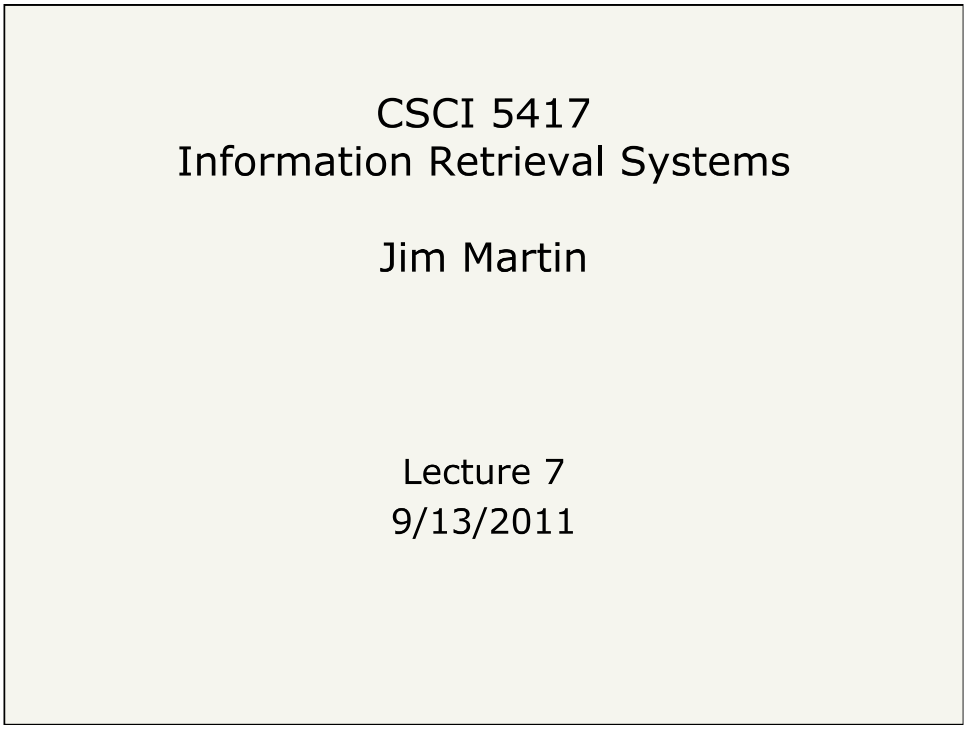
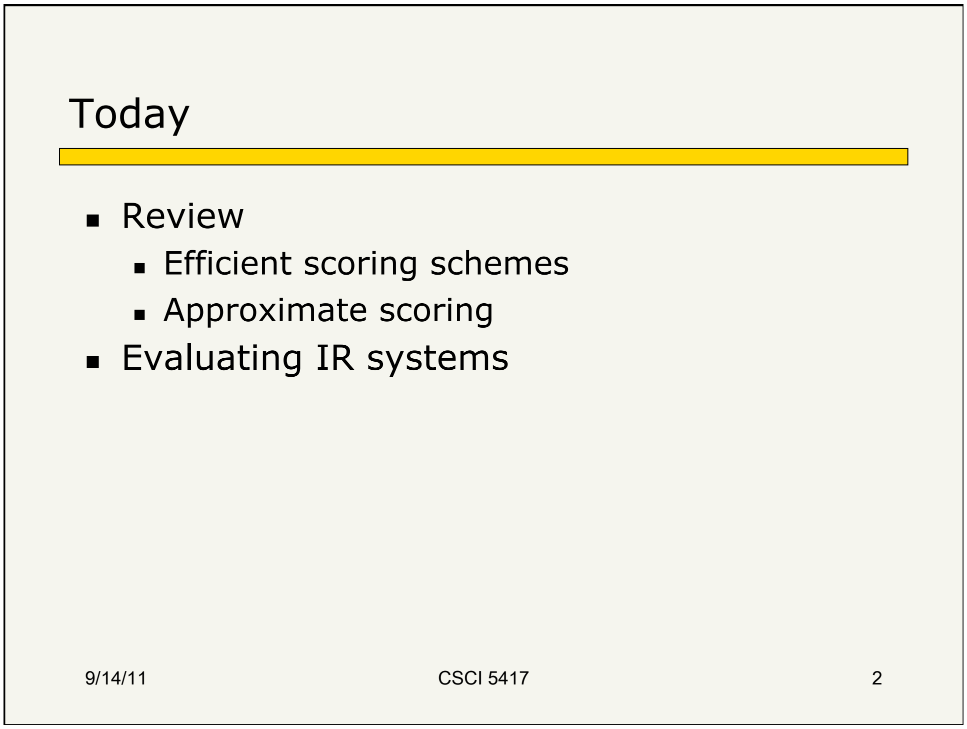


CSCI 5417
Information Retrieval Systems

Jim Martin

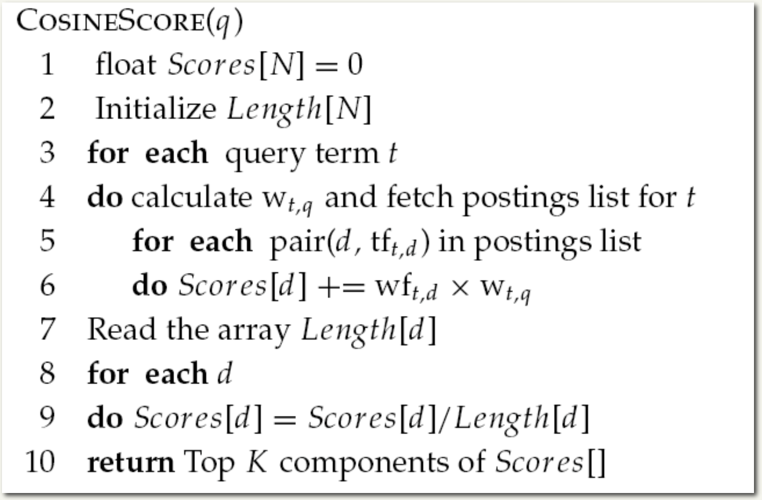
Lecture 7
9/13/2011



Today

- Review
  - Efficient scoring schemes
  - Approximate scoring
- Evaluating IR systems

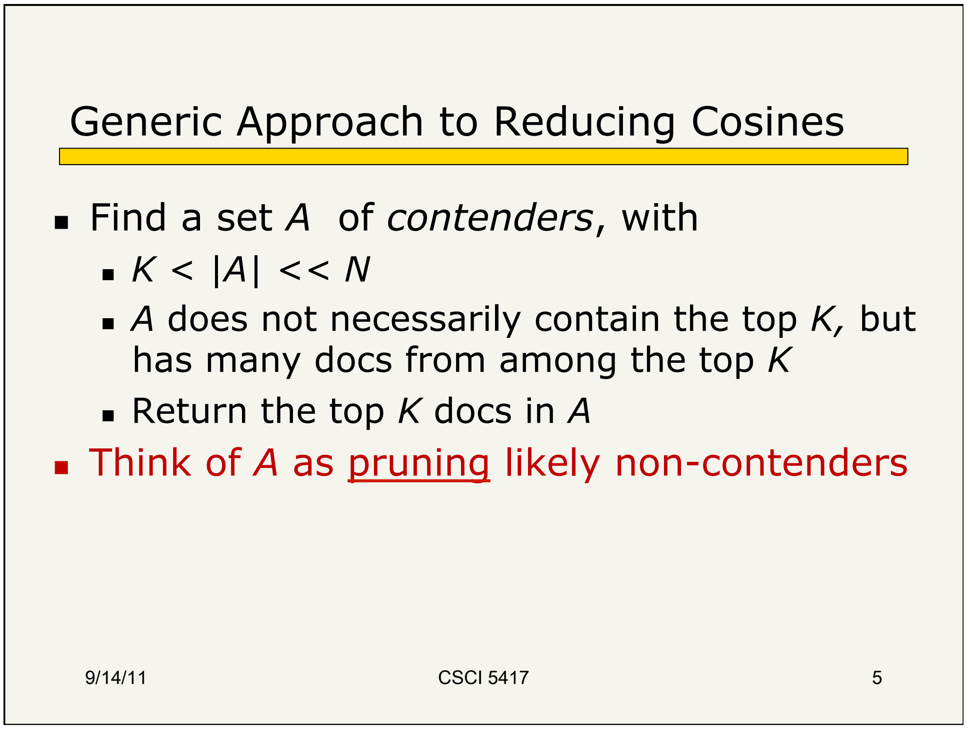
Normal Cosine Scoring



```
CosineScore($q$)
 float $Scores[N] = 0$
 Initialize $Length[N]$
 for each query term $t$
 do calculate $w_{t,q}$ and fetch postings list for $t$
     for each pair($d$, $tf_{t,d}$) in postings list
     do $Scores[d] += wf_{t,d} \times w_{t,q}$
 Read the array $Length[d]$
 for each $d$
 do $Scores[d] = Scores[d]/Length[d]$
 return Top $K$ components of $Scores[]$
```

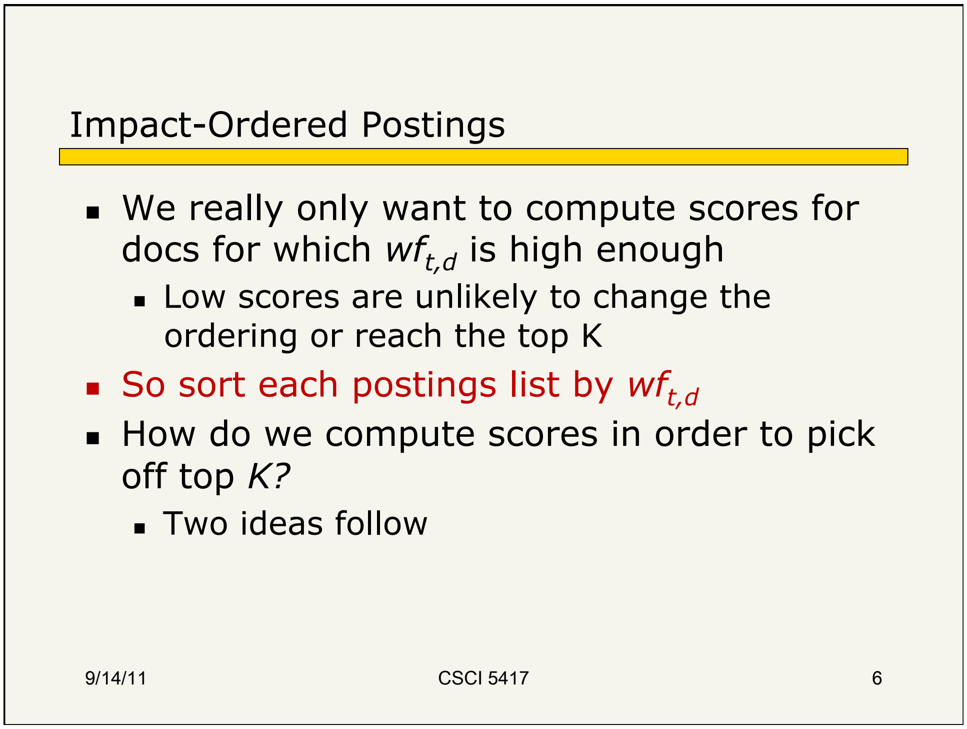
Speedups...

- Compute the cosines faster
- Don’t compute as many cosines



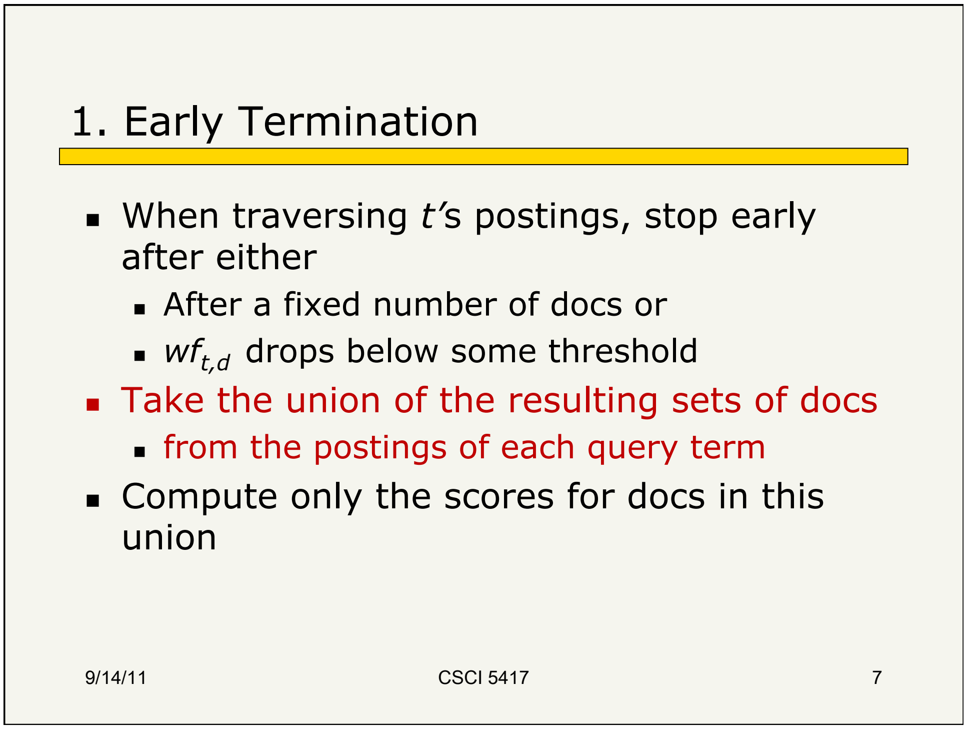
Generic Approach to Reducing Cosines

- Find a set *A* of *contenders*, with
  - $K < |A| << N$
  - *A* does not necessarily contain the top *K,* but has many docs from among the top *K*
  - Return the top *K* docs in *A*
- Think of *A* as pruning likely non-contenders



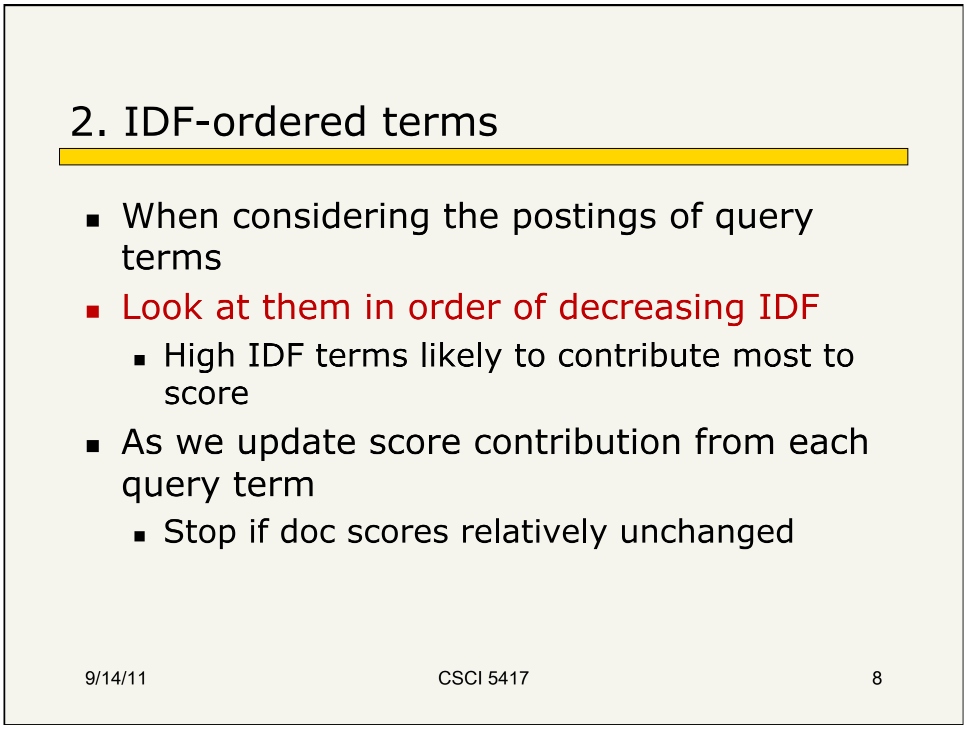
Impact-Ordered Postings

- We really only want to compute scores for docs for which $wf_{t,d}$ is high enough
  - Low scores are unlikely to change the ordering or reach the top K
- So sort each postings list by $wf_{t,d}$
- How do we compute scores in order to pick off top *K?*
  - Two ideas follow



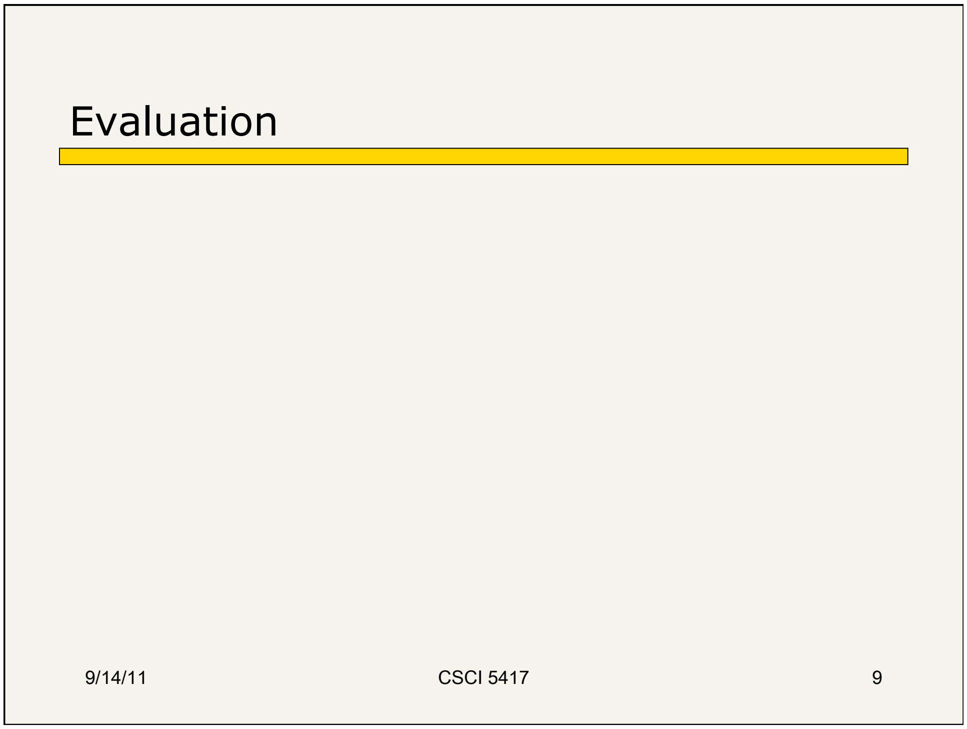
1. Early Termination

- When traversing *t′*s postings, stop early after either
  - After a fixed number of docs or
  - $wf_{t,d}$ drops below some threshold
- Take the union of the resulting sets of docs
  - from the postings of each query term
- Compute only the scores for docs in this union

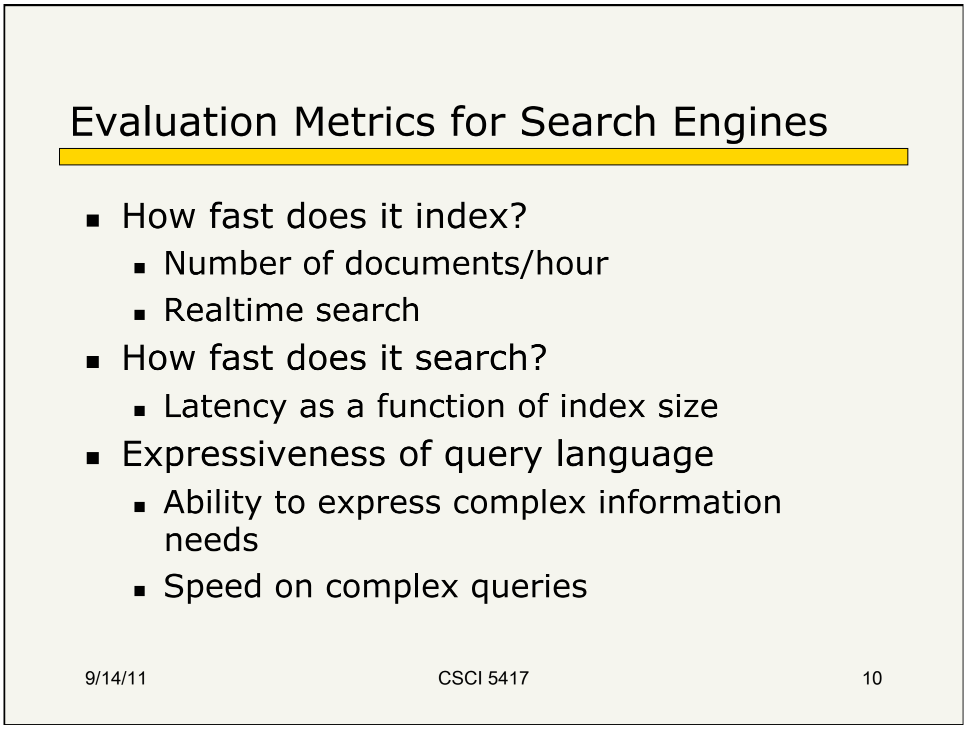


2. IDF-ordered terms

- When considering the postings of query terms
- Look at them in order of decreasing IDF
  - High IDF terms likely to contribute most to score
- As we update score contribution from each query term
  - Stop if doc scores relatively unchanged

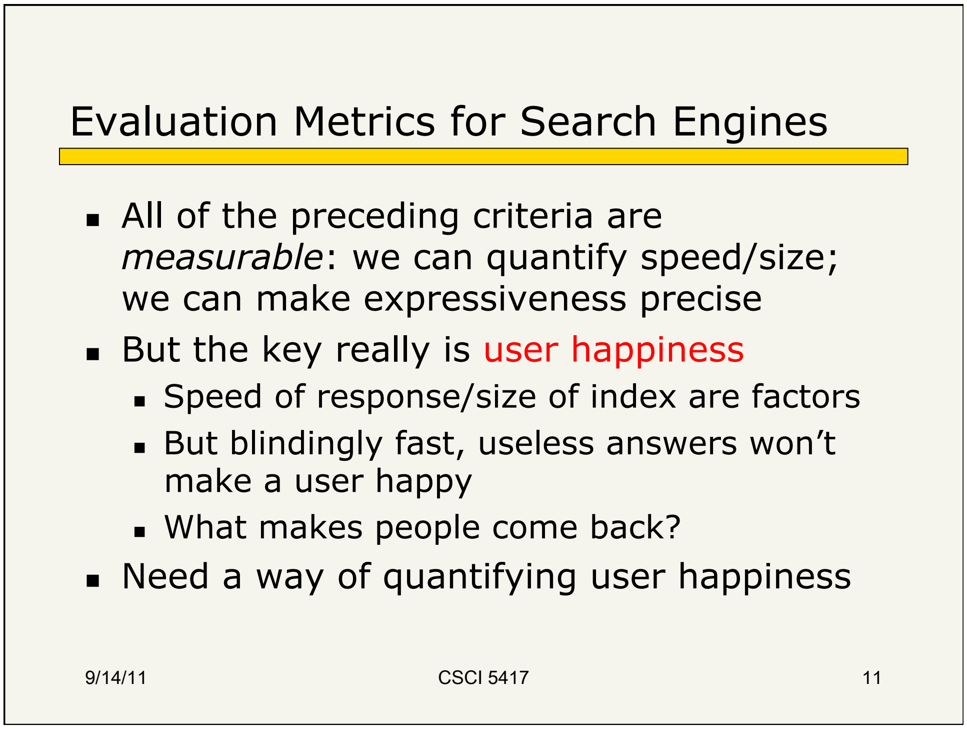


Evaluation



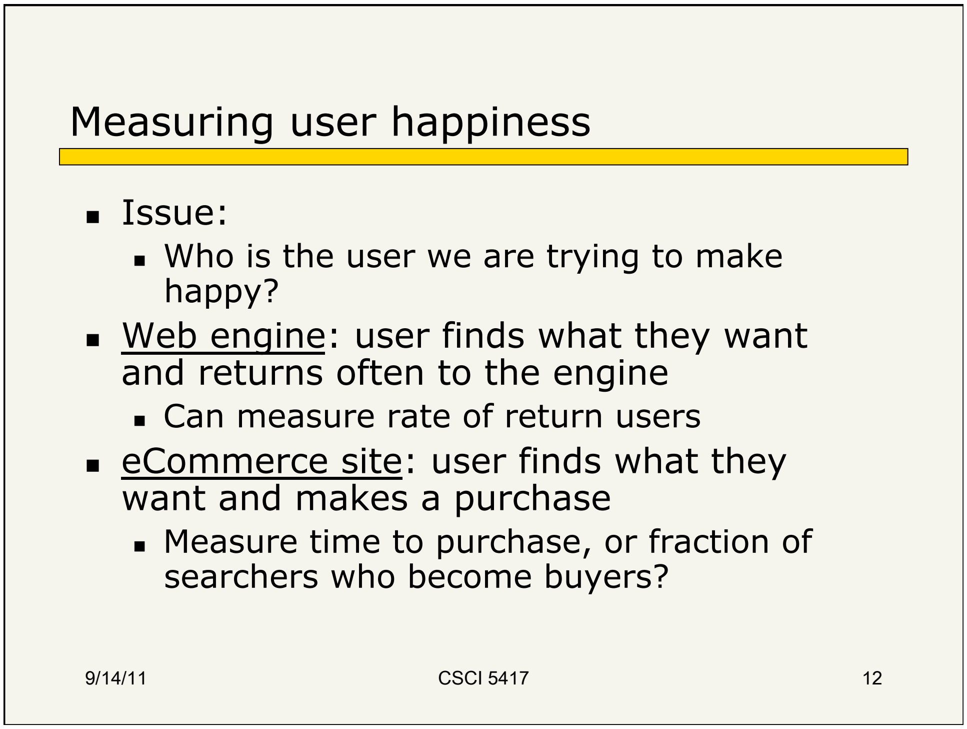
Evaluation Metrics for Search Engines

- How fast does it index?
  - Number of documents/hour
  - Realtime search
- How fast does it search?
  - Latency as a function of index size
- Expressiveness of query language
  - Ability to express complex information needs
  - Speed on complex queries



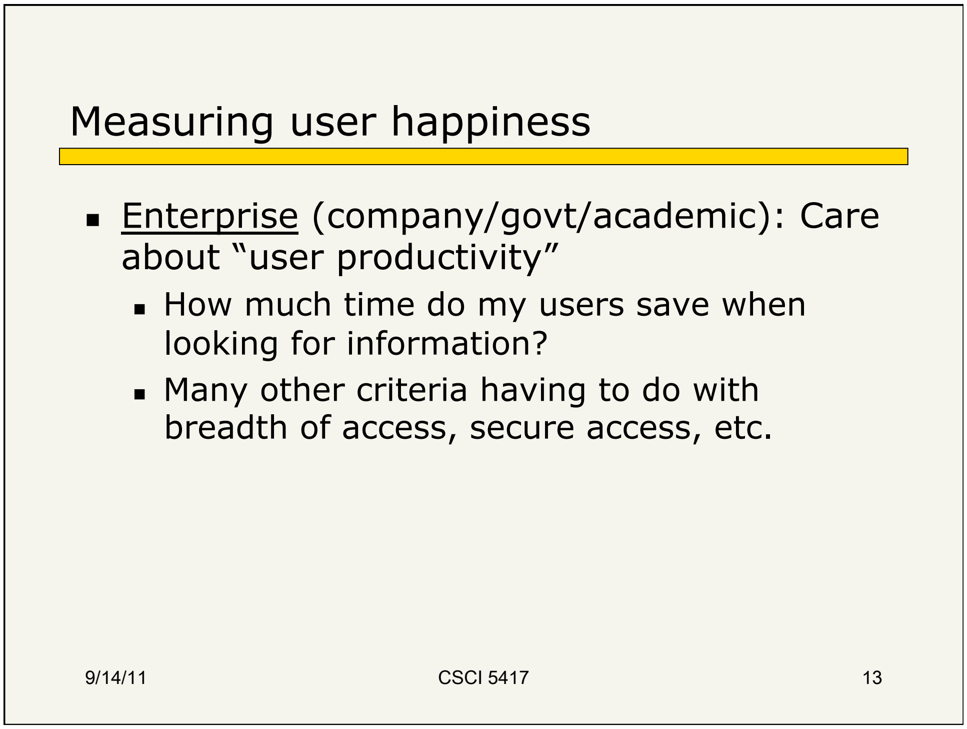
Evaluation Metrics for Search Engines

- All of the preceding criteria are *measurable*: we can quantify speed/size; we can make expressiveness precise
- But the key really is user happiness
  - Speed of response/size of index are factors
  - But blindingly fast, useless answers won't make a user happy
  - What makes people come back?
- Need a way of quantifying user happiness



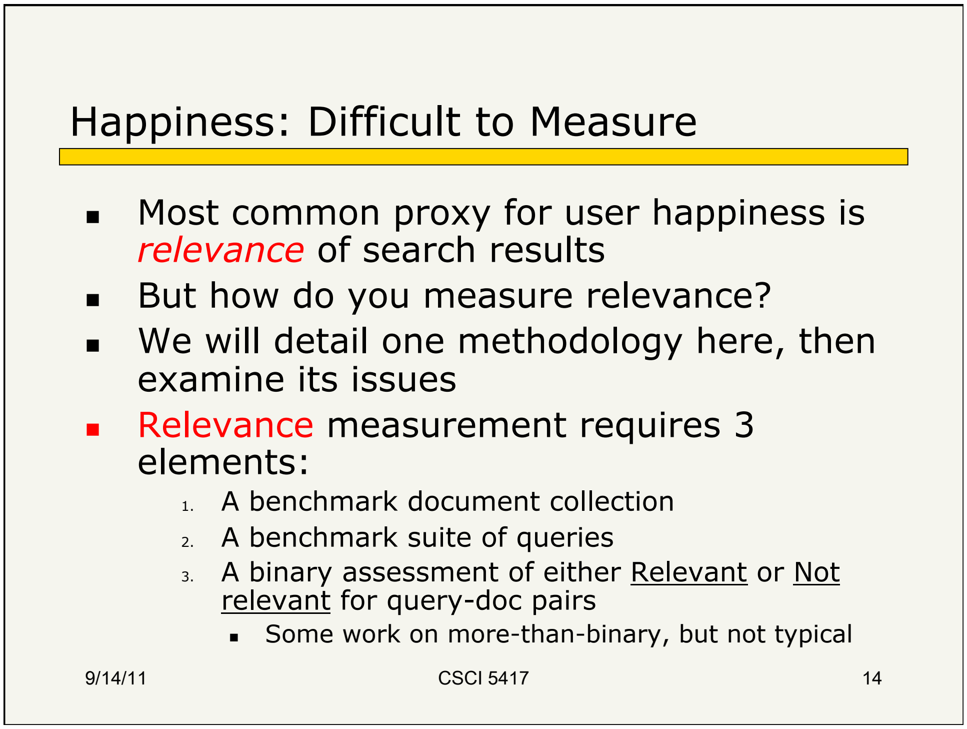
Measuring user happiness

- Issue:
  - Who is the user we are trying to make happy?
- Web engine: user finds what they want and returns often to the engine
  - Can measure rate of return users
- eCommerce site: user finds what they want and makes a purchase
  - Measure time to purchase, or fraction of searchers who become buyers?



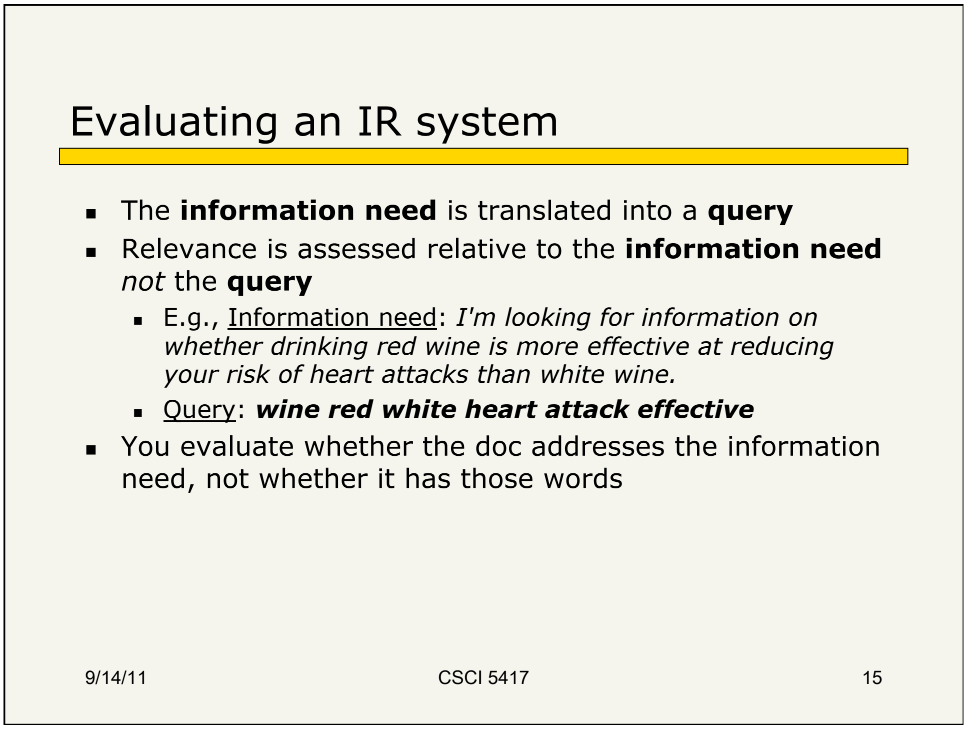
Measuring user happiness

- Enterprise (company/govt/academic): Care about "user productivity"
  - How much time do my users save when looking for information?
  - Many other criteria having to do with breadth of access, secure access, etc.



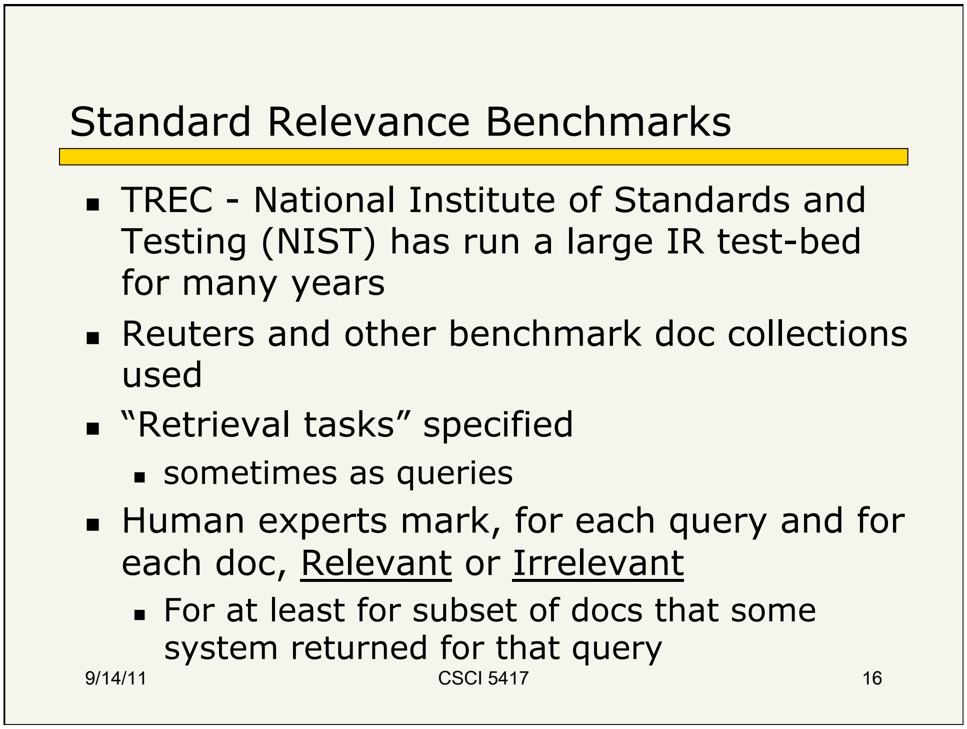
Happiness: Difficult to Measure

- Most common proxy for user happiness is *relevance* of search results
- But how do you measure relevance?
- We will detail one methodology here, then examine its issues
- Relevance measurement requires 3 elements:
  1. A benchmark document collection
  2. A benchmark suite of queries
  3. A binary assessment of either Relevant or Not relevant for query-doc pairs
     - Some work on more-than-binary, but not typical



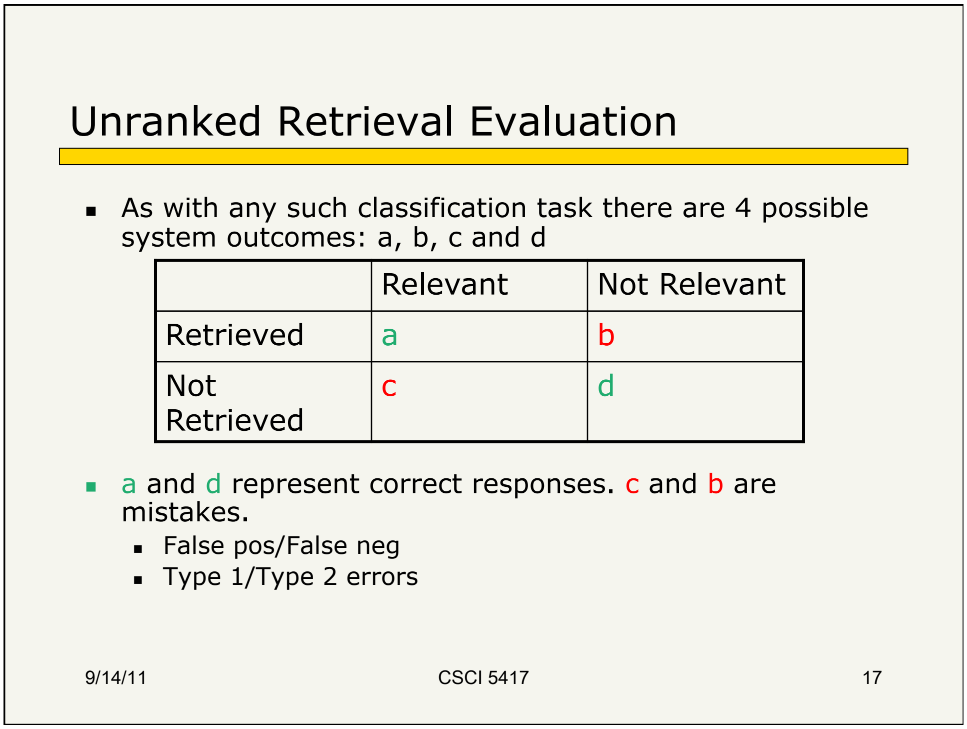
Evaluating an IR system

- The **information need** is translated into a **query**
- Relevance is assessed relative to the **information need** *not* the **query**
  - E.g., Information need: *I'm looking for information on whether drinking red wine is more effective at reducing your risk of heart attacks than white wine.*
  - Query: ***wine red white heart attack effective***
- You evaluate whether the doc addresses the information need, not whether it has those words



Standard Relevance Benchmarks

- TREC - National Institute of Standards and Testing (NIST) has run a large IR test-bed for many years
- Reuters and other benchmark doc collections used
- “Retrieval tasks” specified
  - sometimes as queries
- Human experts mark, for each query and for each doc, Relevant or Irrelevant
  - For at least for subset of docs that some system returned for that query

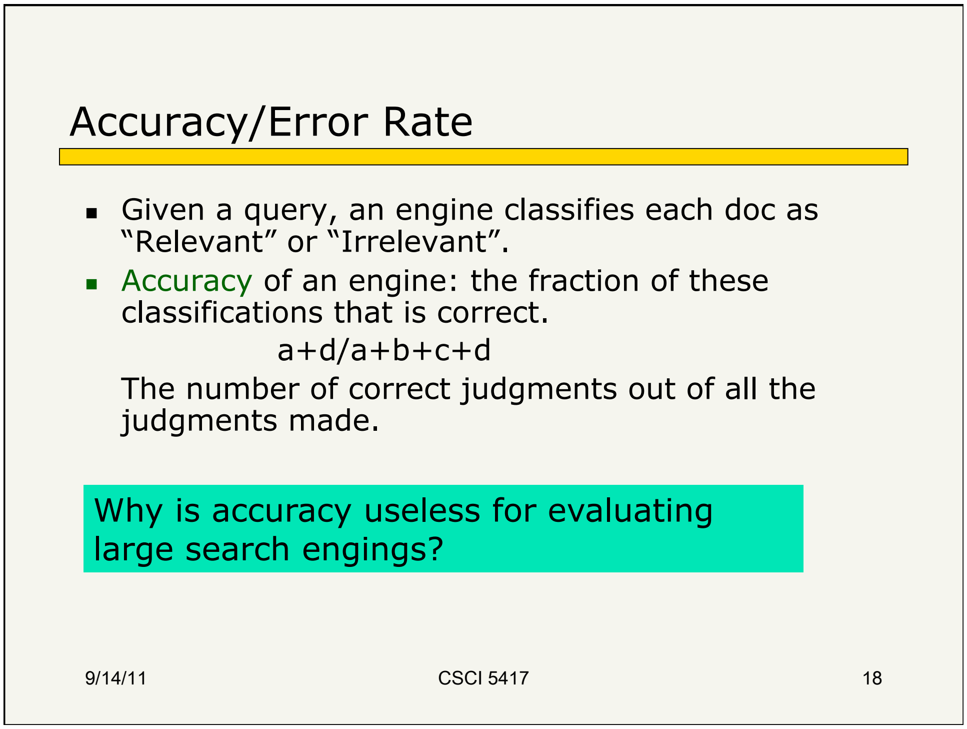


Unranked Retrieval Evaluation

- As with any such classification task there are 4 possible system outcomes: a, b, c and d

	Relevant	Not Relevant
Retrieved	a	b
Not Retrieved	c	d

- a and d represent correct responses. c and b are mistakes.
  - False pos/False neg
  - Type 1/Type 2 errors



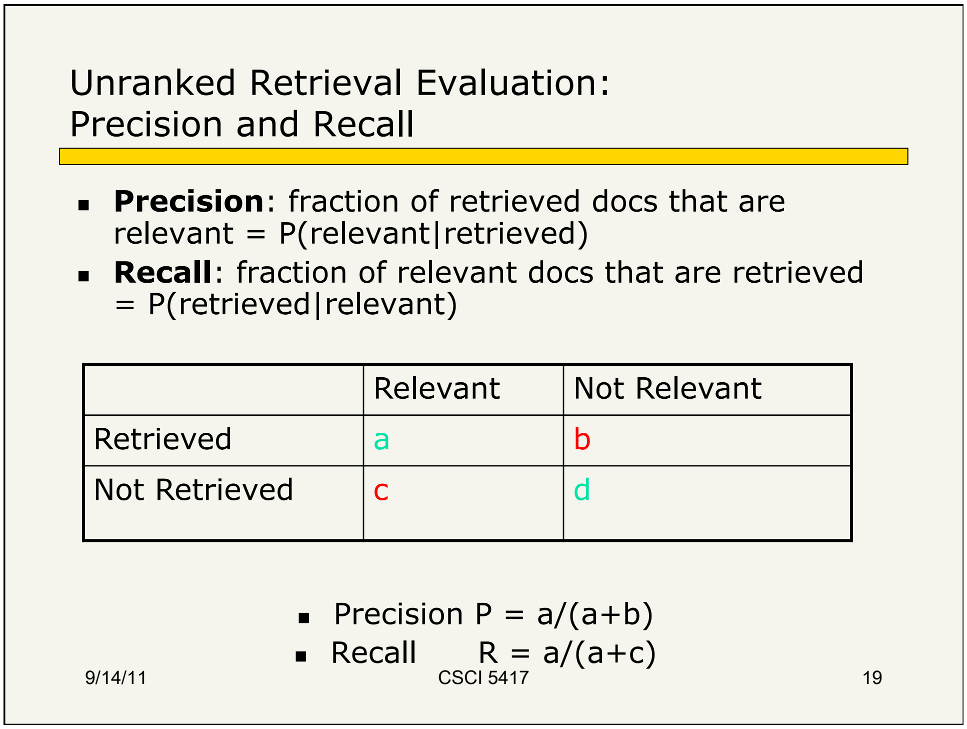
Accuracy/Error Rate

- Given a query, an engine classifies each doc as “Relevant” or “Irrelevant”.
- Accuracy of an engine: the fraction of these classifications that is correct.

  a+d/a+b+c+d

  The number of correct judgments out of all the judgments made.

Why is accuracy useless for evaluating large search engings?

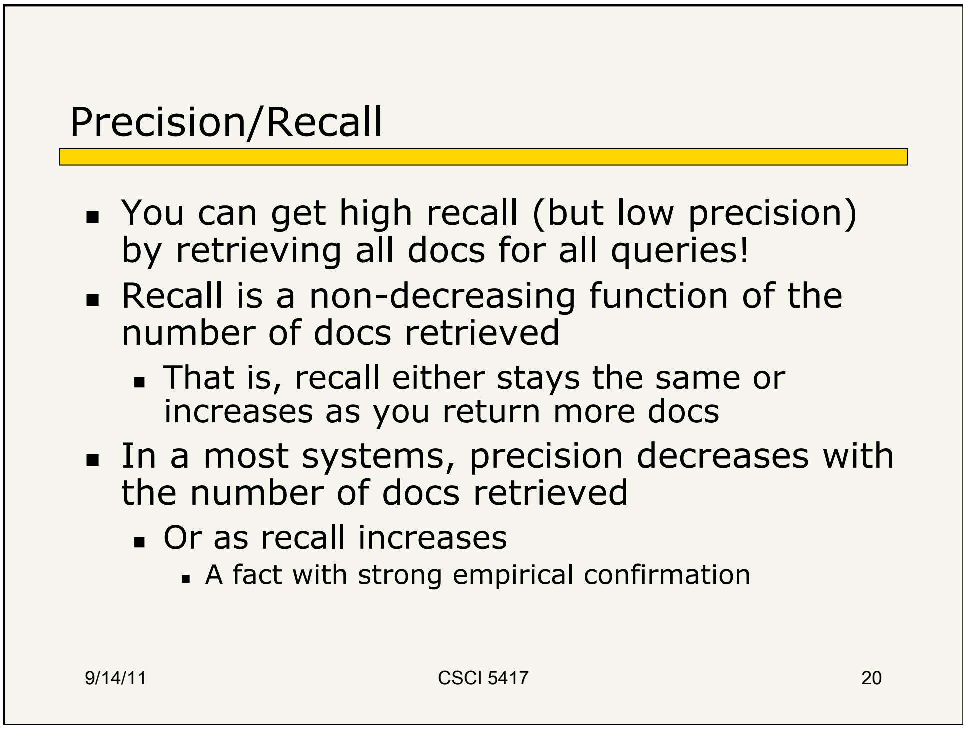


Unranked Retrieval Evaluation: Precision and Recall

- **Precision**: fraction of retrieved docs that are relevant = P(relevant|retrieved)
- **Recall**: fraction of relevant docs that are retrieved = P(retrieved|relevant)

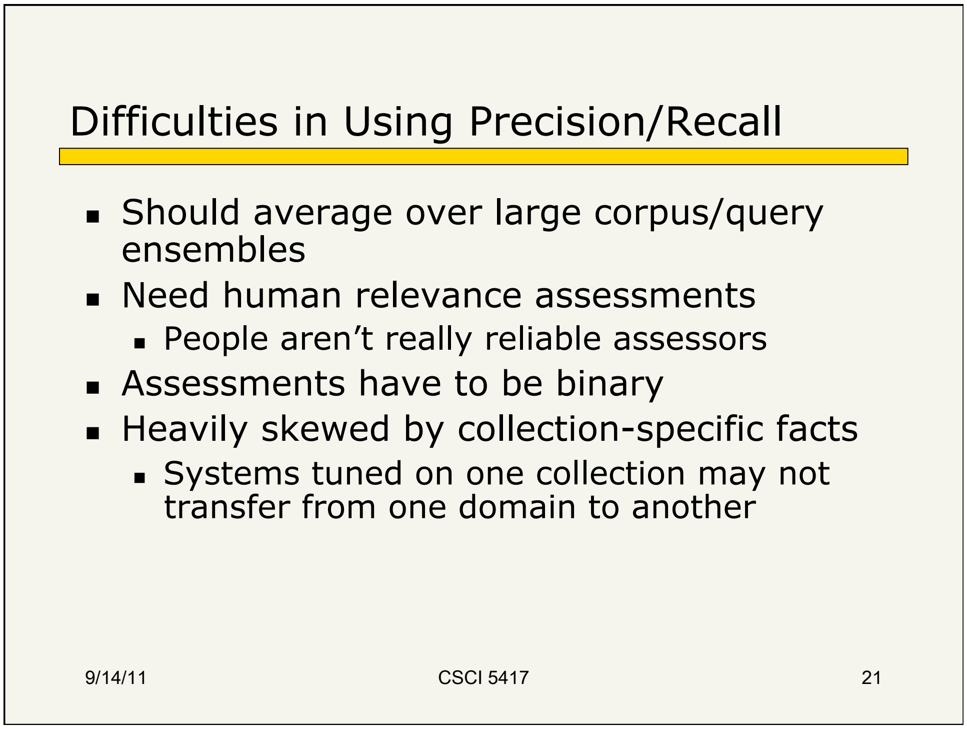
	Relevant	Not Relevant
Retrieved	a	b
Not Retrieved	c	d

- Precision P = a/(a+b)
- Recall R = a/(a+c)



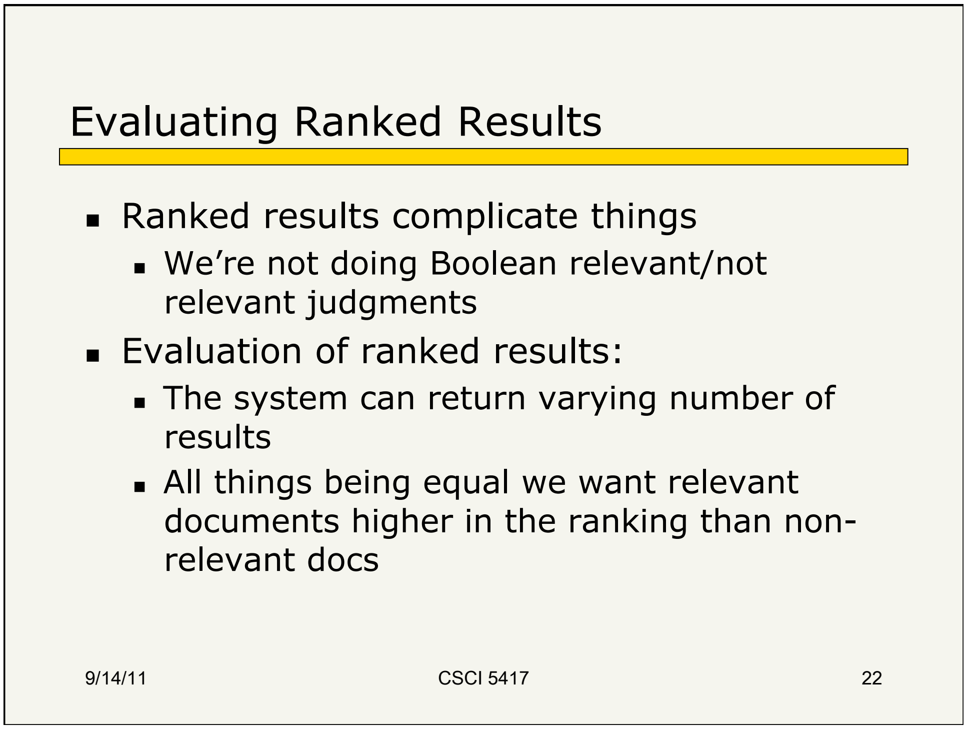
Precision/Recall

- You can get high recall (but low precision) by retrieving all docs for all queries!
- Recall is a non-decreasing function of the number of docs retrieved
  - That is, recall either stays the same or increases as you return more docs
- In a most systems, precision decreases with the number of docs retrieved
  - Or as recall increases
    - A fact with strong empirical confirmation



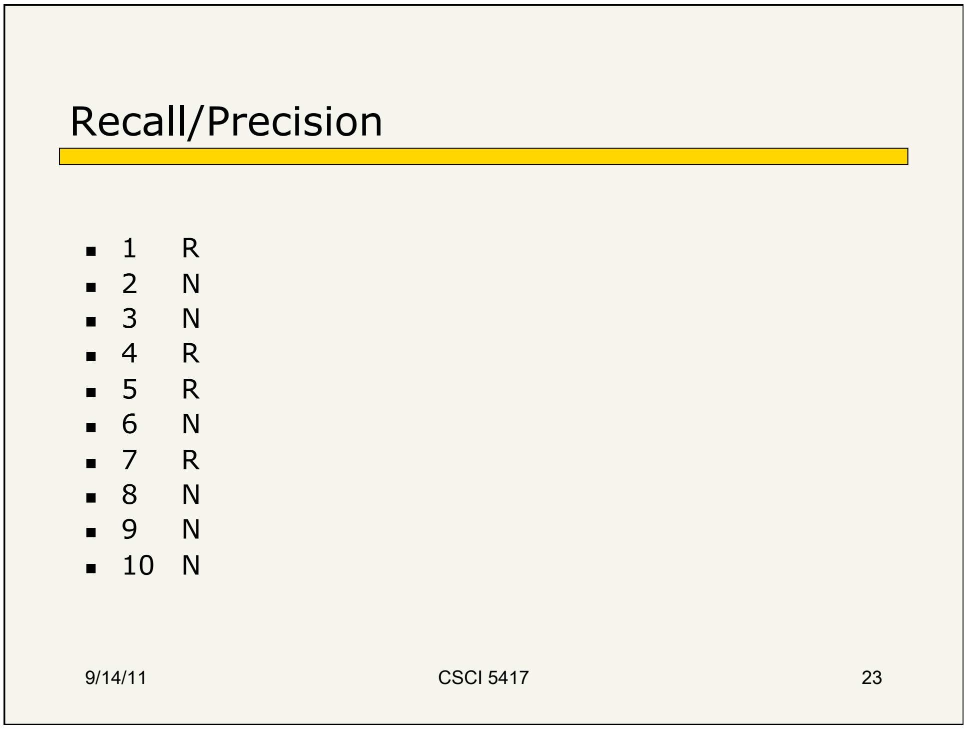
Difficulties in Using Precision/Recall

- Should average over large corpus/query ensembles
- Need human relevance assessments
  - People aren’t really reliable assessors
- Assessments have to be binary
- Heavily skewed by collection-specific facts
  - Systems tuned on one collection may not transfer from one domain to another



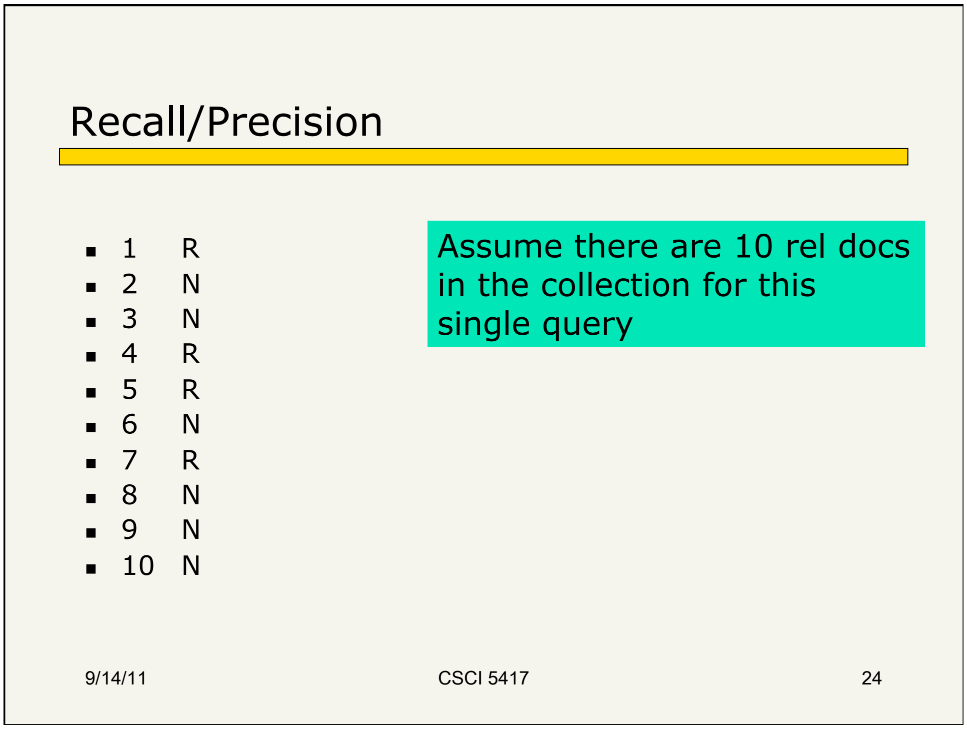
Evaluating Ranked Results

- Ranked results complicate things
  - We’re not doing Boolean relevant/not relevant judgments
- Evaluation of ranked results:
  - The system can return varying number of results
  - All things being equal we want relevant documents higher in the ranking than non-relevant docs



Recall/Precision

1	R
2	N
3	N
4	R
5	R
6	N
7	R
8	N
9	N
10	N



Recall/Precision

1	R
2	N
3	N
4	R
5	R
6	N
7	R
8	N
9	N
10	N

Assume there are 10 rel docs in the collection for this single query

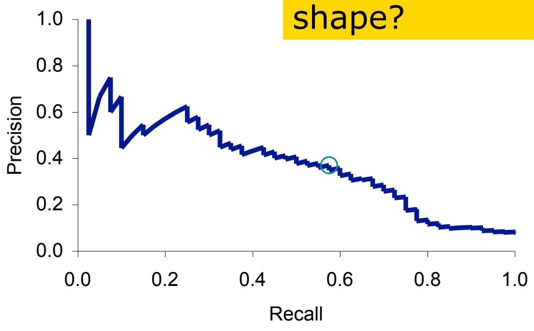
Recall/Precision

1	R
2	N
3	N
4	R
5	R
6	N
7	R
8	N
9	N
10	N

R	P
10%	100%
10	50
10	33
20	50
30	60
30	50
40	57
40	50
40	44
40	40

Assume 10 rel docs in collection

A Precision-Recall curve



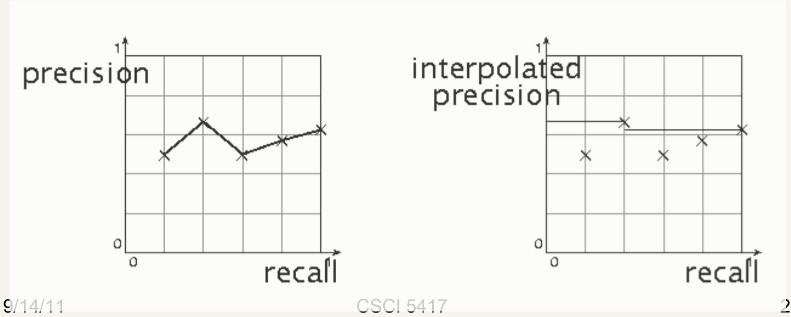
Why the sawtooth shape?

Averaging over queries

- A precision-recall graph for a single query isn't a very useful piece of information
- You need to average performance over a whole bunch of queries.
- But there's a technical issue:
  - Precision-recall calculations fill only some points on the graph
  - How do you determine a value (interpolate) between the points?

Interpolated precision

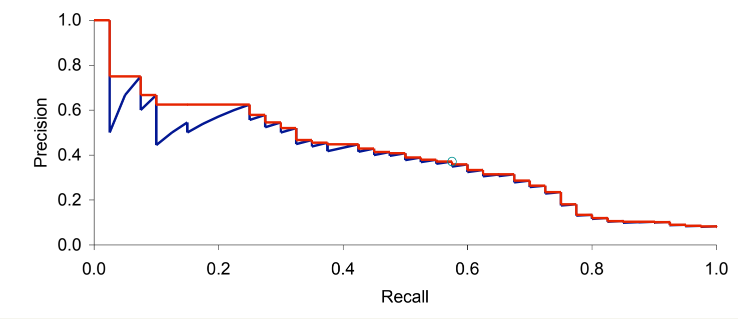
- Idea: if locally precision increases with increasing recall, then you should get to count that…
- So you max of precisions to right of value



Interpolated Values

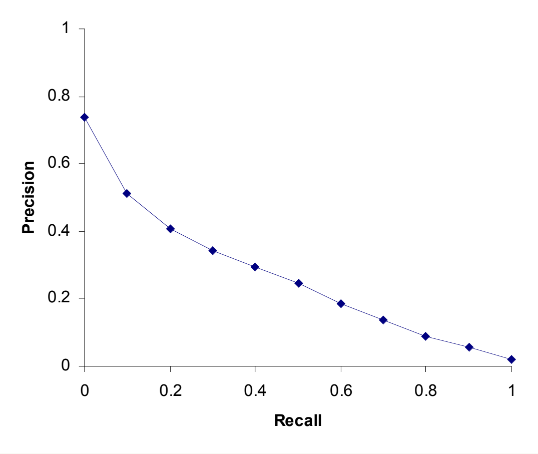
- Ok... Now we can compute R/P pairs across queries... At standard points.
- The usual thing to do is to measure Precision at fixed (11) recall levels for each query.
  - 0 .1 .2 .3 1

An Interpolated Precision-Recall Curve



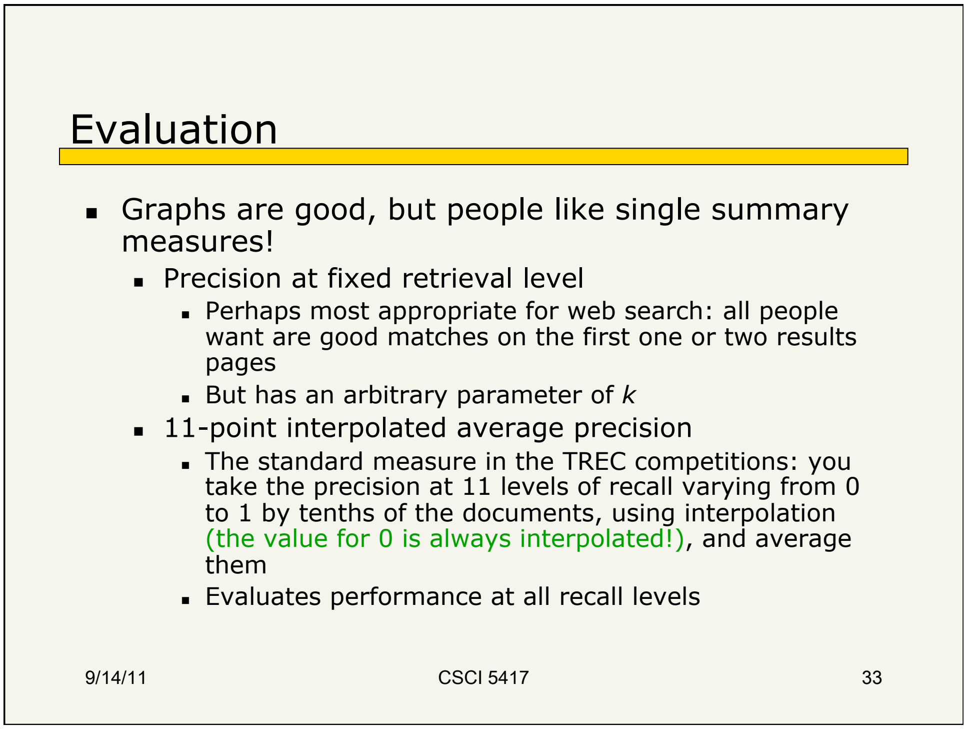
Typical (good) 11 point precisions

- SabIR/Cornell 8A1 11pt precision from TREC 8 (1999)



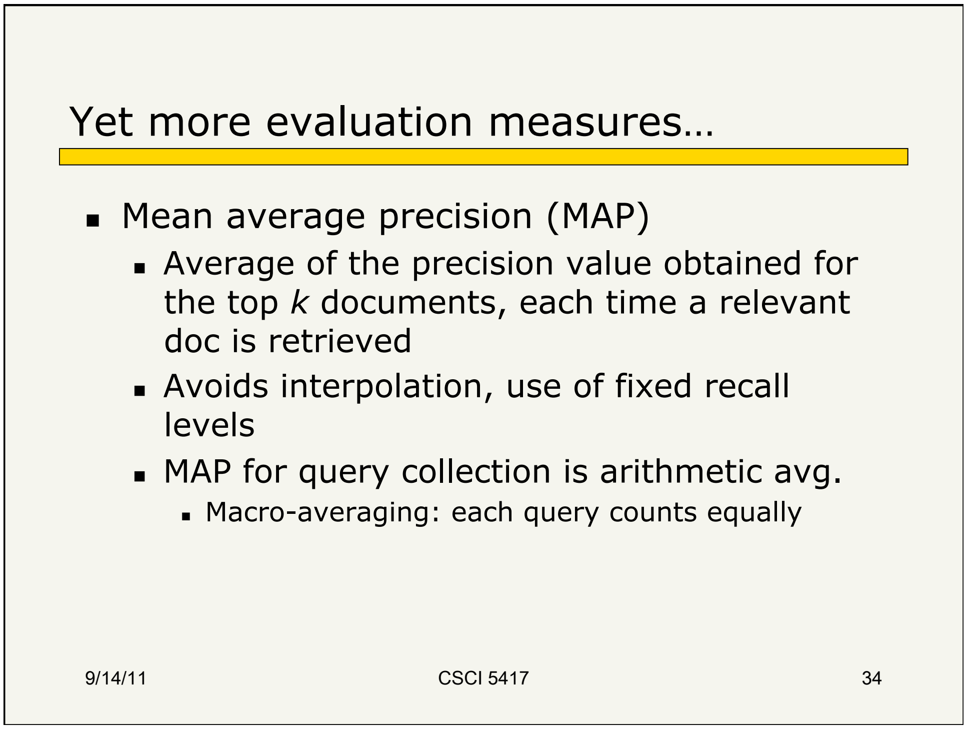
Break

- trec_eval



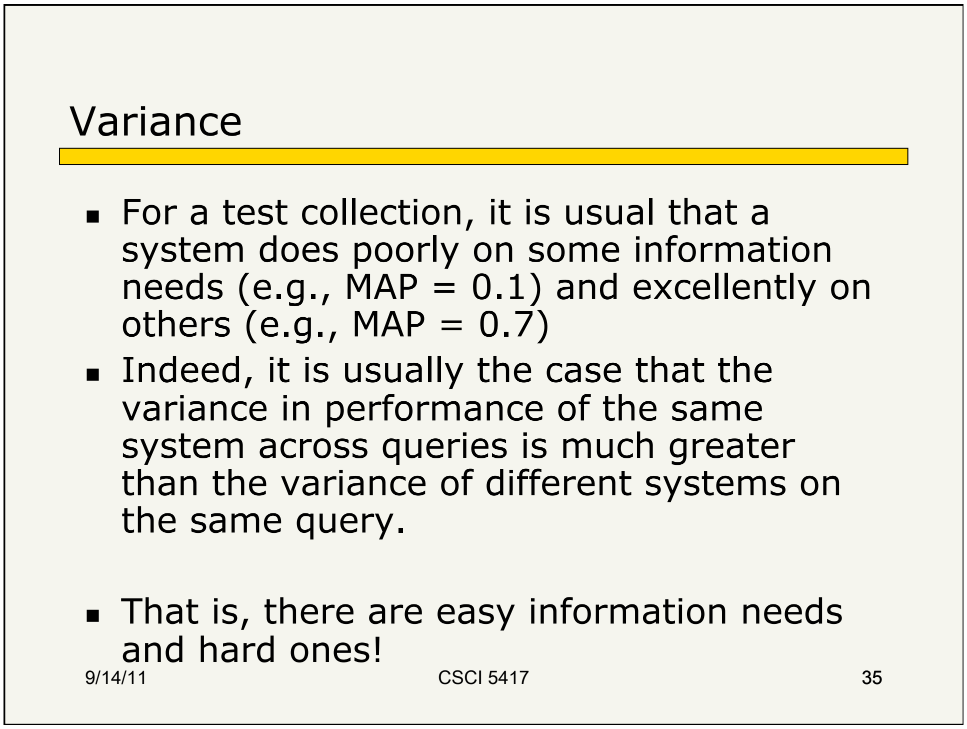
Evaluation

- Graphs are good, but people like single summary measures!
  - Precision at fixed retrieval level
    - Perhaps most appropriate for web search: all people want are good matches on the first one or two results pages
    - But has an arbitrary parameter of *k*
  - 11-point interpolated average precision
    - The standard measure in the TREC competitions: you take the precision at 11 levels of recall varying from 0 to 1 by tenths of the documents, using interpolation (the value for 0 is always interpolated!), and average them
    - Evaluates performance at all recall levels



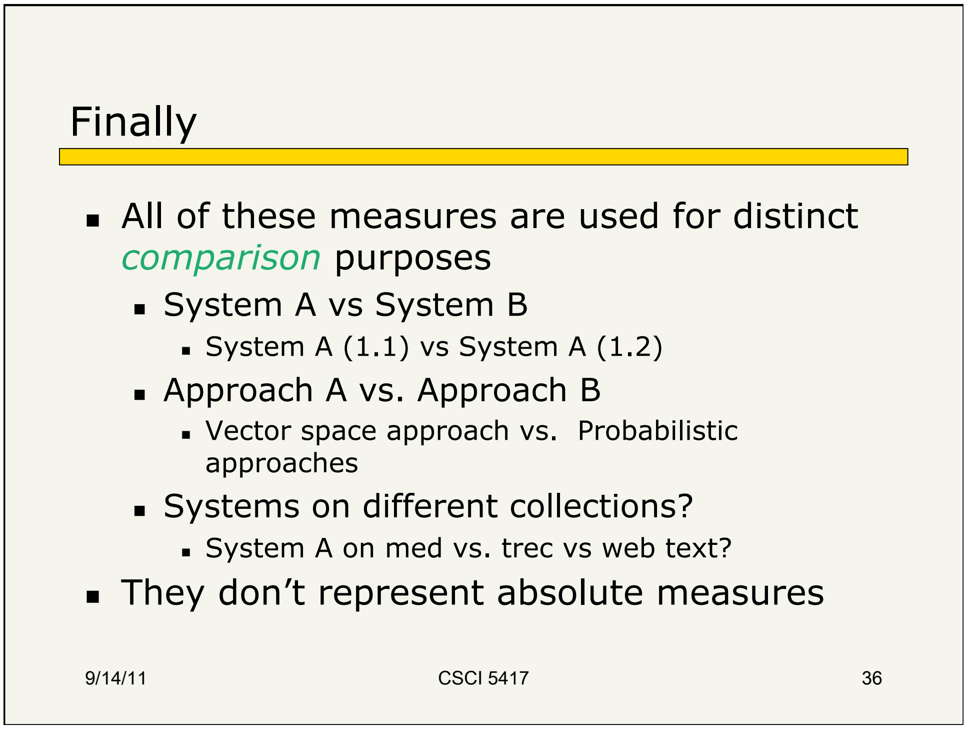
Yet more evaluation measures…

- Mean average precision (MAP)
  - Average of the precision value obtained for the top *k* documents, each time a relevant doc is retrieved
  - Avoids interpolation, use of fixed recall levels
  - MAP for query collection is arithmetic avg.
    - Macro-averaging: each query counts equally



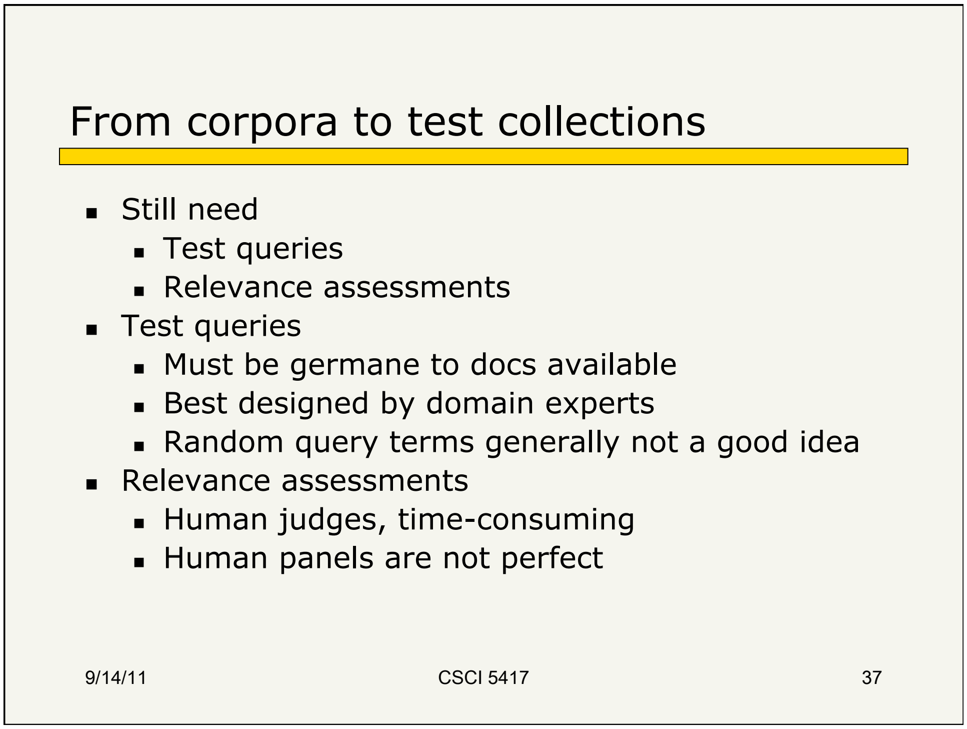
Variance

- For a test collection, it is usual that a system does poorly on some information needs (e.g., MAP = 0.1) and excellently on others (e.g., MAP = 0.7)
- Indeed, it is usually the case that the variance in performance of the same system across queries is much greater than the variance of different systems on the same query.
- That is, there are easy information needs and hard ones!



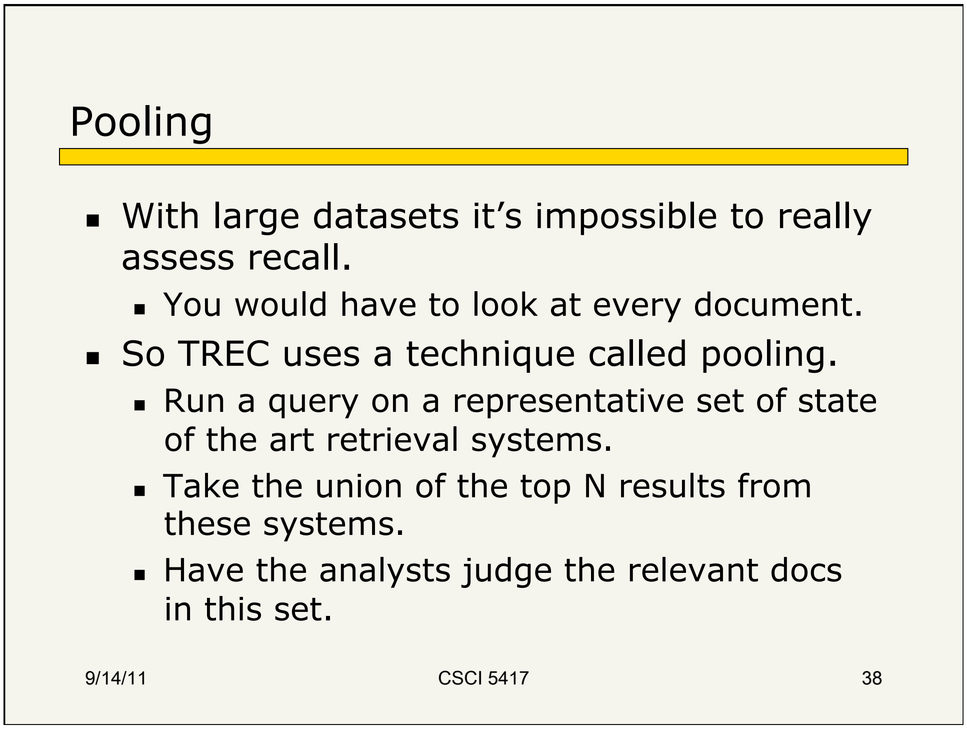
Finally

- All of these measures are used for distinct *comparison* purposes
  - System A vs System B
    - System A (1.1) vs System A (1.2)
  - Approach A vs. Approach B
    - Vector space approach vs. Probabilistic approaches
  - Systems on different collections?
    - System A on med vs. trec vs web text?
- They don’t represent absolute measures



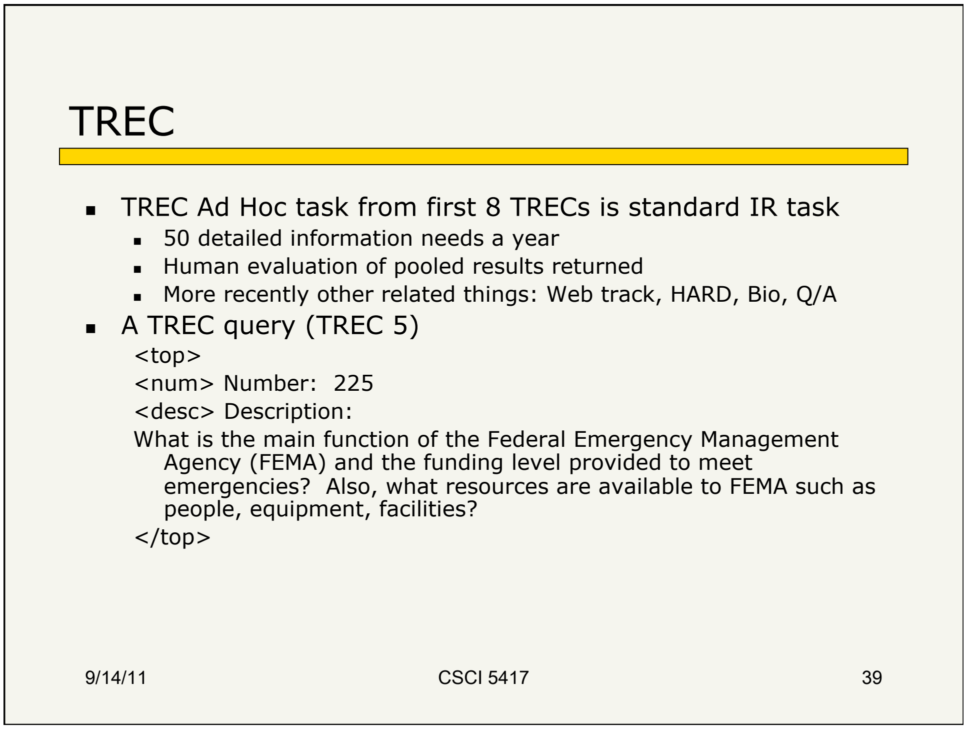
From corpora to test collections

- Still need
  - Test queries
  - Relevance assessments
- Test queries
  - Must be germane to docs available
  - Best designed by domain experts
  - Random query terms generally not a good idea
- Relevance assessments
  - Human judges, time-consuming
  - Human panels are not perfect



Pooling

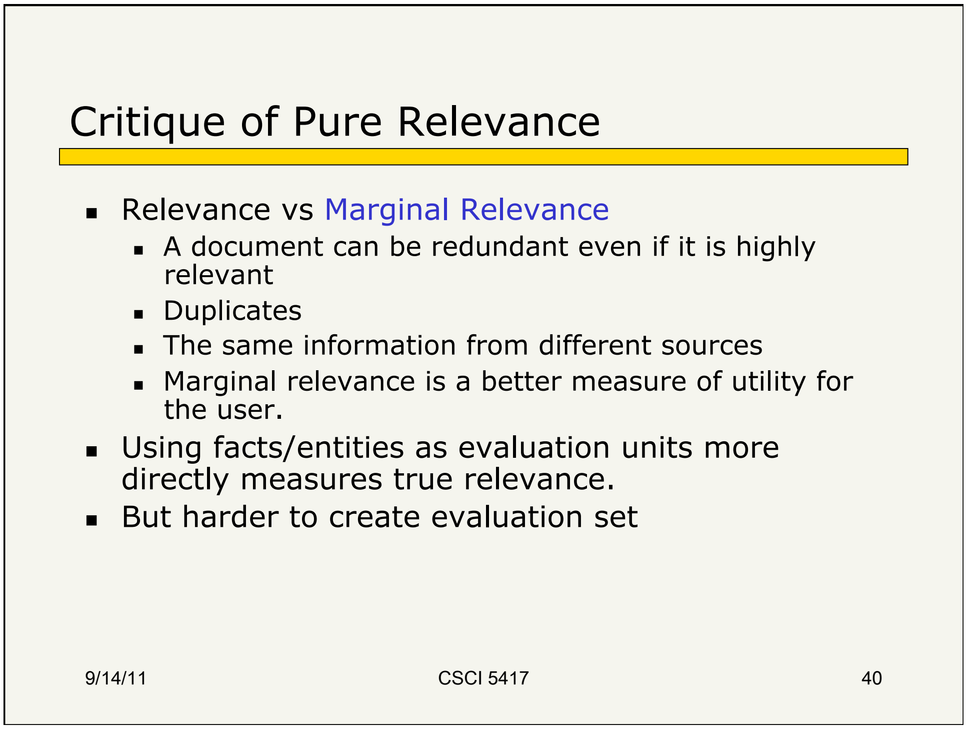
- With large datasets it’s impossible to really assess recall.
  - You would have to look at every document.
- So TREC uses a technique called pooling.
  - Run a query on a representative set of state of the art retrieval systems.
  - Take the union of the top N results from these systems.
  - Have the analysts judge the relevant docs in this set.



TREC

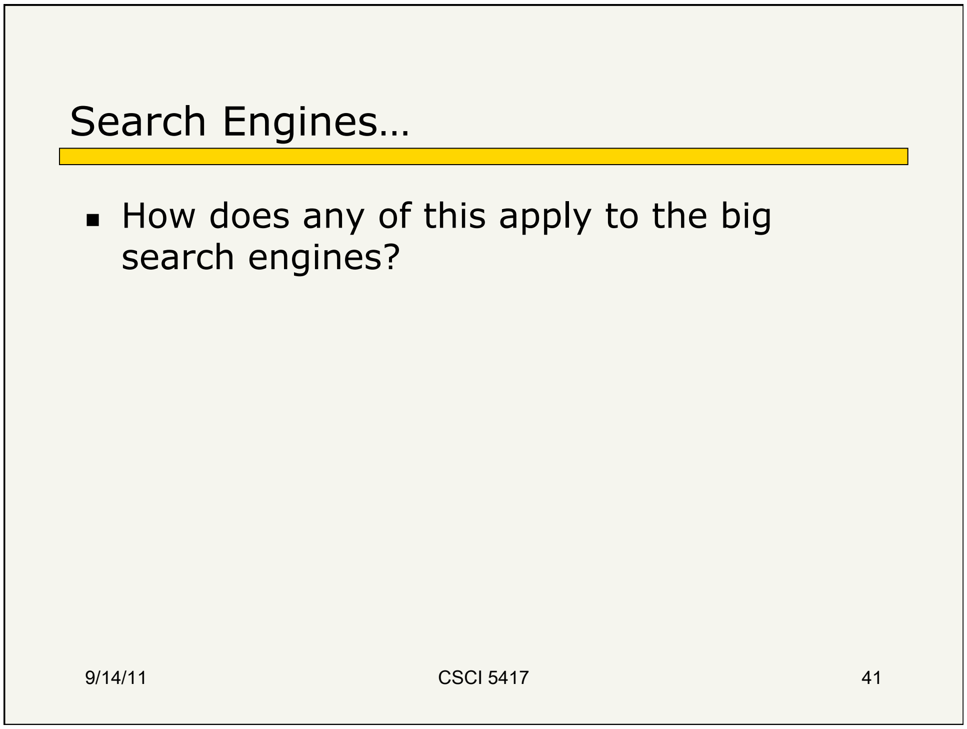
- TREC Ad Hoc task from first 8 TRECs is standard IR task
  - 50 detailed information needs a year
  - Human evaluation of pooled results returned
  - More recently other related things: Web track, HARD, Bio, Q/A
- A TREC query (TREC 5)

```
<top>
<num> Number:  225
<desc> Description:
What is the main function of the Federal Emergency Management
    Agency (FEMA) and the funding level provided to meet
    emergencies?  Also, what resources are available to FEMA such as
    people, equipment, facilities?
</top>
```



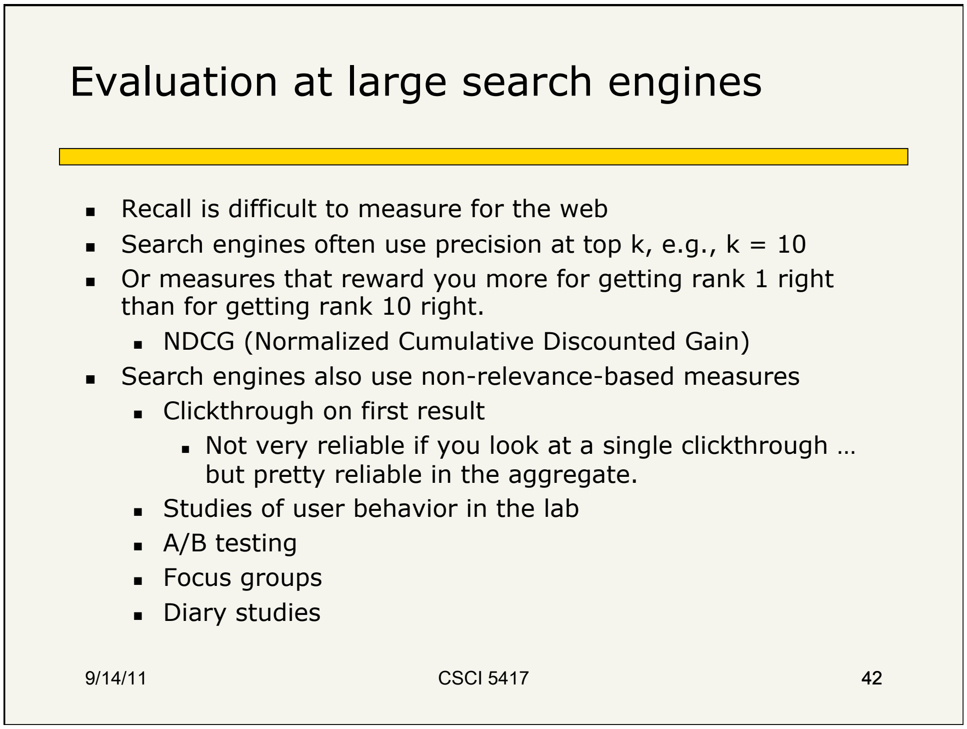
Critique of Pure Relevance

- Relevance vs Marginal Relevance
  - A document can be redundant even if it is highly relevant
  - Duplicates
  - The same information from different sources
  - Marginal relevance is a better measure of utility for the user.
- Using facts/entities as evaluation units more directly measures true relevance.
- But harder to create evaluation set



Search Engines…

- How does any of this apply to the big search engines?



Evaluation at large search engines

- Recall is difficult to measure for the web
- Search engines often use precision at top k, e.g., k = 10
- Or measures that reward you more for getting rank 1 right than for getting rank 10 right.
  - NDCG (Normalized Cumulative Discounted Gain)
- Search engines also use non-relevance-based measures
  - Clickthrough on first result
    - Not very reliable if you look at a single clickthrough … but pretty reliable in the aggregate.
  - Studies of user behavior in the lab
  - A/B testing
  - Focus groups
  - Diary studies

A/B testing

- Purpose: Test a single innovation
- Prerequisite: You have a system up and running.
- Have most users use old system
- Divert a small proportion of traffic (e.g., 1%) to the new system that includes the innovation
- Evaluate with an “automatic” measure like clickthrough on first result
- Now we can directly see if the innovation does improve user happiness.
- Probably the evaluation methodology that large search engines trust most

Next Time

Relevance feedback

Should have read up through Chapter 9.