Test Collections and IR Experimentation

CISC489/689-010, Lecture #11

Wednesday, March 18th

Ben Carterette

Last Time

- Search engine evaluation, esp. system-based evaluation
- Measures:
  - Precision
  - Recall
  - Average precision
  - R-precision
  - DCG
- All measures require *relevance judgments*
- Average over a set of queries

IR Experimentation

- Comparing different engines requires a controlled experimental setting
- Many different factors influence engine effectiveness:
  - Corpus, queries, relevance judgments
  - Parsing decisions, stop list, stemming algorithm
  - Indexing method, compression algorithm, query processing method
  - Retrieval model, model features, model parameters
- These should be controlled to the greatest extent possible to answer the relevant questions

Evaluation Corpus

- A *test collection* consists of:
  - a corpus of documents or other things to search
  - a set of queries with underlying information needs
  - relevance judgments on documents
- The use of test collections and system-based effectiveness evaluation is called the *Cranfield methodology.*
  - Named after Cranfield Aeronautics, where it was invented in the 60s.

Constructing a Test Collection

- Where do the documents come from?
  - Depends on task, domain, availability, …
- Where do the queries come from?
  - Query logs, users who can describe their information need, …
- What do the queries and information needs look like?
  - Depends on task, domain, …
- Where do the relevance judgments come from?
  - Assessors judge documents w.r.t. information needs

Corpus Examples

- News corpora
  - AP: Associated Press articles from 1988-1992.
  - TDT: News articles from AP, Wall Street Journal, NY Times, LA Times, plus audio transcripts
- Web corpora
  - GOV2: Web pages downloaded from .gov domain in 2004
  - Wikipedia: Top 10% of Wikipedia pages
- Genomics corpora
  - Medical journals, clinical reports

Relevance Judgments

- Assessors look at documents and decide whether they're relevant to the information need
- Recall *recall*:
  - # relevant & retrieved / # relevant
  - Proportion of relevant documents that were retrieved
- How do we know the # relevant?
- It is not sufficient to limit search to documents that contain query terms
- Assessors need to read *every* document

Relevance Judgments

- Obtaining relevance judgments is an expensive, time-consuming process
  - who does it?
  - what are the instructions?
  - what is the level of agreement?
- It's not possible to get a judgment on every document
  - and therefore not possible to really know any recall-based measure
- How can we best target judging for efficient and accurate evaluation?

Pooling

- System pooling is often used to select documents
  - top *k results* from the rankings obtained by different search engines (or retrieval algorithms) are merged into a pool
  - duplicates are removed
  - documents are presented in some random order to the relevance judges
- Ensures that judging is targeted to the documents that are least likely to be nonrelevant
- Produces a large number of relevance judgments for each query, although still incomplete

Interactive Searching and Judging

- An assessor uses a search engine to find relevant information
  - Submits a query and judges retrieved documents
- As they learn about the topic, they reformulate queries and resubmit
- Over a series of reformulations, the assessor builds a collection of relevance judgments

Algorithmic Approaches

- Move-to-Front Pooling
  - Target judging to the systems that are returning a lot of relevant documents
  - Goal is to find as many relevant documents as possible
- Minimal Test Collections
  - Target judging to the documents that are most informative for *comparing* systems
  - No preference for relevant or non relevant documents as long as the document carries a lot of information

Statistical Approaches

- Statistical sampling
  - Use a sampling strategy to form a pool
  - Optimal sampling ensures unbiased estimates of evaluation measures

Experimentation

- Suppose we have a test collection
  - A corpus, some queries, and relevance judgments
- We have a question we want to answer
  - e.g. which of *n* engines is best
- What do we do now?
  - Index the corpus with each engine
  - Run the queries on each engine
  - Evaluate the retrieved documents using some measure

Experimental Results

Query	Engine 1	Engine 2	Engine 3
Q1	0.192	0.525	0.778
Q2	0.269	0.595	0.874
Q3	0.187	0.663	0.880
Q4	0.243	0.512	0.808
Q5	0.131	0.564	0.799
Q6	0.208	0.511	0.769
Q7	0.094	0.569	0.787
Q8	0.222	0.518	0.853
Q9	0.104	0.537	0.881
Q10	0.177	0.450	0.781
average	**0.183**	**0.544**	**0.821**

Experimental Results

Query	Engine 1	Engine 2	Engine 3
Q1	0.415	0.311	0.350
Q2	0.286	0.392	0.434
Q3	0.326	0.519	0.568
Q4	0.357	0.436	0.245
Q5	0.409	0.470	0.509
Q6	0.407	0.258	0.449
Q7	0.374	0.391	0.292
Q8	0.341	0.429	0.420
Q9	0.327	0.500	0.517
Q10	0.387	0.464	0.505
average	**0.363**	**0.417**	**0.429**

Significance Tests

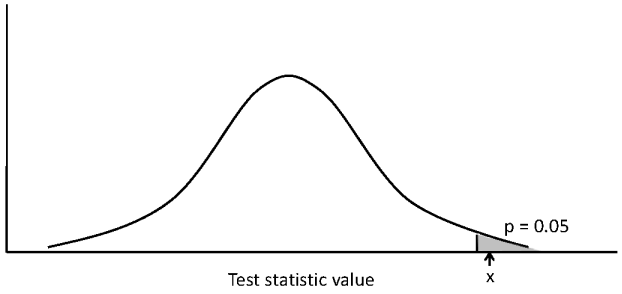
- Given the results from a number of queries, how can we conclude that ranking algorithm A is better than algorithm B?
- A significance test enables us to reject the *null hypothesis* (no difference) in favor of the *alternative hypothesis* (B is better than A)
  - the *power* of a test is the probability that the test will reject the null hypothesis correctly
  - increasing the number of queries in the experiment also increases power of test

Significance Tests

- Calculate effectiveness measure for each query for each engine
- Use those values to compute a *test statistic* that has a certain distribution when the null hypothesis is true (when there is no difference in system performance)
- Obtain a *p-value* from that distribution
- If the *p*-value is less than a given *critical value*, conclude that the null hypothesis is false
  - i.e., there *is* a difference between the systems

One-Sided Test

- Distribution for the possible values of a test statistic assuming the null hypothesis



- shaded area is *region of rejection*

Example Experimental Results

Query	Engine 2	Engine 3	E3 – E2
Q1	0.311	0.350	0.039
Q2	0.392	0.434	0.042
Q3	0.519	0.568	0.049
Q4	0.436	0.245	-0.191
Q5	0.470	0.509	0.039
Q6	0.258	0.449	0.191
Q7	0.391	0.292	-0.099
Q8	0.429	0.420	-0.009
Q9	0.500	0.517	0.017
Q10	0.464	0.505	0.041
average	**0.417**	**0.429**	**0.012**

t-Test

- Assumption is that the difference between the effectiveness values is a sample from a normal distribution
- Null hypothesis is that the mean of the distribution of differences is zero
- Test statistic

$$t = \frac{\overline{B-A}}{\sigma_{B-A}} \cdot \sqrt{N}$$

  - for the example, B-A = 0.012, $\sigma_{B\text{-}A}$ = 0.100, N = 10
  - t = 0.375, p-value = 0.358

Wilcoxon Signed-Ranks Test

- Nonparametric test based on differences between effectiveness scores
- Test statistic

$$w = \sum_{i=1}^{N} R_i$$

R_i is a signed-rank, N is the number of differences $\neq 0$

  - To compute the signed-ranks, the differences are ordered by their absolute values (increasing), and then assigned rank values
  - rank values are then given the sign of the original difference

Wilcoxon Example

- 9 non-zero differences are (in rank order of absolute value):

  0.009, 0.017, 0.039, 0.039, 0.041, 0.042, 0.049, 0.099, 0.191, 0.191

- Signed-ranks:

  -1, +2, +3.5, +3.5, +5, +6, +7, -8, +9.5, -9.5

- *w* = 18, p-value = 0.386

Sign Test

- Ignores magnitude of differences
- Null hypothesis for this test is that
  - P(B > A) = P(A > B) = ½
  - number of pairs where B is "better" than A would be the same as the number of pairs where A is "better" than B
- Test statistic is number of pairs where B>A
- For example data,
  - test statistic is 7, p-value = 0.17
  - cannot reject null hypothesis

Selecting Search Engine Parameters

- Retrieval models often contain parameters that must be tuned to get best performance for specific types of data and queries
  - Vector space: term weights w_{ik}
  - BM25: parameters k_1, k_3, b
  - LM: smoothing parameter α_D
- There may be additional features available
  - Link graph features, click information, …
- How can we decide which model to use, which features to incorporate, and what parameter values they take?

Training Search Engines

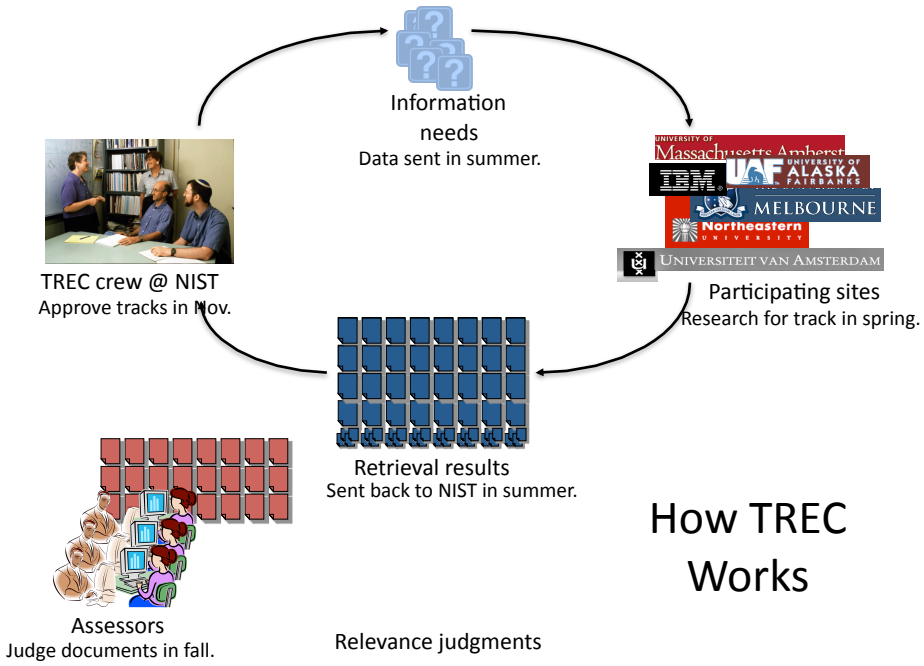
- By experimentation:
  - Use *training* and *test* data sets
  - If less data available, use *cross-validation* by partitioning the data into *K* subsets
  - Using training and test data avoids *overfitting* – when parameter values do not generalize well to other data

Finding Parameter Values

- Many techniques used to find optimal parameter values given training data
  - standard problem in machine learning
- In IR, often explore the space of possible parameter values by *brute force*
  - requires large number of retrieval runs with small variations in parameter values (*parameter sweep)*

TREC

- The Text REtrieval Conference
- A workshop/conference/competition for information retrieval experimentation
- Participating sites (companies & universities) are given a corpus and queries
- They index the corpus, run the queries, and return their results to TREC organizers
- TREC pools the top 100 results for assessment



TREC Test Collections

- TREC has the resources to get tens of thousands of relevance judgments each year
- Dozens of sites participate
- The judgments and queries are released to the research community for use in future experiments
- TREC has provided much of the data used in modern IR research

Test Collections

Collection	Number of documents	Size	Average number of words/doc.
CACM	3,204	2.2 Mb	64
AP	242,918	0.7 Gb	474
GOV2	25,205,179	426 Gb	1073

Collection	Number of queries	Average number of words/query	Average number of relevant docs/query
CACM	64	13.0	16
AP	100	4.3	220
GOV2	150	3.1	180

TREC Topic Example

<top>
<num> Number: 794

<title> pet therapy

<desc> Description:
How are pets or animals used in therapy for humans and what are the benefits?

<narr> Narrative:
Relevant documents must include details of how pet- or animal-assisted therapy is or has been used. Relevant details include information about pet therapy programs, descriptions of the circumstances in which pet therapy is used, the benefits of this type of therapy, the degree of success of this therapy, and any laws or regulations governing it.

</top>

TREC Tracks

- A slate of *tracks* run at TREC each year
- Each track is devoted to a different retrieval task
  - Ad hoc track: take an arbitrary query, provide ranked document results
  - Filtering track: documents come in as a stream, filter them against a "standing query"
  - Question answering: queries are natural-language questions, system must provide natural-language answers
  - Cross-language track: queries are in one language, documents are in a different language
  - Web track: queries designed to reflect uses of Web
  - Million query track: kind of like ad hoc, except designed to study evaluation itself

Million Query Judging Interface