Evaluating Search Engines

CISC489/689-010, Lecture #10

Monday, March 16th

Ben Carterette

IR Basics in 2 Minutes

- Indexing:
  - Parsing, tokenizing, stopping, stemming
  - Compression
  - Inverted lists, vocabulary, collection
- Retrieval:
  - Query processing
  - Retrieval models and scoring documents
- Once the documents are scored and ranked, how do we know whether the system is any good?

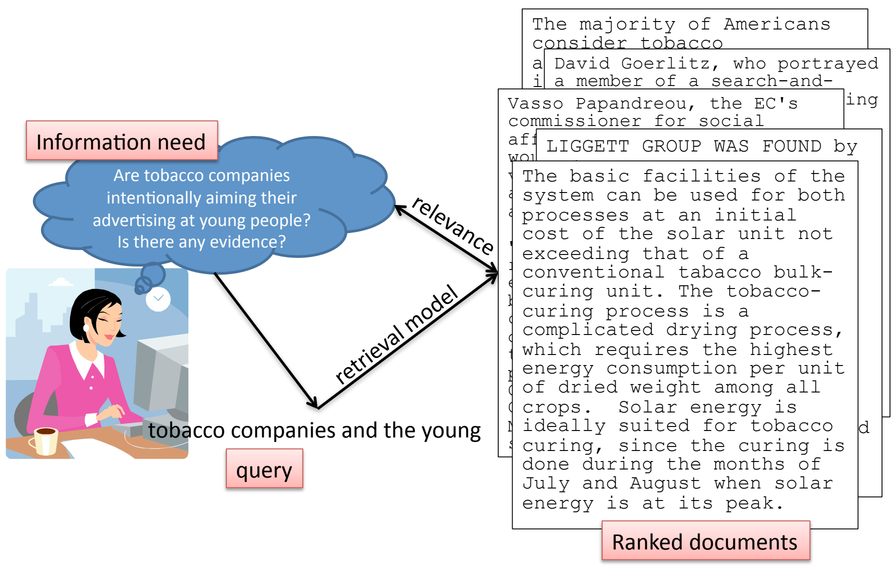
Evaluation

- Evaluation is key to building *effective* and *efficient* search engines
  - measurement usually carried out in controlled laboratory experiments
- Two types of evaluation:
  - User studies: bring in users to interact with engine, measure their responses
  - System-based: have assessors judge the relevance of documents, use judgments to calculate effectiveness measures

Relevance Judgments

- An engine returns a list of documents ranked by score
  - The documents it "thinks" are relevant
- How do we know which are actually relevant and which are not?
  - The person that posed the original query should judge them
  - Indicate whether each document is relevant and how relevant it is

Relevance Judgments



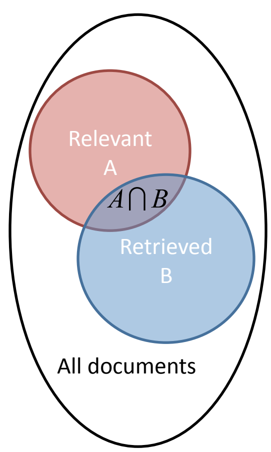
Precision and Recall

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

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

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

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



Classification Errors

- *False Positive* (Type I error)
  - a non-relevant document is retrieved

$$Fallout = \frac{|\overline{A} \cap B|}{|\overline{A}|}$$

- *False Negative* (Type II error)
  - a relevant document is not retrieved
  - 1- *Recall*
- *Precision* is used when probability that a positive result is correct is important

F Measure

- *Harmonic mean* of recall and precision

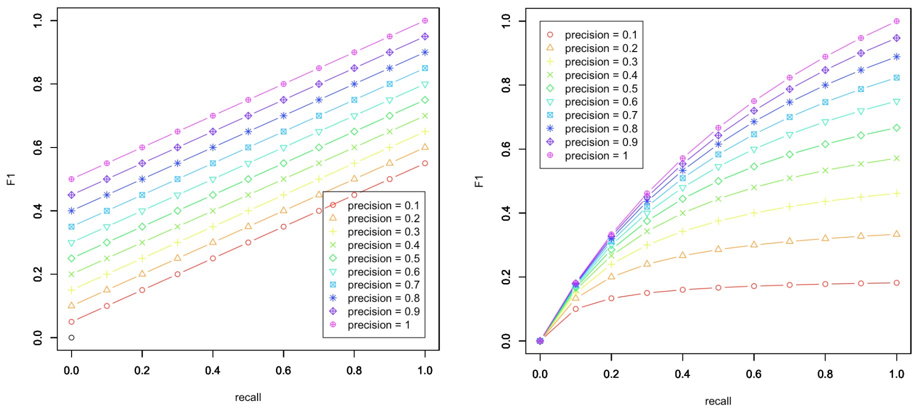
$$F = \frac{1}{\frac{1}{2}(\frac{1}{R} + \frac{1}{P})} = \frac{2RP}{(R+P)}$$

  - harmonic mean emphasizes the importance of small values, whereas the arithmetic mean is affected more by outliers that are unusually large
- More general form

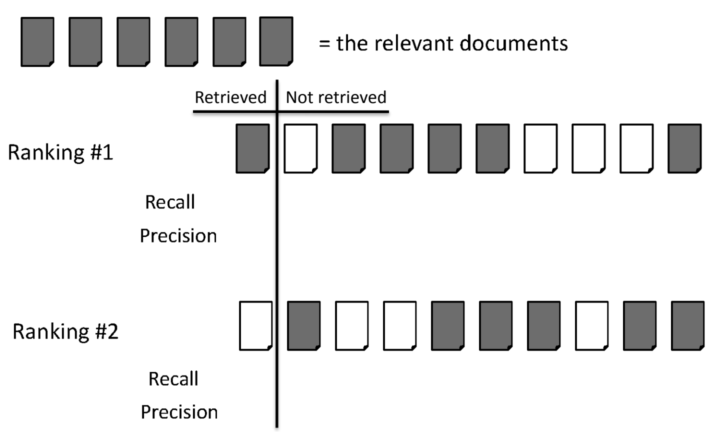
$$F_\beta = (\beta^2 + 1)RP/(R + \beta^2 P)$$

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

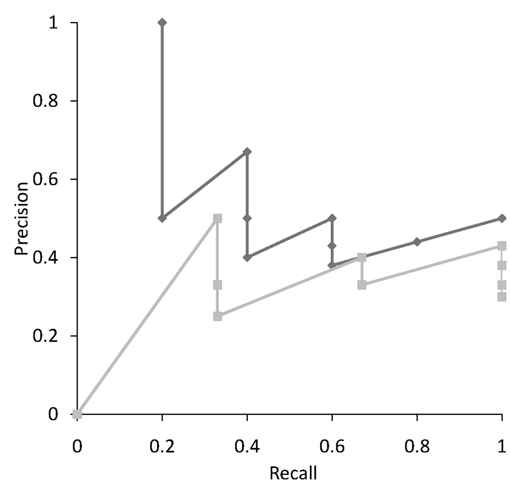
F Measure versus Arithmetic Mean



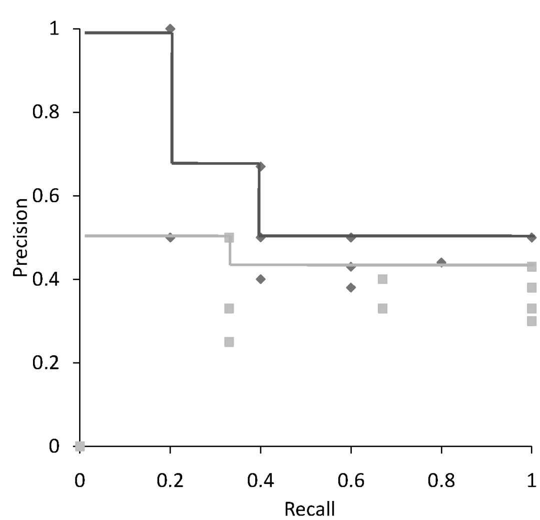
Precision and Recall



Recall-Precision Graph



Interpolation



Interpolation

- To average graphs, interpolate precision at recall level *R*:

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

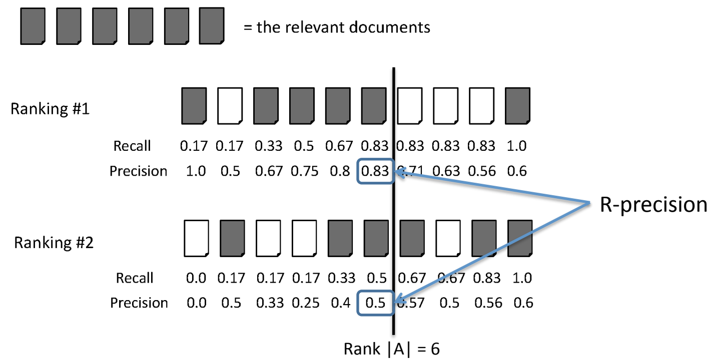
  - where *S* is the set of observed (*R,P*) points
- Defines precision at any recall level as the *maximum* precision observed in any recall-precision point at a higher recall level
  - produces a step function
  - defines precision at recall 0.0
- Why maximum? Why not minimum or average?

Summarizing a Ranking

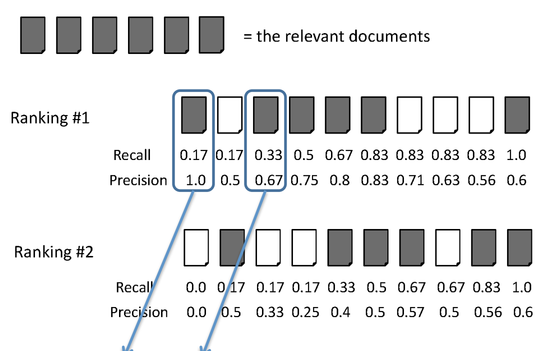
- *Average* precision values over particular ranks or recall points
  - Recall and precision at fixed rank positions
  - Precision at standard recall levels, from 0.0 to 1.0
    - requires *interpolation*
  - Averaging the precision values from the rank positions where a relevant document was retrieved
    - i.e. rank positions at which recall increases

R-Precision

- Precision at rank |A|
  - |A| = the total number of relevant documents



Average Precision



$$\text{Ranking } \#1\text{: } (\boxed{1.0} + \boxed{0.67} + 0.75 + 0.8 + 0.83 + 0.6)/6 = 0.78$$

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

Focusing on Top Documents

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

Focusing on Top Documents

- Precision at Rank k
  – k typically 5, 10, 20
  – easy to compute and understand
  – not sensitive to rank positions less than k
- Reciprocal Rank
  – reciprocal of the rank at which the first relevant document is retrieved
  – very sensitive to rank position

Discounted Cumulative Gain

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

Discounted Cumulative Gain

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

Discounted Cumulative Gain

- *DCG* is the total gain accumulated up to a particular rank *p*:

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

- Alternative formulation:

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

gain

discount

  – used by some web search companies
  – emphasis on retrieving highly relevant documents

DCG Example

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

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

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

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

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

Normalized DCG

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

NDCG Example

- Perfect ranking:

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

  3, 6, 7.89, 8.89, 9.75, 10.52, 10.88, 10.88, 10.88, 10
- NDCG values (divide actual by ideal):

  1, 0.83, 0.87, 0.76, 0.71, 0.69, 0.73, 0.8, 0.88, 0.88
  - NDCG ≤ 1 at any rank position

Using Preferences

- Two rankings described using preferences can be compared using the *Kendall tau coefficient (τ):*

$$\tau = \frac{P - Q}{P + Q}$$

  - *P* is the number of preferences that agree and *Q* is the number that disagree
- For preferences derived from binary relevance judgments, can use *BPREF*

BPREF

- For a query with *R* relevant documents, only the first *R* non-relevant documents are considered

$$BPREF = \frac{1}{R}\sum_{d_r}(1 - \frac{N_{d_r}}{R})$$

  - d_r is a relevant document, and N_{dr} gives the number of non-relevant documents
- Alternative definition

$$BPREF = \frac{P}{P+Q}$$

Evaluation Measures Summary

- Precision at rank k
- Recall at rank k
- F at rank k
- Precision-recall curve
  - Interpolated precision-recall curve
- Average precision
- R-precision
- Reciprocal rank
- Discounted cumulative gain (DCG)
  - Normalized version (NDCG)

Averaging Over Queries

- What if the query I am evaluating is “easy”?
  - i.e. every engine would do well on it
- Or if it’s “hard”?
  - i.e. every engine would do poorly
- What if I intentionally pick a query that’s easy for one engine and hard for another?
  - Is that a valid comparison?
- Instead, evaluate over a set of queries
- Calculate evaluation measures for each query and average over the set

Averaging Across Queries

= relevant documents for query 1

Ranking #1

Recall	0.2	0.2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	1.0
Precision	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5

= relevant documents for query 2

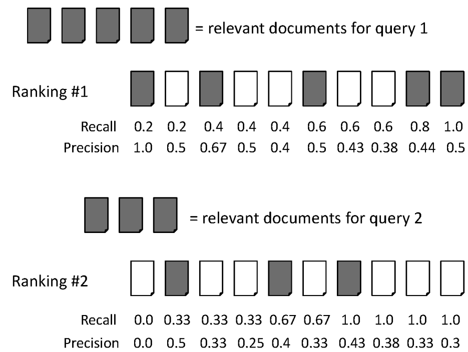
Ranking #2

Recall	0.0	0.33	0.33	0.33	0.67	0.67	1.0	1.0	1.0	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.33	0.43	0.38	0.33	0.3
Average over 2 queries	0.1 0.5	0.27 0.5								

Averaging

- *Mean Average Precision* (MAP)
  - summarize rankings from multiple queries by averaging average precision
  - most commonly used measure in research papers
  - assumes user is interested in finding many relevant documents for each query
  - requires many relevance judgments in text collection
- Recall-precision graphs are also useful summaries

MAP



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

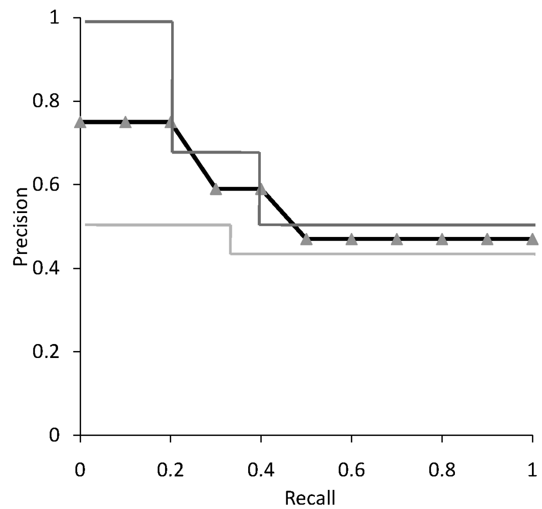
$$\textit{mean average precision} = (0.62 + 0.44)/2 = 0.53$$

Average Precision at Standard Recall Levels

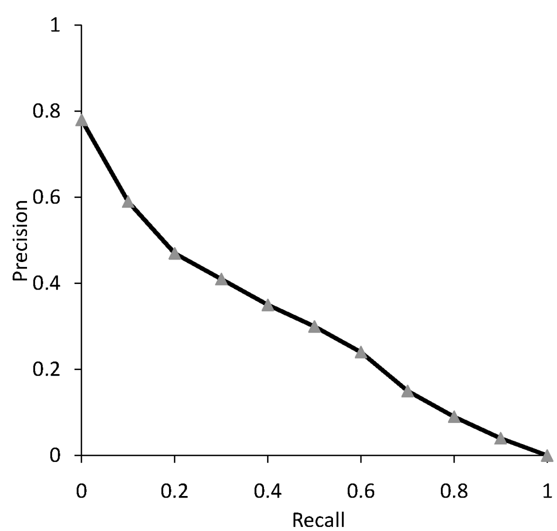
Recall	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Ranking 1	1.0	1.0	1.0	0.67	0.67	0.5	0.5	0.5	0.5	0.5	0.5
Ranking 2	0.5	0.5	0.5	0.5	0.43	0.43	0.43	0.43	0.43	0.43	0.43
Average	0.75	0.75	0.75	0.59	0.47	0.47	0.47	0.47	0.47	0.47	0.47

- Recall-precision graph plotted by simply joining the average precision points at the standard recall levels

Average Recall-Precision Graph



Graph for 50 Queries



Efficiency Metrics

Metric name	Description
Elapsed indexing time	Measures the amount of time necessary to build a document index on a particular system.
Indexing processor time	Measures the CPU seconds used in building a document index. This is similar to elapsed time, but does not count time waiting for I/O or speed gains from parallelism.
Query throughput	Number of queries processed per second.
Query latency	The amount of time a user must wait after issuing a query before receiving a response, measured in milliseconds. This can be measured using the mean, but is often more instructive when used with the median or a percentile bound.
Indexing temporary space	Amount of temporary disk space used while creating an index.
Index size	Amount of storage necessary to store the index files.

Two Types of Evaluation

- System-based
  - Bring in people to judge the relevance of retrieved documents
  - Use those judgments to calculate measurements about system performance
- User-based
  - Bring in people to try out the search engine
  - Ask them whether they liked it, or measure their performance on some task

User versus System Evaluation

User-Based

- More expensive: every system change requires a new user study to evaluate
- More realistic: users are actually using the engine; provide real feedback
- More variance: users are not all able to use engines equally well
- More valid: if set up correctly, users can't bias results
- **Harder**

System-Based

- Less expensive: after changing the system, use the same judgments
- Less realistic: no users involved; have to trust judgments
- Less variance: variance only comes from queries; can easily be decreased
- Less valid: researcher or developer can bias results
- **Easier**

Online Testing

- Test using live traffic on a search engine
- Benefits:
  - real users, less biased, large amounts of test data
- Drawbacks:
  - noisy data, can degrade user experience
- Often done on small proportion (1-5%) of live traffic
- A "happy medium" between user- and system-based evaluations

Query Logs

- Used for both tuning and evaluating search engines
  - also for various techniques such as query suggestion
- Typical contents
  - User identifier or user session identifier
  - Query terms - stored exactly as user entered
  - List of URLs of results, their ranks on the result list, and whether they were clicked on
  - Timestamp(s) - records the time of user events such as query submission, clicks

Query Logs

- Clicks are not relevance judgments
  - although they are correlated
  - biased by a number of factors such as rank on result list
- Can use clickthough data to predict *preferences* between pairs of documents
  - appropriate for tasks with multiple levels of relevance, focused on user relevance
  - various “policies” used to generate preferences

Example Click Policy

- *Skip Above and Skip Next*
  - click data

$$\begin{array}{l} d_1 \\ d_2 \\ d_3 \ \text{(clicked)} \\ d_4 \end{array}$$

  - generated preferences

$$\begin{array}{l} d_3 > d_2 \\ d_3 > d_1 \\ d_3 > d_4 \end{array}$$

Query Logs

- Click data can also be aggregated to remove noise
- *Click distribution* information
  - can be used to identify clicks that have a higher frequency than would be expected
  - high correlation with relevance
  - e.g., using *click deviation* to filter clicks for preference-generation policies

Filtering Clicks

- *Click deviation CD(d, p)* for a result *d* in position *p*:

$$CD(d,p) = O(d,p) - E(p)$$

O(d,p): observed click frequency for a document in a rank position p *over all instances of a given query*

E(p): expected click frequency at rank p *averaged across all queries*

Drawbacks of Log-Based Evaluation

- Difficult to evaluate recall-based measures
  - Users only click on high-ranked documents
  - Difficult to discover relevant documents that the engine is not currently ranking highly
- Difficult to evaluate "tail queries"
  - 40% of queries only appear once in the log
  - No information to aggregate over
- Interdependence between items on a page complicates analysis
  - Ads vs. search results; quality of result at rank 2 versus quality of result at rank 1; etc.