

INFORMATION RETRIEVAL EVALUATION WORKSHOP

Sri Devi Ravana

Associate Professor

Department of Information Systems

Faculty of Computer Science and Information Technology



Where I come from?

- Born and raised in Johor, Malaysia
- Currently living in Kuala Lumpur, Malaysia



**State of Johore in
Peninsular Malaysia**

Background

- PhD in Computer Science, The University of Melbourne
- **Expertise:** Information Retrieval Systems specifically on Evaluation Issues
- **Associate Professor**, Dept. of Information Systems, Faculty of Computer Science & Information Technology, University of Malaya, Malaysia

Evaluation

- Evaluation is key to building *effective and efficient* search engines
 - measurement usually carried out in controlled *laboratory experiments*
 - *online* testing can also be done
- Effectiveness, efficiency and *cost* are related
 - e.g., if we want a particular level of effectiveness and efficiency, this will determine the cost of the system configuration
 - efficiency and cost targets may impact effectiveness

Evaluation Corpus

- *Test collections* consisting of documents, queries, and relevance judgments, e.g.,
 - CACM: Titles and abstracts from the Communications of the ACM from 1958-1979. Queries and relevance judgments generated by computer scientists.
 - AP: Associated Press newswire documents from 1988-1990 (from TREC disks 1-3). Queries are the title fields from TREC topics 51-150. Topics and relevance judgments generated by government information analysts.
 - GOV2: Web pages crawled from websites in the .gov domain during early 2004. Queries are the title fields from TREC topics 701-850. Topics and relevance judgments generated by government analysts.

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

Sample of document under .GOV http://ir.dcs.gla.ac.uk/test_collections/samples/GOV_sampleDoc

Sample under wt10g http://ir.dcs.gla.ac.uk/test_collections/samples/wt2g_sampleDoc

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

Introduction to TREC

- Website: <https://trec.nist.gov/>
- DATA: <https://pages.nist.gov/trec-browser/#adhoc>

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>

Relevance Judgments

- Obtaining relevance judgments is an **expensive, time-consuming process**
 - who does it?
 - what are the instructions?
 - what is the level of agreement?
- TREC judgments
 - depend on task being evaluated
 - generally binary
 - agreement good because of “narrative”

Pooling

- Exhaustive judgments for all documents in a collection is not practical
- Pooling technique is used in TREC
 - top k results (for TREC, k varied between 50 and 200) from the rankings obtained by different search engines (or retrieval algorithms) [runs] are merged into a pool
 - duplicates are removed
 - documents are presented in some random order to the relevance judges
- Produces a large number of relevance judgments for each query, although still incomplete

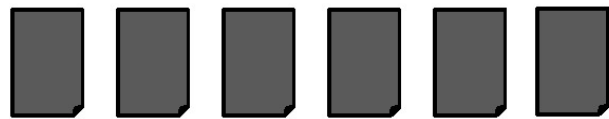
Effectiveness Measures

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

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

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

Ranking Effectiveness



= the relevant documents

1 over 6

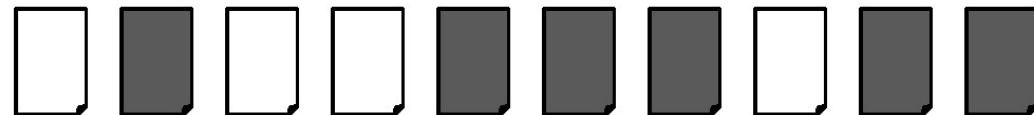
Ranking #1



Recall 0.17 0.17 0.33 0.5 0.67 0.83 0.83 0.83 0.83 1.0

Precision 1.0 0.5 0.67 0.75 0.8 0.83 0.71 0.63 0.56 0.6

Ranking #2



Recall 0.0 0.17 0.17 0.17 0.33 0.5 0.67 0.67 0.83 1.0

Precision 0.0 0.5 0.33 0.25 0.4 0.5 0.57 0.5 0.56 0.6











1 over 1











Summarizing a Ranking

- Calculating recall and precision at fixed rank positions
- Calculating 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

Average Precision

 = the relevant documents

Ranking #1										
Recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	0.83	1.0
Precision	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.6

Ranking #2										
Recall	0.0	0.17	0.17	0.17	0.33	0.5	0.67	0.67	0.83	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.5	0.57	0.5	0.56	0.6




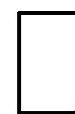
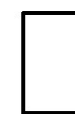

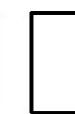



$$\text{Ranking \#1: } (1.0 + 0.67 + 0.75 + 0.8 + 0.83 + 0.6)/6 = 0.78$$


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

Averaging Across Queries

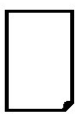


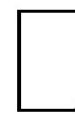

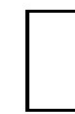


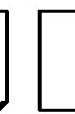
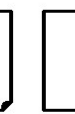
 = relevant documents for query 1

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

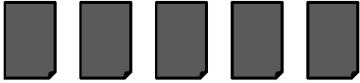
Ranking #2

										
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











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











 = 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

$$\text{average precision query 1} = (1.0 + 0.67 + 0.5 + 0.44 + 0.5)/5 = 0.62$$

$$\text{average precision query 2} = (0.5 + 0.4 + 0.43)/3 = 0.44$$

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

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, question answering
- 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 R (precision@R)
 - R typically 5, 10, 20
 - easy to compute, average, understand
 - not sensitive to rank positions less than R
- Reciprocal Rank
 - reciprocal of the rank at which the first relevant document is retrieved
 - *Mean Reciprocal Rank (MRR)* is the average of the reciprocal ranks over a set of queries
 - very sensitive to rank position

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

1. Compute the effectiveness measure for every query for both rankings.
2. Compute a *test statistic* based on a comparison of the effectiveness measures for each query. The test statistic depends on the significance test, and is simply a quantity calculated from the sample data that is used to decide whether or not the null hypothesis should be rejected.
3. The test statistic is used to compute a *P-value*, which is the probability that a test statistic value at least that extreme could be observed if the null hypothesis were true. Small P-values suggest that the null hypothesis may be false.
4. The null hypothesis (no difference) is rejected in favor of the alternate hypothesis (i.e., *B* is more effective than *A*) if the P-value is $\leq \alpha$, the *significance level*. Values for α are small, typically .05 and .1, to reduce the chance of a Type I error.

Example Experimental Results

Query	A	B	B-A
1	25	35	10
2	43	84	41
3	39	15	-24
4	75	75	0
5	43	68	25
6	15	85	70
7	20	80	60
8	52	50	-2
9	49	58	9
10	50	75	25

t-Test

- Assumption is that the difference between the effectiveness values is a sample **from a normal distribution [Parametrics tests]**
- 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,

$$\overline{B-A} = 21.4, \sigma_{B-A} = 29.1, t = 2.33, \text{p-value} = .02$$

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):
2, 9, 10, 24, 25, 25, 41, 60, 70
- Signed-ranks:
-1, +2, +3, -4, +5.5, +5.5, +7, +8, +9
- $w = 35$, p-value = 0.025

Sign Test

- Ignores magnitude of differences
- Null hypothesis for this test is that
 - $P(B > A) = P(A > B) = \frac{1}{2}$
 - 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

Setting Parameter Values

- Retrieval models often contain parameters that must be tuned to get best performance for specific types of data and queries
- For experiments:
 - 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

Correlation in IR System Ranking

- Measures the *degree of agreement* between two ranked lists (e.g., results from two IR systems or two sets of same systems using different algorithms).
- To assess **stability** of retrieval results across systems or parameter changes.
- To validate **evaluation metrics** e.g., whether Precision@10 correlates with MAP or nDCG.

Common Correlation Measures

- Spearman's Rank Correlation (ρ)
- Kendall's Tau (τ)
- Example Application Compare rankings from: Two retrieval models (e.g., BM25 vs. TF-IDF).

Summary

- No single measure is the correct one for any application
 - choose measures appropriate for task
 - use a combination
 - shows different aspects of the system effectiveness
- Use significance tests (t-test)
- Analyze performance of individual queries

THANK YOU

NOW LETS TRY IT OUT!!!

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 $1/\log(\text{rank})$
 - 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 at a particular rank p :

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

- Alternative formulation:

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

- 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
 - $\text{NDCG} \leq 1$ 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

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

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

- Alternative definition

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

- The bpref measure is designed for situations where relevance judgments are known to be far from complete
- Bpref can be thought of as the inverse of the fraction of judged irrelevant documents that are retrieved before relevant ones.
- Bpref and mean average precision are very highly correlated when used with complete judgments. But when judgments are incomplete, rankings of systems by bpref still correlate highly to the original ranking, whereas rankings of systems by MAP do not.