DAT630
# Retrieval Evaluation

**Search Engines, Chapter 8**

12/10/2016

**Krisztian Balog** | University of Stavanger

# Today

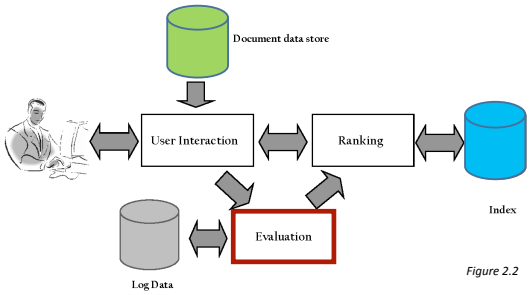



*Figure 2.2*

# 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

- To ensure repeatable experiments and fair comparison of results from different systems
- *Test collections* consist of
  - Documents
  - Queries
  - Relevance judgments
  - (Evaluation metrics)

# Text REtrieval Conference (TREC)

- Organized by the US National Institute of Standards and Technology (NIST)
- Yearly benchmarking cycle
- Development of test collections for various information retrieval tasks
- Relevance judgments created by retired CIA information analysts

# TREC Assessors at Work

# Example Test Collections

- 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.

# Example 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 |

# ClueWeb09/12 collections

- ClueWeb09
  - 1 billion web pages in 10 languages
  - 5TB compressed, 25TB uncompressed
  - http://lemurproject.org/clueweb09/
- ClueWeb12
  - 733 million English web pages
  - http://lemurproject.org/clueweb12/

# 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 Topic Example

<top>
<num> Number: 794

<title> pet therapy Short query (like in web search)

<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 Topic Example

<top>
<num> Number: 794

<title> pet therapy Longer (more precise) version of the query

<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 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?

Description of the criteria for relevance

<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>

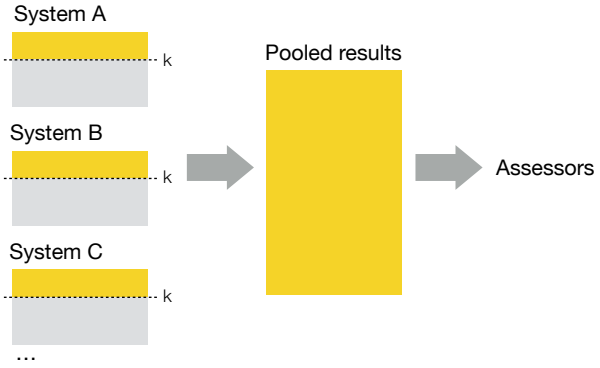
# Relevance Judgments

- 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 is 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) 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

# Pooling



# Crowdsourcing

- Obtain relevance judgments on a crowdsourcing platform
  - "Microtasks", performed in parallel by large, paid crowds
- Platforms
  - Amazon Mechanical Turk (US)
  - Crowdflower (EU)
    - https://www.crowdflower.com/use-case/search-relevance/

# 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
  - Given a set of results for a query and a clicked result at rank position *p*
    - all unclicked results ranked above p are predicted to be less relevant than the result at p
    - unclicked results immediately following a clicked result are less relevant than the clicked result

| click data | generated preferences |
|---|---|
| $d_1$ | $d_3 > d_2$ |
| $d_2$ | $d_3 > d_1$ |
| $d_3$ (clicked) | $d_3 > d_4$ |
| $d_4$ | |

# 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

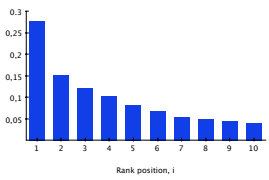
# Filtering Clicks

- 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

# 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

# Effectiveness Measures

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

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

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

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

# 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

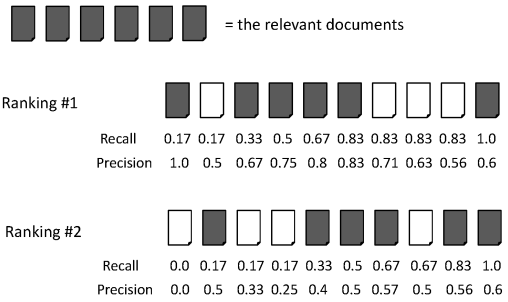
$$F_\beta = (\beta^2+1)RP/(R+\beta^2 P)$$

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

# Evaluating Rankings

- Precision and Recall are set-based metrics
- How to evaluate a ranked list?
  - Calculate recall and precision values at every rank position

# Ranking Effectiveness




# Evaluating Rankings

- Precision and Recall are set-based metrics
- How to evaluate a ranked list?
  - Calculate recall and precision values at every rank position
    - **Produces a long list of numbers** (see previous slide)
  - Need to summarize the effectiveness of a ranking

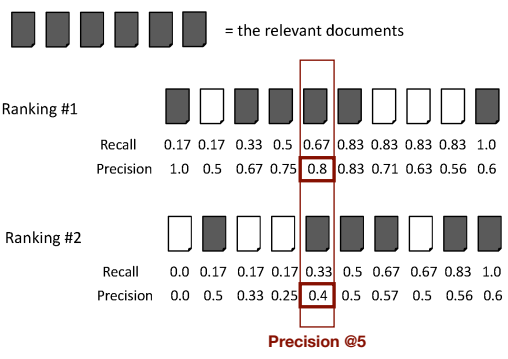
# Summarizing a Ranking

- 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

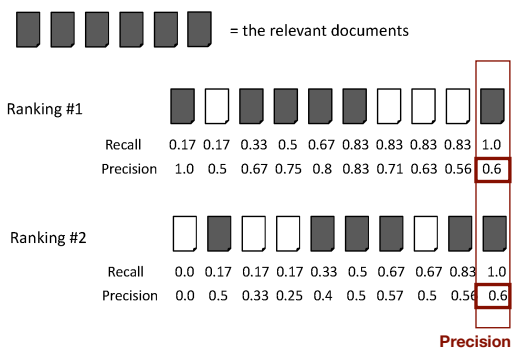
# Fixed Rank Positions

- Compute precision/recall at a given rank position *p*
  - E.g., precision at 20 (P@20)
  - Typically precision at 10 or 20
- This measure does not distinguish between differences in the rankings at positions *1* to *p*

# Example

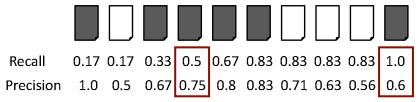



# Example



# Standard Recall Levels

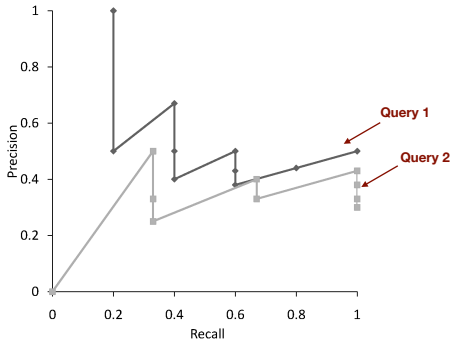
- Calculating precision at standard recall levels, from 0.0 to 1.0
  - Each ranking is then represented using 11 numbers
  - Values of precision at these standard recall levels are often not available, for example:



| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

  - *Interpolation* is needed

# Recall-Precision Graph



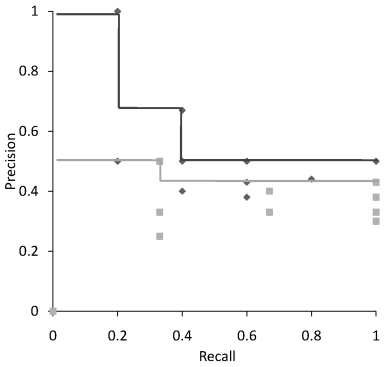

# Interpolation

- To average graphs, calculate precision at standard recall levels:

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

  - 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

# Interpolation




# Average Precision

- Average the precision values from the rank positions where a relevant document was retrieved
  - If a relevant document is not retrieved (in the top K ranks, e.g, K=1000) then its contribution is 0.0
  - Single number that is based on the ranking of all the relevant documents
  - The value depends heavily on the highly ranked relevant documents

# Average Precision

$$AP = \frac{1}{|Rel|} \sum_{\substack{i=1,\ldots,n \\ d_i \in Rel}} P(i)$$

$P(i)$ → **Precision at rank i**

**Total number of relevant documents**
According to the ground truth

**Only relevant documents contribute to the sum**

# 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 |

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

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

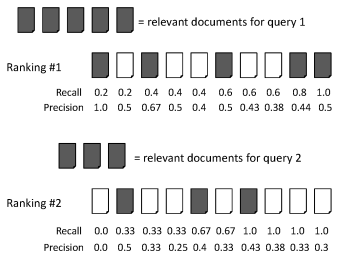
# Averaging Across Queries

- So far: measuring ranking effectiveness on a **single query**
- Need: measure ranking effectiveness on a **set of queries**
- Average is computed over the set of queries

# Mean Average Precision (MAP)

- Summarize rankings from multiple queries by averaging average precision
- Very succinct summary
- Most commonly used measure in research papers
- Assumes user is interested in finding many relevant documents for each query
- Requires many relevance judgments

# MAP




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

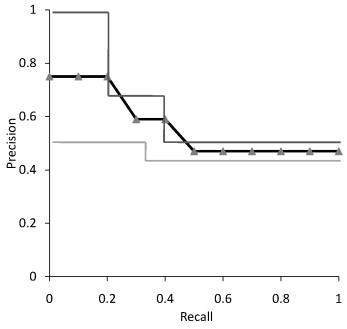
$$mean\ average\ precision = (0.62 + 0.44)/2 = 0.53$$

# Recall-Precision Graph

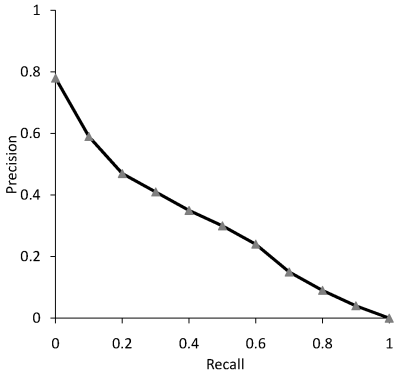
- Give more detail on the effectiveness of the ranking algorithm at different recall levels
- Graphs for individual queries have very different shapes and are difficult to compare
- Averaging precision values for standard recall levels over all queries

| 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 |

# Average Recall-Precision Graph




# Graph for 50 Queries




# Other Metrics

- Focusing on the top documents
- Using graded relevance judgments
  - E.g., web search engine companies often use a 6-point scale: bad (0) … perfect (5)

# 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 is 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
  - 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

# Mean Reciprocal Rank


= the relevant documents

Ranking #1

*Reciprocal rank (RR) = 1/1 = 1.0*

Ranking #2

*Reciprocal rank (RR) = 1/2 = 0.5*

*Mean reciprocal rank (MRR) = (1.0 + 0.5) /2 = 0.75*

# Exercise

# 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 at a particular rank *p*:

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

  - $rel_i$ is the graded relevance level of the document retrieved at rank *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
  - Typically at rank 5 or 10
  - 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
- 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

# Exercise

# Significance Testing

- 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

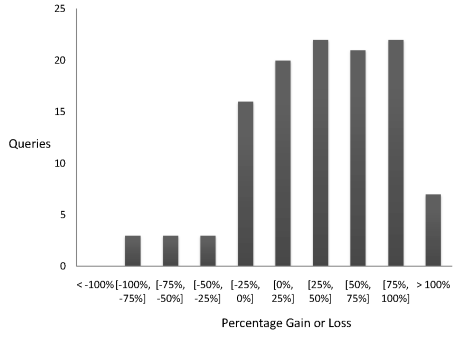
# Recipe

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 $\alpha$ are small, typically .05 and .1, to reduce the chance of a Type I error.

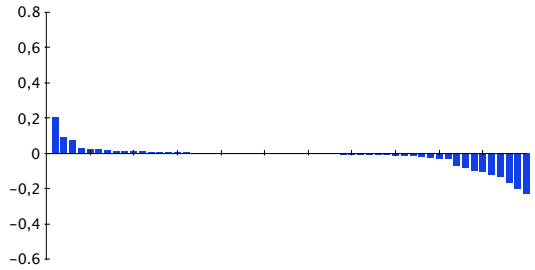
# Performance Analysis

- Typically, system A (baseline) is compared against system B (improved version)
- Average numbers can hide important details about the performance of individual queries
- Important to analyze which queries were helped and which were hurt

# Distribution of query effectiveness improvements




# Query-level performance differences




# Efficiency Metrics

- Elapsed indexing time
  - Amount of time necessary to build a document index on a particular system
- Indexing processor time
  - CPU seconds used in building a document index
    - 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

# Efficiency Metrics

- Query latency
  - The amount of time a user must wait after issuing a query before receiving a response, measured in milliseconds
  - Often measured with the median
- Indexing temporary space
  - Amount of temporary disk space used while creating an index
- Index size
  - Amount of storage necessary to store the index files

# 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
- Analyze performance of individual queries