated average precision”
    - Not used much anymore; “mean average precision” (MAP) more common
    - Values at specific interpolated points still commonly used

Evaluation in document retrieval: outline

- *Relevance and test collections*
- Effectiveness measures
  - *Recall and precision*
  - E and F
  - Expected search length
- TREC Conference
- Other issues and problems

More Single-Valued Measures

- E measure (van Rijsbergen)*

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

- Used to emphasize precision (or recall)
  - essentially a weighted average of precision and recall
  - large α increases importance of precision
- Can transform by $\alpha = 1/(\beta^2 + 1)$, $\beta = P/R$

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

  - When $\beta = 1$ ($\alpha = ½$) equal importance of precision and recall
  - Normalized symmetric difference of retrieved and relevant sets

Symmetric Difference and E

- A is the retrieved set of documents
- B is the relevant set of documents

  $P = |A \cap B| \div |A|$

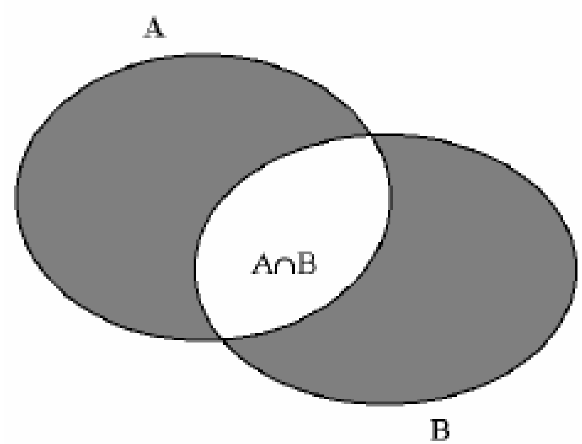
  $R = |A \cap B| \div |B|$

- $A \otimes B$ (the symmetric difference) is the shaded area

  $$|A \otimes B| = |A \cup B| - |A \cap B|$$
  $$= |A| + |B| - 2|A \cap B|$$

- $E_{\beta} = 1 - (2PR \div (P+R))$

  $$= (P+R-2PR) \div (P+R)$$
  $$= \ldots$$
  $$= |A \otimes B| \div (|A| + |B|)$$



F measure

- F = 1- E often used
  - Good results mean larger values of F

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

- “F1” measure is popular: F with β=1
  - Particularly popular with classification researchers

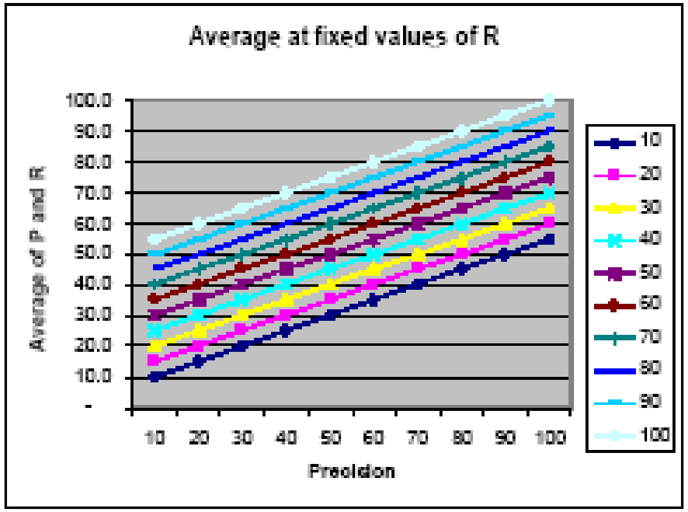
$$F_1 = \frac{2PR}{P + R}$$

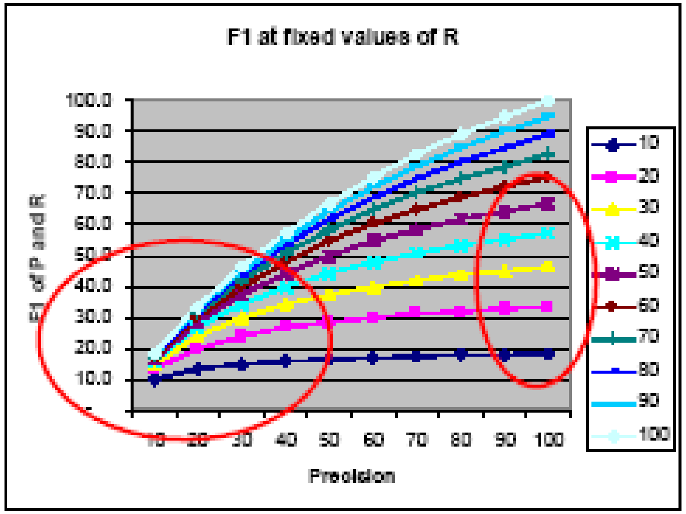
F measure as an average

- Harmonic mean(调和平均) of P and R
  - Inverse of average of their inverses

$$F_1 = \frac{2PR}{P+R} = \frac{1}{\frac{1}{2}\left(\frac{1}{R}+\frac{1}{P}\right)}$$

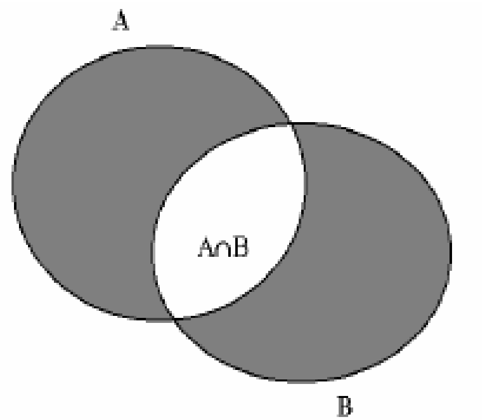
- Heavily penalizes low values of P or R
  - Compared to standard average





F measure, geometric interpretation

- A is the retrieved set of documents
- B is the relevant set of documents
- P = |A∩B| ÷ |A|
- R = |A∩B| ÷ |B|



$$
\begin{aligned}
F_{\beta=1} &= 2PR/(P+R) \\
&= 2\frac{|A \cap B|^2}{|A| \cdot |B|} / \left(|A \cap B| \left(\frac{1}{|A|} + \frac{1}{|B|} \right) \right) \\
&= \frac{2|A \cap B|}{|B| + |A|}
\end{aligned}
$$

Evaluation in document retrieval: outline

- *Relevance and test collections*
- Effectiveness measures
  - *Recall and precision*
  - *E and F*
  - Expected search length
- TREC Conference
- Other issues and problems

Other Single-Valued Measures

- Expected search length*
- Breakeven point (损益平衡点)
  - point at which precision = recall
  - Popular in classification tasks, though not clear what it means
- MRR (Mean Reciprocal Rank)
- Many others...

Expected Search Length

- Evaluation is based on type of information need:
  - 1. only one relevant document required
  - **2. some arbitrary number *n***
  - 3. all relevant documents
  - 4. a given proportion of relevant documents…..
- Two types of ordering
  - Simple ordering: never have two or more documents at the same level of the ordering
  - Otherwise, weak ordering
- *Search length* in a simple ordering
  - the number of non-relevant documents a user must scan before the information need is satisfied
- Search strategy output assumed to be *weak ordering*
  - *Expected search length* appropriate for weak ordering

Expected Search Length

For simple ordering

		1		2																
Rank	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Relevance	N	Y	N	Y	Y	Y	Y	N	Y	N	N	N	Y	N	Y	N	N	N	N	N

For type 2 query with n=2, search length is ?

For query with n=6, search length is ?

For weak ordering

Rank	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	4	4
Relevance	N	N	Y	Y	N	Y	Y	Y	N	Y	Y	N	N	N	N	N	N	N	Y	N
			1	2		3	4	5	X	X	X	X	X							

For type 2 query with n=6, possible search lengths are 3,4,5 or 6 depending on ordering in level 3.

Of the 10 ways in which 2 relevant docs could be distributed in 5, 4 would have search length 3, 3 have search length 4, 2 have search length 5, and 1 has search length 6.

Expected Search Length is ?

Expected Search Length

- ESL(q) = Pnonrel + Fnonrel·Fneeded / (Frel+1)
  - q is the query
  - Pnonrel is the number of documents non-relevant to q in all levels preceding the final
  - Frel is number of relevant documents in final level
  - Fnonrel is number of non-relevant documents in final level
  - Fneeded is the number of relevant documents required from the final level to satisfy the need
- Use mean expected search length for a set of queries
- The measure is criticized for ignoring recall

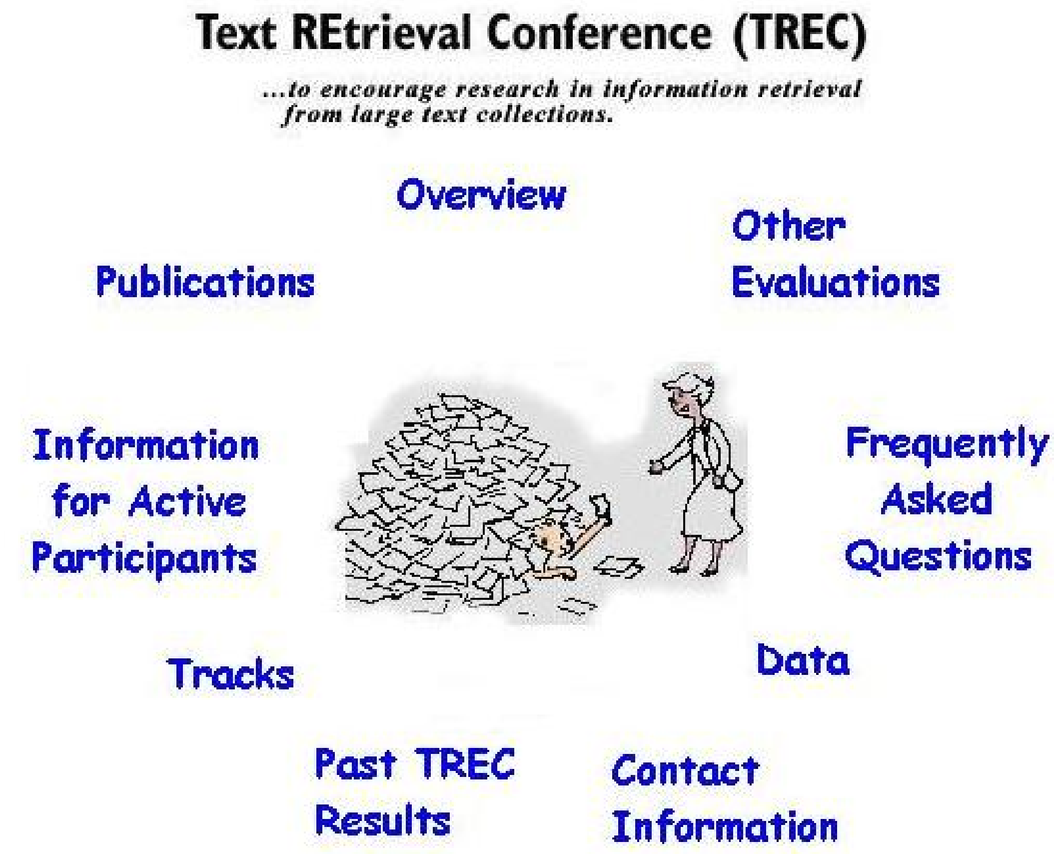
Evaluation Problems

- Retrieval techniques highly collection and query specific
  - Single technique must be tested on multiple collections
  - Comparison of techniques must be on same collection
  - Isolated tests not very useful
- Standard methods assume user knows right collection
- Usually impossible to control all variables with real systems
- Hard to separate effects of retrieval model and interface when model requires user interaction
- Good test collections are very hard (expensive) to produce
- Usually can’t do cost-benefit analysis

Evaluation in document retrieval: outline

- *Relevance and test collections*
- Effectiveness measures
  - *Recall and precision*
  - *E and F*
  - *Expected search length*
- TREC Conference
- Other issues and problems

TREC Conference



TREC Conference (Cont’ d)

- Established in 1992 to evaluate large-scale IR
  - Retrieving documents from a gigabyte to terabytes collection
- Has run continuously since then
  - TREC 2010 conference: **Nov 16-19,** at NIST (National Institute of Standards and Technology) in Gaithersburg, Md. USA
  - Run by NIST’ s Information Access Division
  - Initially sponsored by DARPA as part of Tipster program
  - Now supported by many, including DARPA, ARDA, and NIST
- Probably most well known IR evaluation setting
  - Started with 25 participating organizations in 1992 evaluation
  - In 2007, there were about 87 groups all over the world.
- Proceedings available on-line (http://trec.nist.gov)
  - Overview and call for participation information of TREC 2010 at
  - http://trec.nist.gov/call2010.html

TREC general format

- TREC consists of IR research tracks
  - –Ad-hoc retrieval (web track, up to one billion Web pages for 2010), routing, cross-language, scanned documents, speech recognition, query, video, filtering, Spanish, question answering, novelty, Chinese, high precision, interactive, Web, database merging, NLP, …
- Each track works on roughly the same model
  - November: track approved by TREC community
  - Winter: track’s members finalize format for track
  - Spring: researchers train system based on specification
  - Summer: researchers carry out formal evaluation
- Usually a “blind” evaluation: researchers do not know answer
  - Fall: NIST carries out evaluation
  - November: Group meeting (TREC) to find out:
    - How well your site did
    - How others tackled the problem
  - Many tracks are run by volunteers outside of NIST (e.g., Web)
- “Coopetition(竞争中的合作)” model of evaluation
  - Successful approaches generally adopted in next cycle

TREC: pros and cons

- Widely recognized, premier annual IR evaluation
- What is good
  - Brings together a wide range of active researchers
  - Huge distributed resources applied to common task
  - Substantial gains on tasks rapidly
  - Valuable evaluation corpora (语料库) usually available after track completes
- What is less good
  - Annual evaluation can divert resources from research
    - Evaluations often require significant engineering effort
    - Some tracks moving to bi-annually evaluation as a result
  - Recently, an explosion of tracks
    - Means less energy applied to individual tasks
    - TREC program committee keeps a tight rein on number of tracks
- On balance?
  - Depends on your prejudices

Homework 5

Backup

Why significance tests?

- System A beats System B on one query
  - Is it just a lucky query for System A?
  - Maybe System B does better on some other query
  - Need as many queries as possible
    - Empirical research suggests 25 is minimum needed
    - TREC tracks generally aim for at least 50 queries
- System A and B identical on all but one query
  - If System A beats System B by enough on that one query, average will make A look better than B
- As above, could just be a lucky break for System A
  - Need A to beat B frequently to believe it is really better
- E.g. system A is only 0.00001% better than System B
  - Even if it’s true on every query, does it mean much?
- Significance tests consider those issues

Sign Test Example

- For techniques A and B, compare average precision for each pair of results generated by queries in test collection
- If difference is large enough, count as + or -, otherwise ignore
- Use number of +’s and the number of significant differences to determine significance level
- For example, for 40 queries…
  - Technique A produced a better result than B 12 times
  - B was better than A 3 times
  - And 25 were “the same” …
  - $p < 0.035$ and technique A *is* significantly better than B at the 5% level
  - If A>B 18 times and B>A 9 times…
  - $p < 0.122$ and A is *not* significantly better than B at the 5% level (Chi-square test)

$$\chi^2 = \frac{\left(|n_+ - n_-| - 1\right)^2}{n_+ + n_-}$$

Where n_+ is the times that A performances better than B,
n_- is the times that B performances better than A,
the value p should be queried from the χ^2 test table.
(See attached file “x2检验.mht” for more info.)

Evaluation in document retrieval: outline

- *Types of evaluation*
- *Relevance and test collections*
- *Effectiveness measures*
  - *Recall and precision*
  - *E and F*
  - *Expected search length*
- *Significance tests*
- Other issues and problems

Feedback Evaluation

- Relevance feedback covered later
  - Two-pass approaches
  - Create better query out of results from original query
- How to treat documents that have been seen before?
  - Rank freezing
    - Ranks of relevant documents fixed for subsequent iterations
    - Compare ranking with original ranking
    - Performance can't get worse
  - Residual collection
    - All previously seen documents (e.g. top n) removed from collection
    - Compare reranking with original ranking (n+1…D)
- Both approaches problematic
  - Users probably want to see good documents move to top of list

User Perceptions

- Effectiveness measures give quality of retrieved list
- Other measures important
  - Time to complete a retrieval task
  - User "satisfaction"
  - How well users believe system works
- An "intelligent" IR system is one that does not look stupid to the users
- User studies difficult to design and expensive to conduct
- Hard to isolate effects of search engine and user interface
- Hard to control for individual performance differences
- TREC "interactive" track

Computational Aspects

- Most models give theoretical bounds on costs for query evaluation and collection building
- For large collections
  - Evaluation must be “nearly” independent of collection size
  - Building time should be no worse than linear
- Staged retrieval
  - Use low cost model to get a set of potentially relevant documents
  - Apply more sophisticated techniques to refine or organize the retrieved set
- Tradeoff between cost and discrimination power
- Optimization a key issue with terabyte-sized collections

Swets’ criteria

- “Properties of a desirable measure of retrieval performance”
  - Solely based on the ability of the retrieval system to distinguish between wanted and unwanted items (not efficiency)
  - Should express discrimination power independent of any “acceptance criterion” employed by system or user
  - Measure should be a single number
  - Should allow complete ordering of different performances, indicate the amount of difference, and assess performance in absolute terms