Evaluation
Why Evaluation?

• Does a new idea really improve the search?
  – Search engines are complicated
  – Unfortunately, many ideas may not work in practice
  – One’s feeling that design A is better than design B may often cheat us

• Evaluation: test the performance of a design by simulations and experiments
  – Many ideas often may have little or no impact when tested using quantitative experiments
Test Collection for IR Evaluation

- A document collection
- A test suite of information needs, expressible as queries
- A set of relevance judgments – gold standard / ground truth
  - A standard way is to have a binary assessment of either relevant or irrelevant for each query-document pair
  - At least 50 information needs may be considered sufficient
Relevance to Information Need

- Information need: “whether drinking red wine is more effective than drinking white wine at reducing risk of heart attacks?”
- Query: “wine, red, white, heart attack, effective”
- Judgment 1: do the documents in the answer contain all those keywords?
  - Even a document passing the judgment may not answer the information need question
- Judgment 2: do the documents in the answer discuss my information need question?
  - This judgment should be used in the evaluation
Subjective and Objective Judgments

- Query: “java”
  - Information need 1: programming language Java
  - Information need 2: java coffee
  - Information need 3: island Java
- Bearing different information needs in mind will have different judgments on the relevance of a set of answers
- In evaluation, we should use the queries that can express the information need clearly
  - Ambiguous queries should be avoided for general information retrieval evaluation
Effectiveness and Efficiency

• The two major issues in search engines
  - Effectiveness: how well the ranking generated by a search engine corresponds to the ranking based on user relevance judgments
  - Efficiency: the time and space requirements for the algorithm to generate the ranking
• A few other factors may affect effectiveness and efficiency such as interface, query suggestion, and relevance feedback
  - It is hard to evaluate many factors at a time
• Evaluation is typically done in tightly defined experimental settings
Effectiveness versus Efficiency

• Techniques giving a small boost to effectiveness may not be included in a search engine implementation if they have a significant adverse effect on an efficiency measure such as query throughput.

• A search engine that is extremely fast is of no use unless it produces good results.

• A two-step approach
  – Improving the effectiveness of search
  – When a technique has been established as being potentially useful, try to find efficient implementations.

• So far, there is no reliable technique that significantly improves effectiveness that cannot be incorporated into a search engine due to efficiency considerations.
Cost

• Efficient implementation of a specific technique may require a huge investment in processors, memory, disk, and networking

• Effectiveness, efficiency and cost
  – Picking targets for any two, the third can be determined
  – Extreme cases
    • Searching using a pattern matching utility such as grep – poor effectiveness, poor efficiency, low cost
    • Searching using an organization like the Library of Congress – effective, very expensive
Using Test Collections

• In a search engine, there are often many parameters that can be adjusted to tune system performance
• Tuning the parameters to optimize for one test data set?
  – Over-fitting – if the search engine fits a specific workload too well, it may not work well if the workload changes
• Using multiple development test collections
  – Unbiased estimation of performance
  – Tuning for multiple test collections is more robust
  – Repeatability of experimental results
Cranfield

- The pioneering test collection
- Collected in UK in late 1950s
  - Cranfield is named after the place in UK where the experiments were done
- 1,398 abstracts of aerodynamics journal articles
- 225 queries and exhaustive relevance judgments of all (query, document) pairs
- Too small nowadays
CACM

- Bibliographic records
  - Titles and abstracts from *the Communication of ACM* from 1958-1979
  - 3,204 documents, 2.2 MB, on average 64 words/document
- Queries and relevance judgments generated by computer scientists
  - 64 queries, on average 13 words/query and 16 relevant documents/query
  - Example of query: “Security considerations in local networks, network operating systems, and distributed systems”
  - Relevance judgments by the same people asking the queries
AP and GOV2

• Created as part of the TREC (for text retrieval conference) series sponsored by the National Institute of Standards and Technology (NIST)
  • AP: Associated Press newswire documents from 1988-1990 (TREC disks 1-3)
    – 242,918 documents, 0.7 GB, on average 474 words/document
    – 100 queries, on average 4.3 words/query and 220 relevant documents/query
  • Gov2: Web pages crawled from web sites in the .gov domain during early 2004
    – 25,205,179 documents, 426 GB, on average 1073 words/document
    – 150 queries, on average 3.1 words/query and 180 relevant documents/query
• Topics and relevance judgments generated by government information analysts
TREC Topic Example

A short query
<title>pet therapy</title>
<desc>Description:
How are pets or animals used in therapy for humans and what are the benefits?
</desc>

A longer version of the query
<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.
</narr>

The criteria for relevance

Limitations in TREC

• Emphasizing recall in relevance judgment
  – TREC analysts judged a document as relevant if it contains the information that could be used to help write a report on the query topic

• Binary relevance judgments – either relevant or irrelevant

• Classroom discussion: can you give an example task that TREC relevance judgments are not helpful?
NTCIR and CLEF

• **NTCIR: NII Test Collections for IR Systems**
  – Similar size to the TREC collections
  – Focusing on East Asian language and cross-language information retrieval – queries are made in one language over a document collection containing documents in one or more other languages

• **CLEF: Cross Language Evaluation Forum**
  – Concentrated on European languages and cross-language information retrieval
Reuters and Newsgroups

- Reuters-21578: 21,578 newswire articles
- Reuters-RCV1: Reuters Corpus Volume 1 consisting of 806,791 documents
- Reuters data sets are most used for text classification
- 20 newsgroups by Ken Lang
  - 1,000 articles from each of 20 Usenet newsgroups
  - After removing duplicates, 18,941 articles in the whole data set
Relevance Judgments

- It is impossible to evaluate relevance on every document in a large collection
- Pooling
  - Use a set of search engines/algorithms
  - The top-k results (k is between 20 to 50 in TREC) are merged into a pool, duplicates are removed
  - Present the documents in a random order to analysts for relevance judgments
- Empirically, the relevance judgments are complete enough to produce accurate comparisons for new search techniques
- Classroom discussion: in what situation the pooling technique does not work?
Kappa Statistics

• If we have multiple judges on one information need, how consistent are those judges?

• Kappa statistics is for categorical judgments and corrects a simple agreement rate for the rate of chance agreement
  – $\kappa = (P(A) - P(E)) / (1 - P(E))$
  – $P(A)$ is the proportion of the times that the judges agreed
  – $P(E)$ is the proportion of the times they would be expected to agree by chance

• Using marginal statistics to estimate $P(E)$
Example

<table>
<thead>
<tr>
<th></th>
<th>judge 1 relevance</th>
<th>judge 2 relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>judge 1</td>
<td>300</td>
<td>20</td>
</tr>
<tr>
<td>relevance</td>
<td>no</td>
<td>10</td>
</tr>
<tr>
<td>total</td>
<td>310</td>
<td>90</td>
</tr>
</tbody>
</table>

Observed proportion of the times the judges agreed

\[ P(A) = \frac{300 + 70}{400} = \frac{370}{400} = 0.925 \]

Pooled marginals

\[ P(\text{nonrelevant}) = \frac{80 + 90}{400 + 400} = \frac{170}{800} = 0.2125 \]
\[ P(\text{relevant}) = \frac{320 + 310}{400 + 400} = \frac{630}{800} = 0.7878 \]

Probability that the two judges agreed by chance

\[ P(E) = P(\text{nonrelevant})^2 + P(\text{relevant})^2 = 0.2125^2 + 0.7878^2 = 0.665 \]

Kappa statistic

\[ \kappa = \frac{P(A) - P(E)}{1 - P(E)} = \frac{0.925 - 0.665}{1 - 0.665} = 0.776 \]
Multiple Judges

- Kappa value is 1 if two judges always agree, 0 if they agree only at the rate given by chance, negative if they are worse than random (negatively correlated)
- If there are more than 2 judges, calculate an average pair-wise kappa value
- A rule of thumb
  - A kappa value between 0.67 and 0.8 is taken as fair agreement
  - Agreement below 0.67 is seen as data providing dubious basis for an evaluation
Developing Evaluation Corpus

- Use a document database that is representative for the target application in terms of the number, size and type of documents
- Use a set of representative queries submitted by users of the target applications
  - Using query log
  - Collect query examples from potential users
- Relevance judgments by either the people who asked the questions, or by independent judges who have been instructed in how to determine relevance for the application being evaluated
  - Three levels: definitely relevant, definitely irrelevant, possibly relevant
Using Queries Logs in Evaluation

• Using implicit relevance feedback may help to substantially reduce the cost of creating an evaluation corpus
• Query log data can potentially support a much more extensive and realistic evaluation
• Major drawback: query log data is not as precise as explicit relevance judgments
• Privacy concern – anonymization by removing identifying information or queries that may contain personal data
Query Log Data

- User identifier or user session identifier
  - User login id in services, search toolbar id, cookies
  - A session is a series of queries submitted to a search engine over a limited amount of time (e.g., 30 minutes)
- Query terms – exactly as the user entered
- Clickthrough data: a list of URLs of results, their ranks on the result list, and whether they were clicked on
- Timestamps
Other Implicit Relevance Judgments

• Dwell time: the amount of time the user spends on a clicked result
  – The elapsed time from the initial click to the time when the user comes back to the results page or exists the search

• Search exist action: the way the user exists the search application
  – Entering another URL, closing the browser window, timing out, printing a page
Bias in Clickthrough Data

• Very biased toward pages that are highly ranked, popular, or having a good snippet

• Using preferences to remove bias
  – Rather than relevance judgments, predict user preferences between pairs of documents

• Skip above and skip next
  – A query returns ranked list “d1, d2, d3, d4” and d3 is clicked
  – Skip above: d3 > d1, d3 > d2
  – Skip next: d3 > d4
Aggregating Clickthrough Data

- Using click distribution
- For a given query q, use all instances of q in the log to compute the observed click frequency $O(d, p)$ for result d in rank position p
- Expected click frequency $E(p)$ for all results at rank p
- Click deviation $CD(d, p) = O(d, p) - E(p)$
  - The larger, the more likely d is relevant to the query
Precision and Recall

- Precision = \#relevant-documents-retrieved / \#retrieved-documents = P(relevant | retrieved)
  - \( P = \frac{tp}{tp + fp} \)
- Recall = \#relevant-documents-retrieved / \#relevant-documents = P(retrieved | relevant)
  - \( R = \frac{tp}{tp + fn} \)

<table>
<thead>
<tr>
<th></th>
<th>relevant</th>
<th>nonrelevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>retrieved</td>
<td>true positives (tp)</td>
<td>false positives (fp)</td>
</tr>
<tr>
<td>not retrieved</td>
<td>false negatives (fn)</td>
<td>true negatives (tn)</td>
</tr>
</tbody>
</table>
Accuracy

• Accuracy = (tp + tn) / (tp + fp + fn + tn)
• Not a good measure for information retrieval
  – 99.9% of documents are irrelevant in most of the cases
  – Label every document as irrelevant has high accuracy but useless in information retrieval

<table>
<thead>
<tr>
<th></th>
<th>relevant</th>
<th>nonrelevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>retrieved</td>
<td>true positives (tp)</td>
<td>false positives (fp)</td>
</tr>
<tr>
<td>not retrieved</td>
<td>false negatives (fn)</td>
<td>true negatives (tn)</td>
</tr>
</tbody>
</table>
Trading off Between P and R

- F measure – the weighted harmonic mean of precision and recall

\[
F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \quad \text{where} \quad \beta^2 = \frac{1-\alpha}{\alpha}
\]

- \( \alpha \in [0, 1] \), \( \beta \in [0, \infty] \)
- Why harmonic mean?
  - The harmonic mean is always less than or equal to the arithmetic mean and the geometric mean
  - When P and R differ greatly, the harmonic mean is closer to their minimum than to their arithmetic mean

- When \( \alpha = \frac{1}{2} \) (i.e., \( \beta = 1 \)), \( F_1 = \frac{2PR}{(P + R)} \)
Fallout

• Type I errors – false positive: an irrelevant document is retrieved
  – Precision is not related directly to this error!
  – Fallout: the type I error rate, \( \frac{fp}{tp + fp} \)
  – Fallout is often very small if the search engine returns a small number of documents from a huge collection

• Type II errors – false negative: a relevant document is not retrieved
  – Captured by recall
Evaluation of Ranked Lists

- Recall and precision values at every rank position

| Ranking #1 | Recall | 0.17 | 0.17 | 0.33 | 0.5 | 0.67 | 0.83 | 0.83 | 0.83 | 0.83 | 1.0 |
| Ranking #2 | Recall | 0.0  | 0.17 | 0.17 | 0.17 | 0.33 | 0.5  | 0.67 | 0.67 | 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 #1 | Precision | 1.0 | 0.5  | 0.67 | 0.75 | 0.8  | 0.83 | 0.71 | 0.63 | 0.56 | 0.6  |
| Ranking #2 | Precision | 0.0 | 0.5  | 0.33 | 0.25 | 0.4  | 0.5  | 0.57 | 0.5  | 0.56 | 0.6  |
Precision at Rank $p$

- If there are many relevant documents for a query, or if the relevant documents are widely distributed in the ranking, a list of recall-precision values for every rank position is long and unwieldy.
- Only compute the precision at the predefined rank positions.
  - If the precision for a ranking at rank position is higher than the precision for another rank, the recall must be also higher (classroom discussion: why?)
  - Precision at 10 and 20 are often used.
  - The IR task is changed – find the most relevant documents at a given rank instead of finding as many relevant documents as possible.
Precision-Recall Curve

• Saw tooth shape
  – If the (k+1)-th document retrieved is irrelevant, the recall is the same as for the top-k documents, but the precision has dropped
  – If it is relevant, then both the precision and the recall increase

• Removing jiggles using an interpolated precision
  – $p_{\text{interp}}(r) = \max_{r' \geq r} p(r')$
  – Almost anyone would be prepared to look at a few more documents if it would increase the precision
Precision-Recall Curve Example
11-Point Interpolated Avg Precision

- The interpolated precision is measured at the 11 recall levels of 0.0, 0.1, 0.2, …, 1.0
- Macroaverage on multiple queries: compute the measure of interest for each query and then average the measures

<table>
<thead>
<tr>
<th>recall</th>
<th>interp. precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>1.00</td>
</tr>
<tr>
<td>0.1</td>
<td>0.67</td>
</tr>
<tr>
<td>0.2</td>
<td>0.63</td>
</tr>
<tr>
<td>0.3</td>
<td>0.55</td>
</tr>
<tr>
<td>0.4</td>
<td>0.45</td>
</tr>
<tr>
<td>0.5</td>
<td>0.41</td>
</tr>
<tr>
<td>0.6</td>
<td>0.36</td>
</tr>
<tr>
<td>0.7</td>
<td>0.29</td>
</tr>
<tr>
<td>0.8</td>
<td>0.13</td>
</tr>
<tr>
<td>0.9</td>
<td>0.10</td>
</tr>
<tr>
<td>1.0</td>
<td>0.08</td>
</tr>
</tbody>
</table>
Average Precision

- The average of the precision values from the rank positions where a relevant document was retrieved, i.e., when recall increases:
  - Ranking #1: \((1.0 + 0.67 + 0.75 + 0.8 + 0.83 + 0.6) / 6 = 0.78\)
  - Ranking #2: \((0.5 + 0.4 + 0.5 + 0.57 + 0.56 + 0.6) / 6 = 0.52\)
Why Is Average Precision Popular?

- A single value measure based on the ranking of all the relevant documents
  - Finding as many relevant documents as possible
- The value depends heavily on the highly ranked relevant documents
  - The top ranked documents are the most important
- Remaining issue: it cannot be used to measure multiple queries
Mean Average Precision

- The average of the average precision on each query
- Assumption: the user is interested in finding many relevant documents for each query
  - Requiring a considerable effort to acquire the relevance judgments

Average precision Q1 = \( \frac{1.0 + 0.67 + 0.5 + 0.44 + 0.5}{5} = 0.62 \)

Average precision Q2 = \( \frac{0.5 + 0.4 + 0.43}{3} = 0.44 \)

Mean average precision = \( \frac{0.62 + 0.44}{2} = 0.53 \)
Geometric Mean of Avg Precision

• Using the geometric mean

\[ GMAP = \sqrt[n]{\prod_{i=1}^{n} AP_i} \quad \text{or} \quad GMAP = e^{\frac{1}{n} \sum_{i=1}^{n} \log AP_i} \]

• Emphasizing the impact of queries with low performance
Reciprocal Rank

- Top ranked documents may be considered more important
- Reciprocal rank: the rank at which the first relevant document is retrieved
  - If the first 4 documents for query q is “irrelevant, relevant, irrelevant, irrelevant”, the reciprocal rank is $\frac{1}{2} = 0.5$
  - Very sensitive to the rank positions
- The mean reciprocal rank: the average of the reciprocal ranks over a set of queries
R-precision

- Assumption: we have a set of known relevant documents Rel
  - Rel may not be complete and can be obtained by the pool method
- Calculate the precision of the top Rel documents returned
- Break-even point: the point where the precision and the recall on Rel are identical
ROC Analysis

- **ROC**: receiver operating characteristics
  - Plotting the true positive rate (sensitivity) against the false-positive
  - Sensitivity is another name of recall
  - False-positive rate = $\frac{fp}{fp + tn}$
  - Specificity = $\frac{tn}{fp + tn}$, almost 1 for most cases, not very useful in IR

ROC curve graph
Discounted Cumulative Gain (DCG)

- Highly relevant documents are more useful than marginally relevant ones
- The lower the ranked position of a relevant document, the less useful it is for the users
- Using graded relevance as a measure
  - Accumulating gain starting at the top of the ranking, reduced or discounted at lower rank
    \[
    DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}
    \]
    \[
    DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2 (i+1)}
    \]
    - \(rel_i\) is the graded relevance level of the document retrieved at rank \(i\), e.g., in binary relevance judgments, \(rel_i\) is either 0 or 1; in a 6 point scale ranging from “bad” to “perfect”, \(rel_i\) is from 0 to 5
    - The 2\textsuperscript{nd} form has been used by some search engine companies and strongly emphasizes on strongly relevant documents
DCG Example

• Suppose a 4 point scale is used in relevance judgments

• A ranking of answers to a query: 3, 2, 3, 0, 0, 1, 2, 2, 3, 0

• Discounted gains are: 3, 2/1=2, 3/1.59=1.89, 0, 0, 1/2.59=0.39, 2/2.81=0.71, 2/3=0.67, 3/3.17=0.95, 0

• DCG at each rank is to accumulate these numbers: 3, 5, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61

• Use DCG at rank $p$, $p$ is often set to 5 and 10
Normalized DCG (NDCG)

- Normalizing DCG by the perfect ranking
- For a ranking of answers to a query: 3, 2, 3, 0, 0, 1, 2, 2, 3, 0, the perfect ranking is 3, 3, 3, 2, 2, 2, 1, 0, 0, 0
  - The ideal DCG values: 3, 6, 7.89, 8.89, 9.75, 10.52, 10.88, 10.88, 10.88, 10.88
- The normalized DCG values: 1, 0.83, 0.87, 0.76, 0.71, 0.69, 0.73, 0.8, 0.88, 0.88
Kendall Tau Coefficient

• How can we compare a ranking and a user preference?
• Let $P$ be the number of preferences that agree and $Q$ be the number that disagree
  \[ \tau = \frac{P - Q}{P + Q} \]
  – 1 when all preference agree, -1 when all disagree

• Learning from partial preferences
  – If there were 15 preferences learned from clickthrough data, and a ranking agreed with 10, $\tau = (10 - 5) / 15 = 0.33$
  – The effectiveness of this measure is unclear yet
BPREF

• The number of relevant and irrelevant documents are balanced to facilitate averaging across queries

• For a query with R relevant documents, only the first R irrelevant documents are considered
  – Equivalent to use R x R preferences, i.e., all relevant documents are preferred to all irrelevant ones
    
    \[ BPREF = \frac{1}{R} \sum_{d_r} (1 - \frac{N_{d_r}}{R}) \]  or equivalently \[ \frac{P}{P + Q} \]

  – \( d_r \) is a relevant document, \( N_{d_r} \) is the number of irrelevant documents that are ranked higher than \( d_r \)
  – \( N_{d_r} \) can be regarded as the number of preferences that disagree for binary relevance judgments
Marginal Relevance

- So far, whether a document is relevant is judged independent from other documents
  - In many search engines, documents are presented to users in a ranked list
- Marginal relevance: whether a document still has distinctive usefulness after the user has looked at certain other documents
  - Even if a document is highly relevant, its information can be completely redundant with other documents that have already been examined (e.g., duplicate documents)
- Maximizing marginal relevance leads to optimizing diversity in answers
Query Throughput and Latency

- **Query throughput**: the number of queries processed per second
  - Hardware, document set, and query set dependent
  - Cannot capture latency

- **Query latency**: the elapsed time the system takes between the user issues a query and the system delivers the answers
  - Often measured by the median or a percentile bound
Other Efficiency Metrics

- Elapsed indexing time: the amount of time necessary to build a document index on a particular system
- Indexing processor time: the CPU seconds used in building a document index
- Indexing temporary space: the amount of disk space used while creating an index
- Index size: the amount of storage necessary to store the index files
Significance Tests

- Are two algorithms different in effectiveness?
  - The null hypothesis: there is NO difference
  - The alternative hypothesis: there is a difference – B is better than A (the baseline method)
- Matched pair experiments: the rankings that are compared are based on the same set of queries for both algorithms
- Possible errors of significant tests
  - Type I: the null hypothesis is rejected when it is true
  - Type II: the null hypothesis is accepted when it is false
- The power of a hypothesis test: the probability that the test will reject the null hypothesis correctly
  - Reducing the type II errors
Procedure of Comparison

- Using a particular set of queries
- Procedure
  - Compute the effectiveness measure for every query
  - Compute a test statistic based on a comparison of the effectiveness measures for each query
    - E.g., the t-test, the Wilcoxon signed-rank test, and the sign test
  - Compute a P-value: the probability that a test statistic value at least that extreme could be observed if the null hypothesis were true
  - The null hypothesis is rejected if the P-value \( \leq \alpha \), where \( \alpha \) is the significance level which is used to minimize the type I errors
- One-sided (one-tailed) tests: whether B is better than A (the baseline method)
  - Two-sided tests: whether A and B are different – the P-value is doubled
Distribution of Test Statistics

Test statistic value

$p = 0.05$

$x$
T-test

- Assuming data values are sampled from normal distributions
  - In a matched pair experiment, assuming the difference between the effectiveness values is a sample from a normal distribution

- The null hypothesis: the mean of the distribution of difference is 0

\[ t = \frac{\bar{B} - \bar{A}}{\sigma_{B-A} / \sqrt{N}} \]

- \( \bar{B} - \bar{A} \) is the mean of the differences, \( \sigma_{B-A} \) is the standard deviation of the differences

\[ \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i^2 - \bar{x}^2)} \]
## Example

<table>
<thead>
<tr>
<th>Query</th>
<th>A</th>
<th>B</th>
<th>B - A</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25</td>
<td>35</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>43</td>
<td>84</td>
<td>41</td>
</tr>
<tr>
<td>3</td>
<td>39</td>
<td>15</td>
<td>-24</td>
</tr>
<tr>
<td>4</td>
<td>75</td>
<td>75</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>43</td>
<td>68</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>85</td>
<td>70</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>80</td>
<td>60</td>
</tr>
<tr>
<td>8</td>
<td>52</td>
<td>50</td>
<td>-2</td>
</tr>
<tr>
<td>9</td>
<td>49</td>
<td>58</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>50</td>
<td>75</td>
<td>25</td>
</tr>
</tbody>
</table>

\[ B - A = 21.4 \]

\[ \sigma_{B-A} = 29.1 \]

\[ t = 2.33 \]

P-value = 0.02

significant at a level of \( \sigma = 0.05 \) – the null hypothesis can be rejected
Issues in T-test

• Data is assumed to be sampled from normal distributions
  – Generally inappropriate for effectiveness measures
  – However, experiments showed that t-test produces very similar results to the randomization test which does not assume any distribution (the most powerful nonparametric test)

• T-test assumes that the evaluation data is measured on an interval scale
  – Effectiveness measures are ordinal – the magnitude of the differences are not significant
  – Use the Wilcoxon signed-rank test and the sign test, which make less assumption about the effectiveness measure, but are less powerful
Wilcoxon Signed-Rank Test

- Assumption: the differences between the effectiveness values can be ranked, but the magnitude is not important

\[ w = \sum_{i=1}^{N} R_i \]

- \( R_i \) is a signed-rank, \( N \) is the number of non-zero differences

- Procedure
  - The differences are sorted by their absolute values increasing order
  - Differences are assigned rank values (ties are assigned the average rank)
  - The rank values are given the sign of the original difference

- The null hypothesis: the sum of the positive ranks will be the same as the sum of the negative ranks
Example

<table>
<thead>
<tr>
<th>Query</th>
<th>A</th>
<th>B</th>
<th>B-A</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25</td>
<td>35</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>43</td>
<td>84</td>
<td>41</td>
</tr>
<tr>
<td>3</td>
<td>39</td>
<td>15</td>
<td>-24</td>
</tr>
<tr>
<td>4</td>
<td>75</td>
<td>75</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>43</td>
<td>68</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>85</td>
<td>70</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>80</td>
<td>60</td>
</tr>
<tr>
<td>8</td>
<td>52</td>
<td>50</td>
<td>-2</td>
</tr>
<tr>
<td>9</td>
<td>49</td>
<td>58</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>50</td>
<td>75</td>
<td>25</td>
</tr>
</tbody>
</table>

The non-zero differences in rank order of absolute value:
2, 9, 10, 24, 25, 25, 41, 60, 70

The signed ranks: -1, +2, +3, -4, +5.5, +5.5, +7, +8, +9

\[ w = 35 \]

P-value = 0.025

significant at a level of \( \sigma = 0.05 \) – the null hypothesis can be rejected
Sign Test

- Completely ignore the magnitude of the differences
  - In practice, we may require that a 5-10\% difference is needed to be considered as different
- The null hypothesis: \( P(B > A) = P(A > B) = \frac{1}{2} \)
- Sum up the number of pairs \( B > A \)
Example

<table>
<thead>
<tr>
<th>Query</th>
<th>A</th>
<th>B</th>
<th>B-A</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25</td>
<td>35</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>43</td>
<td>84</td>
<td>41</td>
</tr>
<tr>
<td>3</td>
<td>39</td>
<td>15</td>
<td>-24</td>
</tr>
<tr>
<td>4</td>
<td>75</td>
<td>75</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>43</td>
<td>68</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>85</td>
<td>70</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>80</td>
<td>60</td>
</tr>
<tr>
<td>8</td>
<td>52</td>
<td>50</td>
<td>-2</td>
</tr>
<tr>
<td>9</td>
<td>49</td>
<td>58</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>50</td>
<td>75</td>
<td>25</td>
</tr>
</tbody>
</table>

7 pairs out of 10 B > A
P-value = 0.17 – the probability that we observe 7 successes out of 10 trials where the probability of success is 0.5
Cannot reject the null hypothesis
A/B Testing

• Precisely one thing is changed between the current system and a proposed system
• A small proportion of traffic (e.g., 1-10%) is randomly directed to the proposed system
Which Measure Should We Used?

• In many settings, all of the following measures and tests can be carried out with little additional effort
  – Mean average precision – single number summary, popular measure, pooled relevance judgments
  – Average NDCG – single number summary for each rank level emphasizing top ranked documents, relevance judgments only need to a specific rank depth (e.g., 10)
  – Recall-precision graph – conveying more information than a single number measure, pooled relevance judgments
  – Average precision at rank 10 – emphasizing top ranked documents, easy to understand, relevance judgments limited to top-10
Summary

• Information retrieval system evaluation
• Standard test collections
• Logging
• Effectiveness metrics
  – Unranked retrieval sets versus ranked retrieval results
  – Single query versus multiple queries
  – Using multiple judges
• Efficiency metrics
• Training, testing and statistics
• A/B testing