Information Retrieval Evaluation

(COSC 488)

Nazli Goharian
nazli@cs.georgetown.edu

Measuring Effectiveness

- An algorithm is deemed incorrect if it does not have a “right” answer.
- A heuristic tries to guess something close to the right answer. Heuristics are measured on “how close” they come to a right answer.
- IR techniques are essentially heuristics because we do not know the right answer.
- So we have to measure how *close* to the right answer we can come.

Experimental Evaluations

- Batch (ad hoc) processing evaluations
  - Set of queries are run against a static collection
  - Relevance judgments identified by human evaluators are used to evaluate system
- User-based evaluation
  - Complementary to batch processing evaluation
  - Evaluation of users as they perform search are used to evaluate system (time, clickthrough log analysis, frequency of use, interview,…)

3

Some of IR Evaluation Issues

- How/what data set should be used?
- How many queries (topics) should be evaluated?
- What metrics should be used to compare systems?
- How often should evaluation be repeated?

4

Existing Testbeds mainly used for Academic Research

- Cranfield (1970): A small (megabytes) domain specific testbed with fixed documents and queries, along with an exhaustive set of relevance judgment

- TREC (Text Retrieval Conference- sponsored by NIST; starting 1992): Various data sets for different tasks.
  - Most use 25-50 queries (topics)
  - Collections size (2GB, 10GB, half a TByte (GOV2), …….and 25 TB ClueWeb)
  - No exhaustive relevance judgment

5

Existing Testbeds (Cont'd)

- GOV2 (Terabyte):
  - 25 million pages of web; 100-10,000 queries; 426 GB
- Genomics:
  - 162,259 documents from the 49 journals; 12.3 GB
- ClueWeb09 :
  - 1 billion web pages (ten languages)
- ClueWeb12:
  - 870 million English web pages
- Text Classification datasets:
  - Reuters-21578 (newswires)
  - Reuters RCV1 (806,791 docs),
  - 20 Newsgroups (20,000 docs; 1000 doc per 20 categories)
  - Others: WebKB (8,282), OHSUMED(54,710), GENOMICS (4.5 million),….

6

TREC

- Text Retrieval Conference- sponsored by NIST
- Various benchmarks for evaluating IR systems.
- Sample tasks:
  - Ad-hoc: evaluation using new queries
  - Routing: evaluation using new documents
  - Other tracks: CLIR, Multimedia, Question Answering, Biomedical Search, etc.
  - For more info see: http://trec.nist.gov/

TREC Relevance Information & Pooling

- TREC uses *pooling* to approximate the number of relevant documents and identify these documents, called *relevance judgments (qrels)*
- For this, TREC maintains a set of documents, queries, and a set of relevance judgments that list which documents should be retrieved for each query (*topics*)
- In *pooling*, only top documents returned by the participating systems are evaluated, and the rest of documents, even relevant, are deemed non-relevant

8

Problem…

- Building larger test collections along with <u>complete relevance judgment</u> is difficult or impossible, as it demands assessor time and many diverse retrieval runs.

9

Evaluating Various Search tasks

- TREC evaluation paradigm, using *Pooling*, has shown success for specific user task of *topical information* (*ad hoc*).

- Other users tasks:
  - *Navigational:* finding specific sites
  - *Transactional:* finding specific item (buy books, etc.)

➔Not dealing with set of relevant documents but with rather a single correct answer!

10

Logging

- Search companies utilize query logs containing user interaction with a search engine
- Much more data available
- Privacy issues need to be considered
- Relevance judgment done via
  - Using clickthrough data -- biased towards highly ranked pages or pages with good snippets
  - Page dwell time

Evaluating Web Search Engines

- Dynamic environment (Facts):
  - Collection grows/changes rapidly and indicies are constantly updated
  - User interests and popular queries change
  - Web queries are typically short (1-3 terms), thus difficult to capture users' need
  - Search algorithms are continually refined
  - Users only view top 10 results for 85% of their queries
  - Users do not revise their query after the first try for 75% of their queries
  - Majority of queries occur only a few times (55% occurs less than 5 times)
  - Top queries are changing over time too.

Evaluating Web Search Engines (Cont'd)

- Web is too large to calculate recall, thus need measures that are not recall-based
- Hundreds of millions of queries per day, thus need large sample of queries to represent the population of even one day
- Repeat evaluations frequently

13

Measures in Evaluating IR

- *Recall* is the fraction of relevant documents retrieved from the set of total relevant documents collection-wide. Also called *true positive rate*.
- *Precision* is the fraction of relevant documents retrieved from the total number retrieved.

14

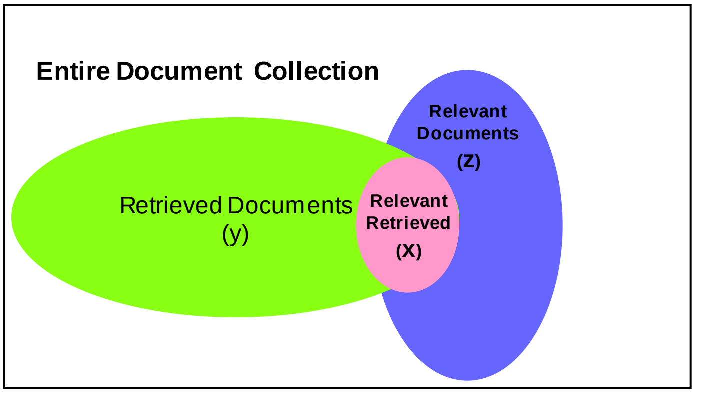
Precision / Recall

Precision

x / y

Recall

x / z



15

Precision / Recall Example

- Consider a query that retrieves 10 documents.
- Lets say the result set is.

  D1
  D2
  D3
  D4
  D5
  D6
  D7
  D8
  D9
  D10

- With all 10 being relevant, Precision is 100%
- Having only 10 relevant in the whole collection, Recall is100%

16

Example (continued)

- Now lets say that only documents two and five are relevant.
- Consider these results:

  D1
  D2
  D3
  D4
  D5
  D6
  D7
  D8
  D9
  D10

- Two out of 10 retrieved documents are relevant thus, precision is 20%. Recall is (2/total relevant) in entire collection.

17

Levels of Recall

- If we keep retrieving documents, we will ultimately retrieve all documents and achieve 100 percent recall.
- That means that we can keep retrieving documents until we reach x% of recall.

18

Levels of Recall (example)

- Retrieve top 2000 documents.
- Five relevant documents exist and are also retrieved.

DocId	Recall	Precision
100	.20	.01
200	.40	.01
500	.60	.006
1000	.80	.004
1500	1.0	.003

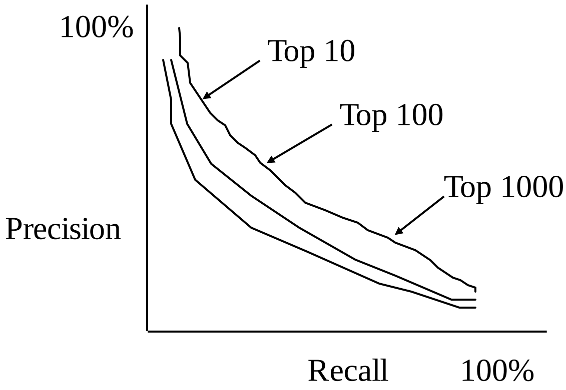
19

Recall / Precision Graph

- Compute precision (interpolated) at 0.0 to 1.0, in intervals of 0.1, levels of recall.
- Optimal graph would have straight line -- precision always at 1, recall always at 1.
- Typically, as recall increases, precision drops.

20

Precision/Recall Tradeoff



21

Search Tasks

- Precision-Oriented (such as in web search)
- Recall-Oriented (such as analyst task)

  number of relevant documents that can be identified in a time frame. Usually ~5 minutes time frame is chosen.

22

More Measures…

- *F Measure – trade off precision versus recall*

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

• Balanced *F Measure* considers equal weight on Precision and Recall:

$$F_{\beta=1} = \frac{2PR}{P+R}$$

23

More Measures…

- MAP (Mean average Precision)
  - Average Precision – Mean of the precision scores for a single query after each relevant document is retrieved.

  * Commonly 10-points of recall is used!
  - MAP is the mean of average precisions for a query batch
- P@10 - Precision at 10 documents retrieved (in Web searching). Problem: the cut-off at *x* represents many different recall levels for different queries - also P@1. (P@x)
- R-Precision – Precision after R documents are retrieved; where R is number of relevant documents for a given query.

24

Example

- For Q1: D2 and D5 are only relevant:
  D1, D2, D3 not judged, D4, D5, D6, D7, D8, D9, D10
- For Q2: D1, D2, D3 and D5 are only relevant:
  D1, D2, D3, D4, D5, D6, D7, D8, D9, D10

P of Q1: 20%

AP of Q1: (1/2 + 2/5)/2 = 0.45

P of Q2: 40%

AP of Q2: (1+1+1+4/5)/4 = 0.95

MAP of system: $(AP_{q1} + AP_{q2})/2 = (0.45 + 0.94)/2 = 0.69$

P@1 for Q1: 0; P@1 for Q2: 100%;

R-Precision Q1: 50%; Q2: 75%

Example

- For Q1: D2 and D5 are only relevant:
  D1, D2, D3 not judged, D4, D5, D6, D7, D8, D9, D10
- For Q2: D1, D2, D3 and D5 are only relevant:
  D1, D2, D3, D4, D5, D6, D7, D8, D9, D10

Recall points P_{Q1}	0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.4
Recall points P_{Q2}	0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 0.8
$AP_{Q1\&2}$ $MAP_{Q1\&2}$	0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.75 0.6 0.73

More Measures…

Discounted Cumulative Gain (DCG)

- Another measure (Reported to be used in Web search) that considers the *top ranked* retrieved documents.
- Considers the *position* of the document in the result set (*graded relevance*) to measure *gain* or *usefulness*.
  - The lower the position of a relevant document, less useful for the user
  - Highly relevant documents are better than marginally relevant ones
  - The gain is accumulated starting at the top at a particular rank *p*
  - The gain is discounted for lower ranked documents

33

Normalized Discounted Cumulative Gain (NDCG)

- Manual relevance is given to the retrieved documents as 0-3 (0=non-relevant, 3=highly relevant)

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

- Generally *normalized* using the *ideal DCG*, $IDCG_p$, defined as the ordered documents in the decreasing order of relevance.

$$nDCG_p = \frac{DCG_p}{IDCG_p}$$

- Generally is calculated over a set of queries

34

nDCG (Example)

- d1, d2, d3, d4, d5 (in the order of their rank)
- Relevance: 3, 3, 1, 0, 2
- DCG_p = 3 + (3/1 + 1/1.59 + 0 + 2/2.32)=7.49
- Ideal order based on relevance: 3,3,2,1,0
- IDCG = 3 + (3/1 + 2/1.59 + 1/ 2 + 0) = 7.75
- $nDCG_p$ = DCG/IDCG = 7.49/7.75 = 0.96

35

Known-item Search Evaluation

- Ranking the best site or item being searched
  - find a single known resource for a given query. Closer the rank of the item to the top, better for the user.
  - Evaluation Metric: Mean Reciprocal Ranking (MRR)
    - Weight of item (correct answer) in location 1 is 1
    - Weight of item in location n is 1/n

$$MRR = \frac{\sum_{q=1}^{n} \frac{1}{rankq}}{n}$$

36

Known-Item Search & MRR

$$MRR = \frac{\sum_{q=1}^{n} \frac{1}{rankq}}{n}$$

Example:

– MRR=0.25 means on average the system finds the known-item in position number 4 of result set.

– MRR= 0.75 means finding the item between ranks 1 and 2 on average.

Cost of Manual Evaluation

Search engines: 5

Queries: 300

Top documents: 20

Time to evaluate each result: 30 seconds (optimistic)

➔(300q * 20r * 5s) = 30,000 results to evaluate

➔10.4 days to complete the task (not sleeping!)

➔31 days (8-hour working days) to complete

➔➔ Not scalable to dynamic env. such as Web!

(Research in progress!)

Measuring Efficiency

- Indexing time
- Indexing temporary space
- Index size
- Query throughput (number of queries processed per second)
- Query latency (time taken in milliseconds till a user query is answered)