



Εύρεση & Διαχείριση Πληροφορίας στον Παγκόσμιο Ιστό

Διδάσκων –
Δημήτριος Κατσαρός



EVALUATING SEARCH ENGINES



Measures for a search engine

- How fast does it index
 - Number of documents/hour
 - (Average document size)
- 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
- Uncluttered UI
- Is it free?



Measures for a search engine

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



Measuring user happiness

- Issue: who is the user we are trying to make happy?
 - Depends on the setting
- Web engine:
 - User finds what they want and return to the engine
 - Can measure rate of return users
 - User completes their task – search as a means, not end
 - See Russell <http://dmrussell.googlepages.com/JCDL-talk-June-2007-short.pdf>
- eCommerce site: user finds what they want and buy
 - Is it the end-user, or the eCommerce site, whose happiness we measure?
 - 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: elusive to measure

- Most common proxy: *relevance* of search results
- But how do you measure relevance?
- We will detail a methodology here, then examine its issues
- Relevance measurement requires 3 elements:
 1. A benchmark document collection
 2. A benchmark suite of queries
 3. A usually binary assessment of either Relevant or Nonrelevant for each query and each document
 - Some work on more-than-binary, but not the standard



Evaluating an IR system

- Note: 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 these words



Standard relevance benchmarks

- TREC - National Institute of Standards and Technology (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 Nonrelevant
 - or at least for subset of docs that some system returned for that query



Unranked retrieval evaluation: Precision and Recall

- **Precision:** fraction of retrieved docs that are relevant = $P(\text{relevant} | \text{retrieved})$
- **Recall:** fraction of relevant docs that are retrieved = $P(\text{retrieved} | \text{relevant})$

	Relevant	Nonrelevant
Retrieved	tp	fp
Not Retrieved	fn	tn

- Precision $P = tp / (tp + fp)$
- Recall $R = tp / (tp + fn)$



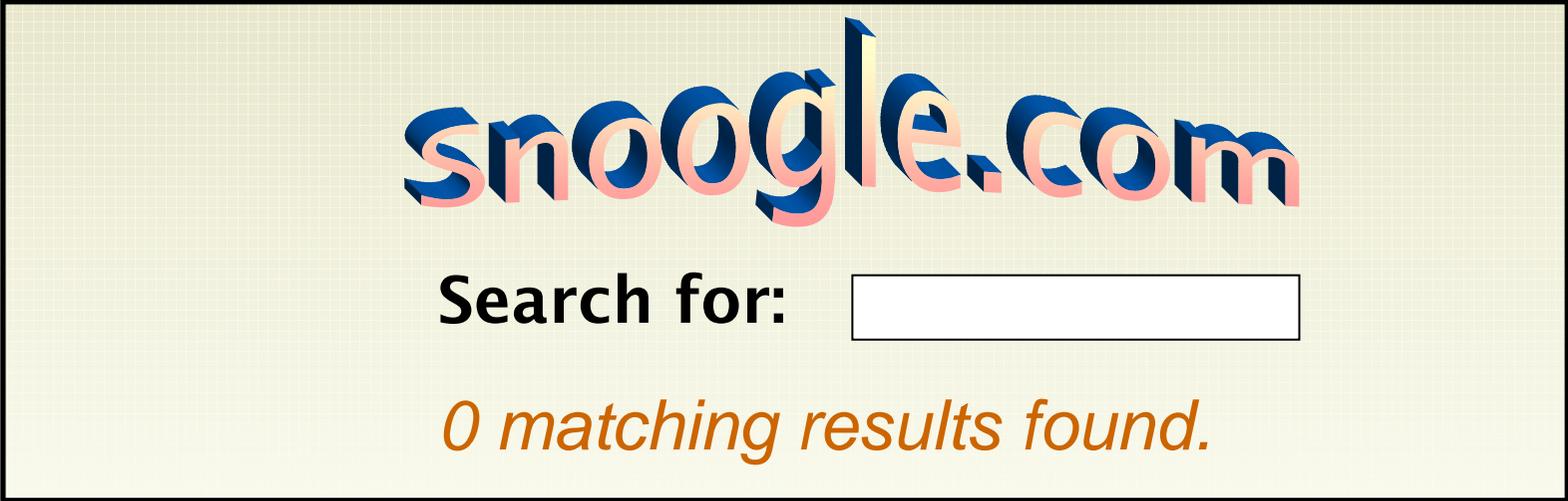
Should we instead use the accuracy measure for evaluation?

- Given a query, an engine classifies each doc as “Relevant” or “Nonrelevant”
- The **accuracy** of an engine: the fraction of these classifications that are correct
 - $(tp + tn) / (tp + fp + fn + tn)$
- **Accuracy** is a commonly used evaluation measure in machine learning classification work
- Why is this not a very useful evaluation measure in IR?



Why not just use accuracy?

- How to build a 99.9999% accurate search engine on a low budget....



snoogle.com

Search for:

0 matching results found.

- People doing information retrieval *want to find something* and have a certain tolerance for junk.



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
- In a good system, precision decreases as either the number of docs retrieved or recall increases
 - This is not a theorem, but a result with strong empirical confirmation



Difficulties in using precision/recall

- Should average over large document collection/query ensembles
- Need human relevance assessments
 - People aren't reliable assessors
- Assessments have to be binary
 - Nuanced assessments?
- Heavily skewed by collection/authorship
 - Results may not translate from one domain to another



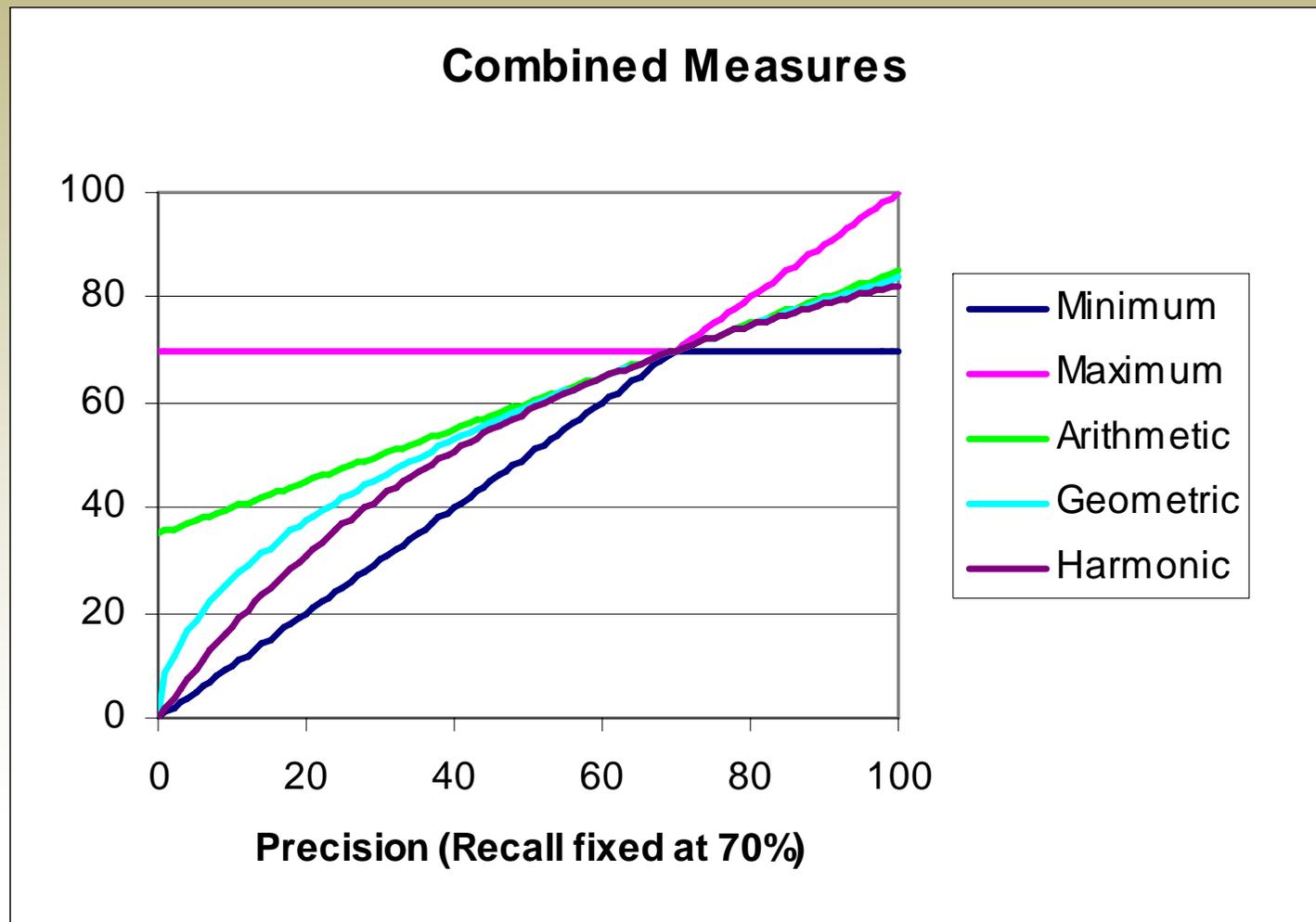
A combined measure: F

- Combined measure that assesses precision/recall tradeoff is **F measure** (weighted harmonic mean):

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

- People usually use balanced F_1 measure
 - i.e., with $\beta = 1$ or $\alpha = \frac{1}{2}$
- Harmonic mean is a conservative average
 - See CJ van Rijsbergen, *Information Retrieval*

F_1 and other averages

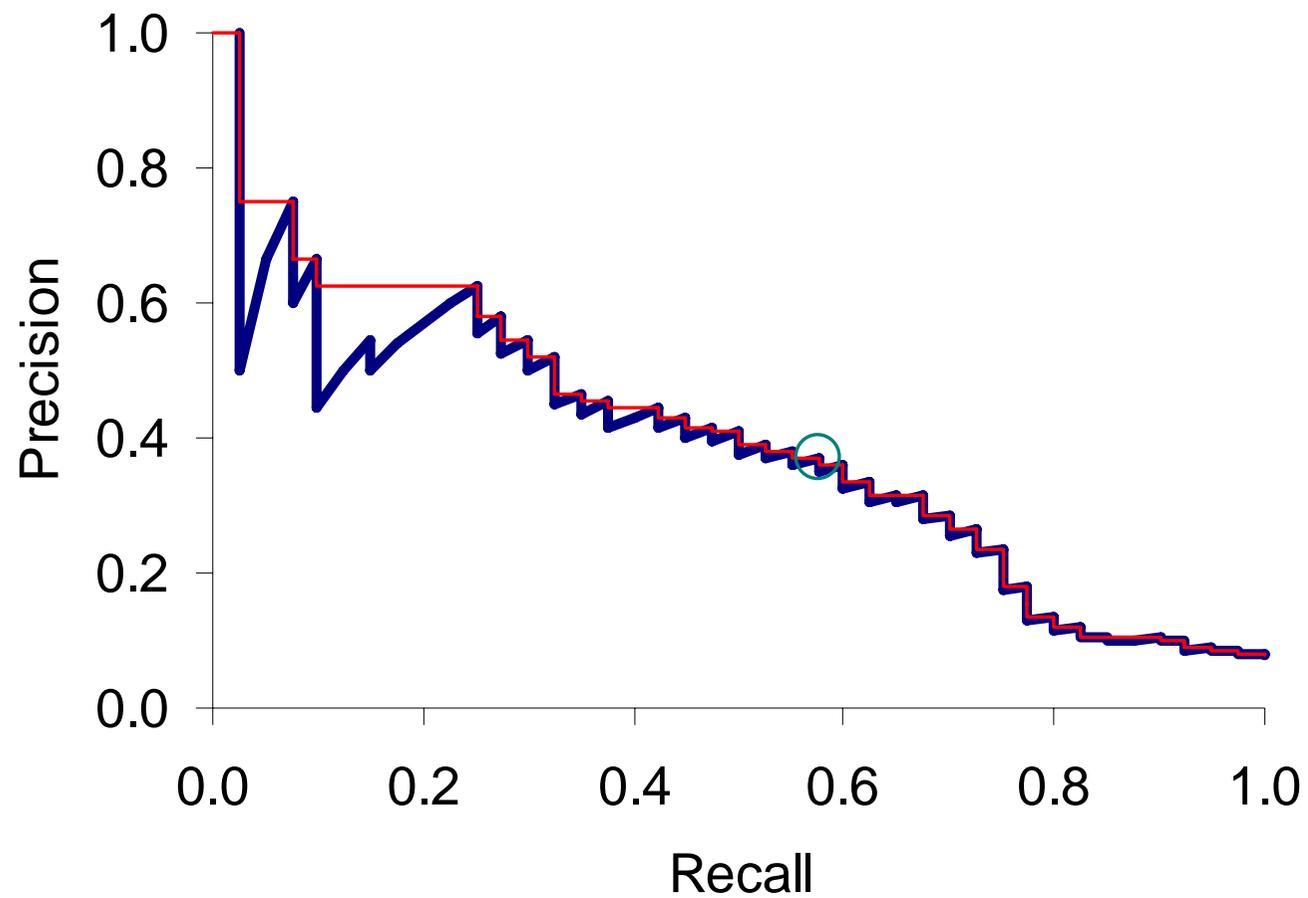




Evaluating ranked results

- Evaluation of ranked results:
 - The system can return any number of results
 - By taking various numbers of the top returned documents (levels of recall), the evaluator can produce a *precision-recall curve*

A precision-recall curve



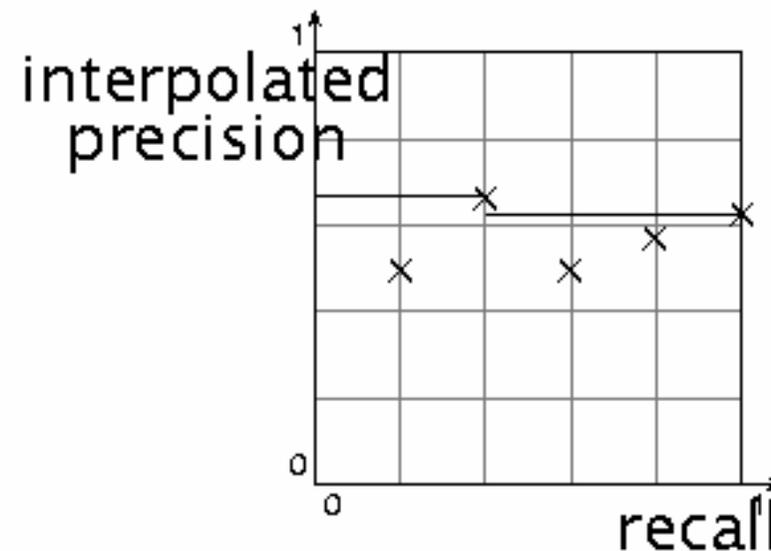
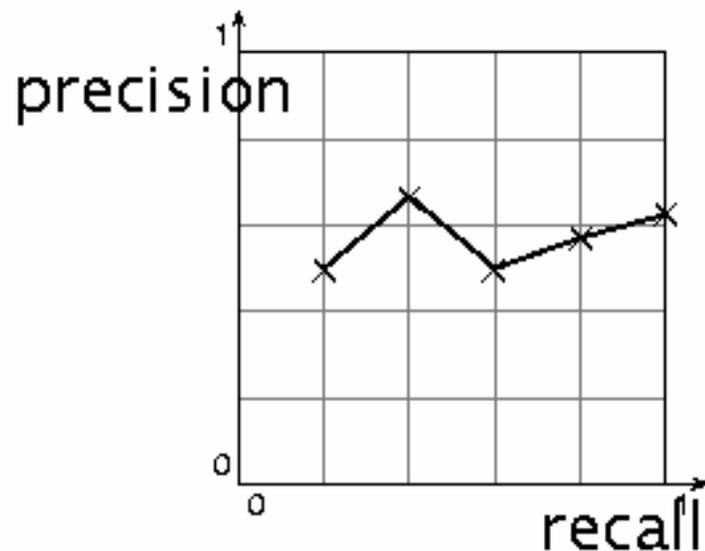


Averaging over queries

- A precision-recall graph for one query isn't a very sensible thing to look at
- You need to average performance over a whole bunch of queries.
- But there's a technical issue:
 - Precision-recall calculations place 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



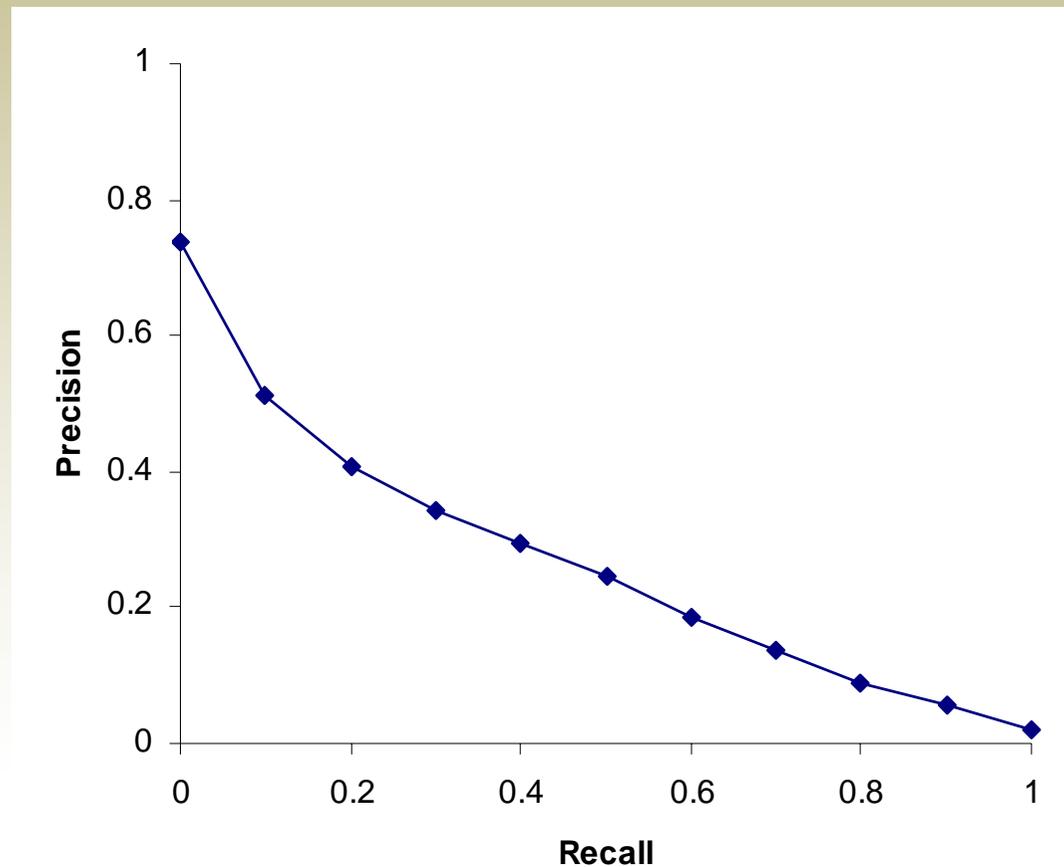


Evaluation

- Graphs are good, but people want summary measures!
 - Precision at fixed retrieval level
 - Precision-at- k : Precision of top k results
 - Perhaps appropriate for most of web search: all people want are good matches on the first one or two results pages
 - But: averages badly and has an arbitrary parameter of k
 - 11-point interpolated average precision
 - The standard measure in the early 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

Typical (good) 11 point precisions

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





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 ave.
 - Macro-averaging: each query counts equally
- R-precision
 - If have known (though perhaps incomplete) set of relevant documents of size Rel , then calculate precision of top Rel docs returned
 - Perfect system could score 1.0.



Variance

- For a test collection, it is usual that a system does crummily 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!