# Evaluation

David Kauchak

cs160

Fall 2009

*adapted from:*

http://www.stanford.edu/class/cs276/handouts/lecture8-evaluation.ppt

# Administrative

- How are things going?
- Slides
- Points

# Zipf’s law

# IR Evaluation

- For hw1, you examined 5 systems. How did you evaluate the systems/queries?
- What are important features for an IR system?
- How might we automatically evaluate the performance of a system? Compare two systems?
- What data might be useful?

# Measures for a search engine

- How fast does it index (how frequently can we update the index)
- How fast does it search
- How big is the index
- Expressiveness of query language
- UI
- Is it free?

- Quality of the search results

# Measuring user performance

- Who is the user we are trying to make happy and how can we measure this?
- Web search engine
  - user finds what they want and return to the engine
  - measure rate of return users
  - Financial drivers
- eCommerce site
  - user finds what they want and make a purchase
  - Is it the end-user, or the eCommerce site, whose happiness we measure?
  - Measure: time to purchase, or fraction of searchers who become buyers, revenue, profit, …
- Enterprise (company/govt/academic)
  - Care about “user productivity”
  - How much time do my users save when looking for information?

# Common IR evaluation

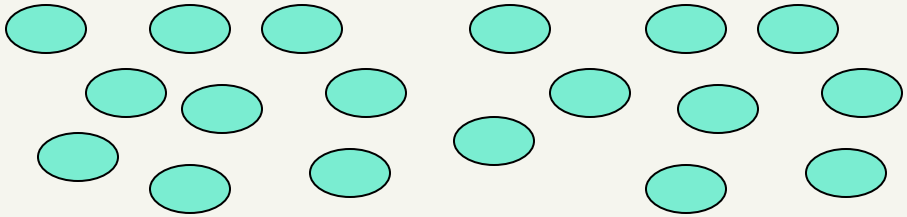
- Most common proxy: *relevance* of search results
- Relevance is assessed relative to the **information need** *not* the **query**
- Information need: *I'm looking for information on whether drinking red wine is more effective at reducing your risk of heart attacks than white wine*
- Query: ***wine red white heart attack effective***
- You evaluate whether the doc addresses the information need, **NOT** whether it has these words

# Data for evaluation

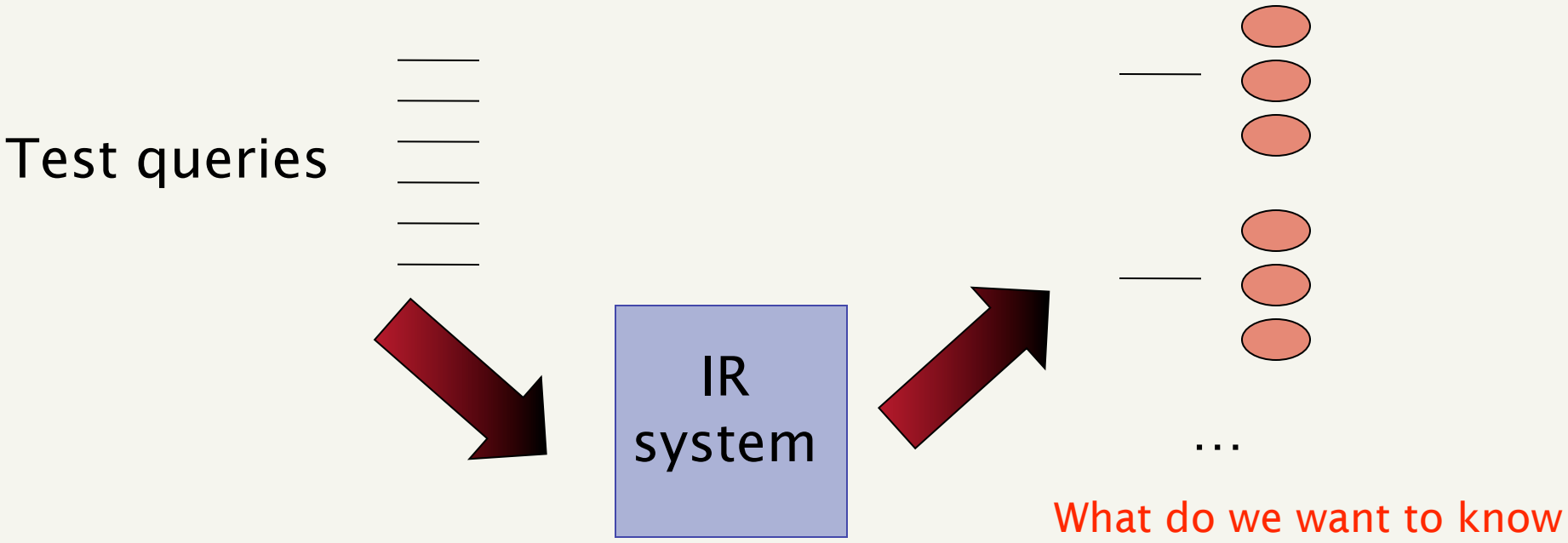
Test queries

IR system

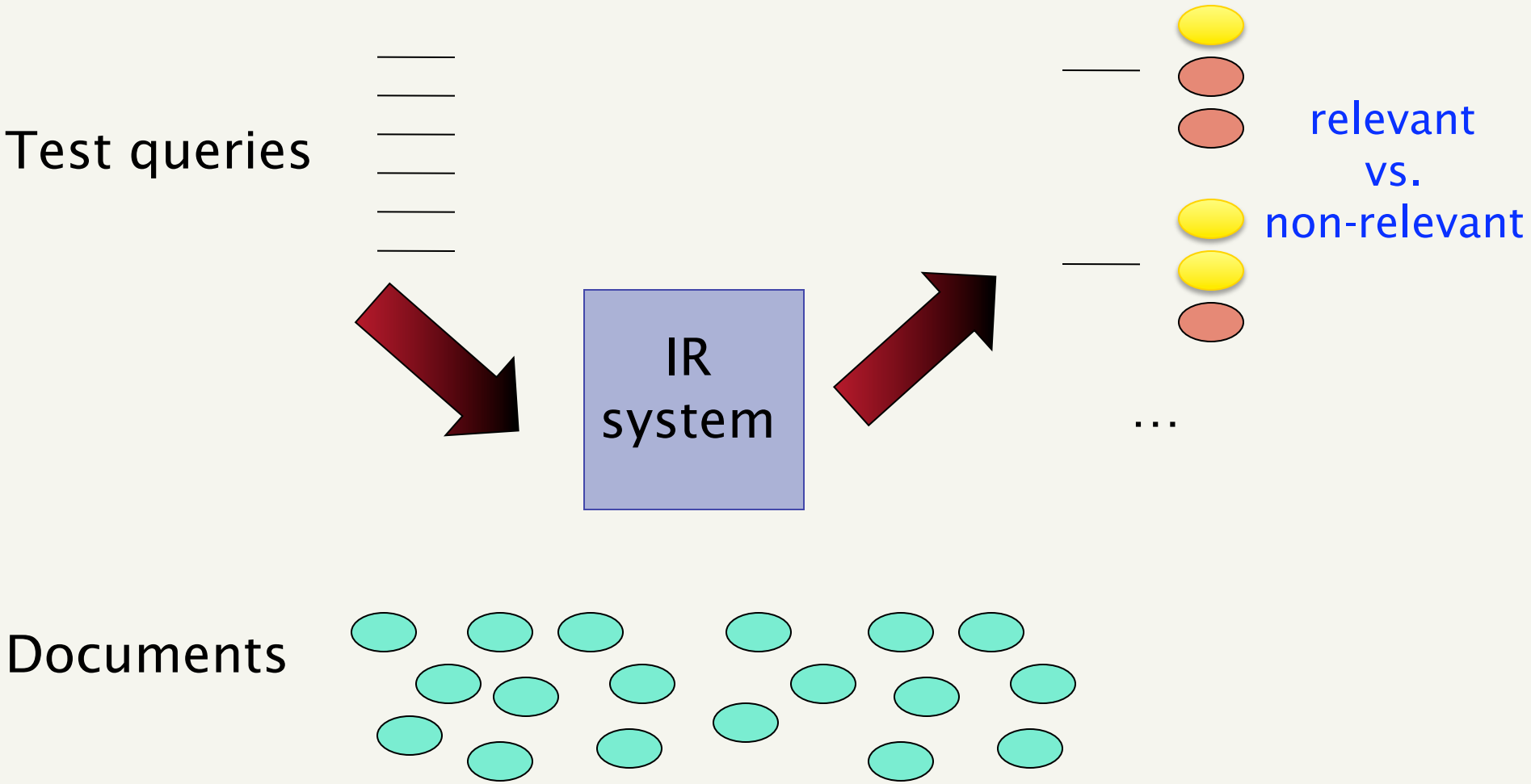
Documents



# Data for evaluation



# Data for evaluation



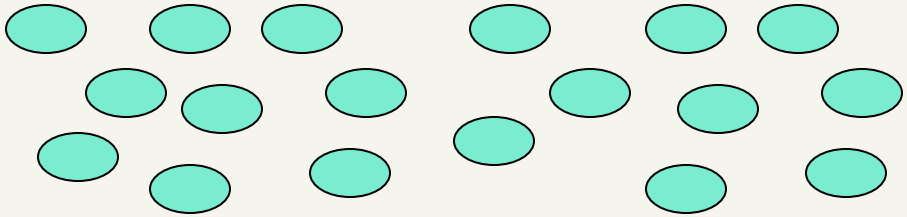
# Data for evaluation

Test queries

What if we want to test another system?
10 more systems?

IR system2

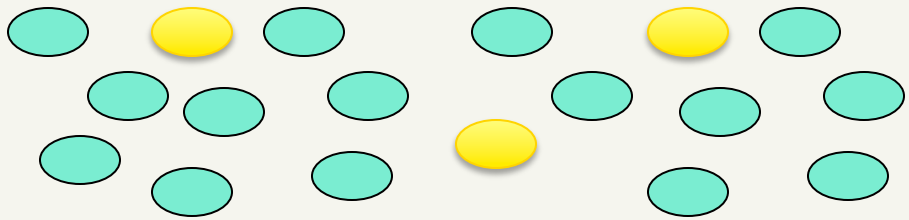
Documents



# Data for evaluation: option 1

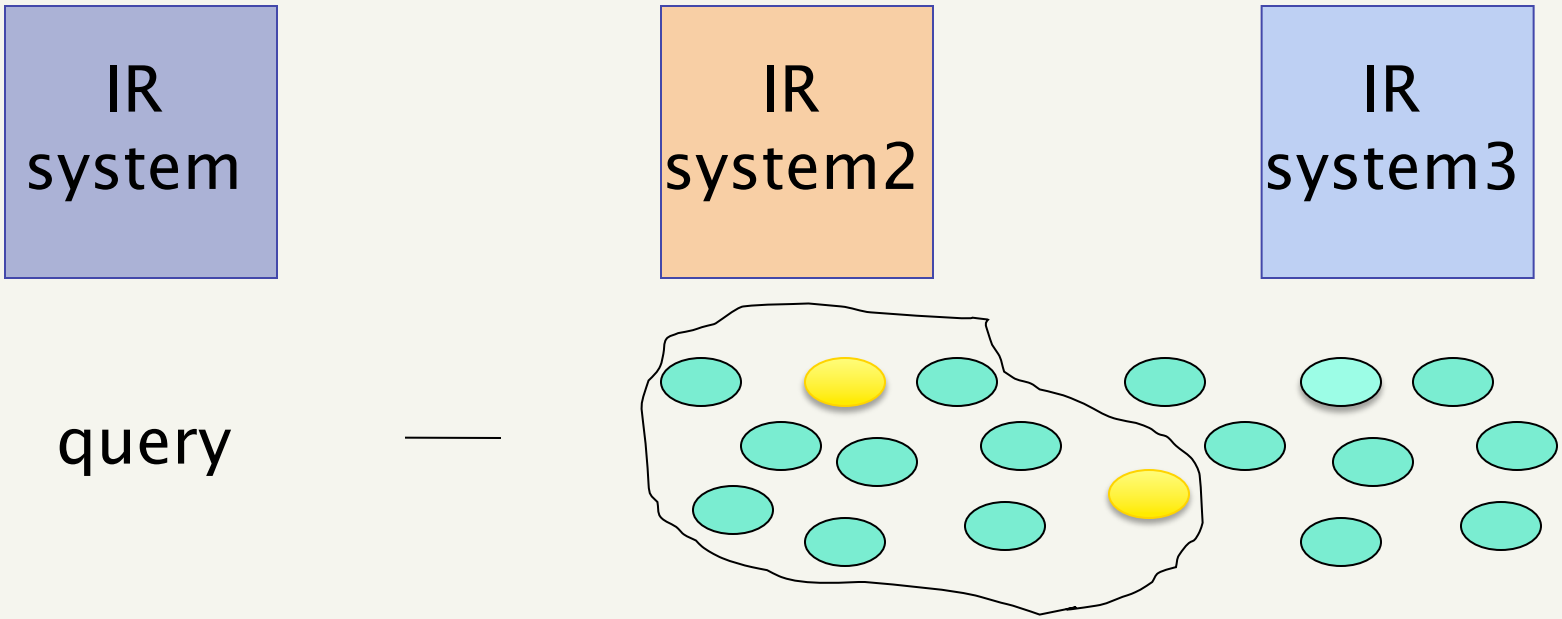
- For each query, identify ALL the relevant (and non-relevant) documents
- Given a new system, we know whether the results retrieved are relevant or not

query



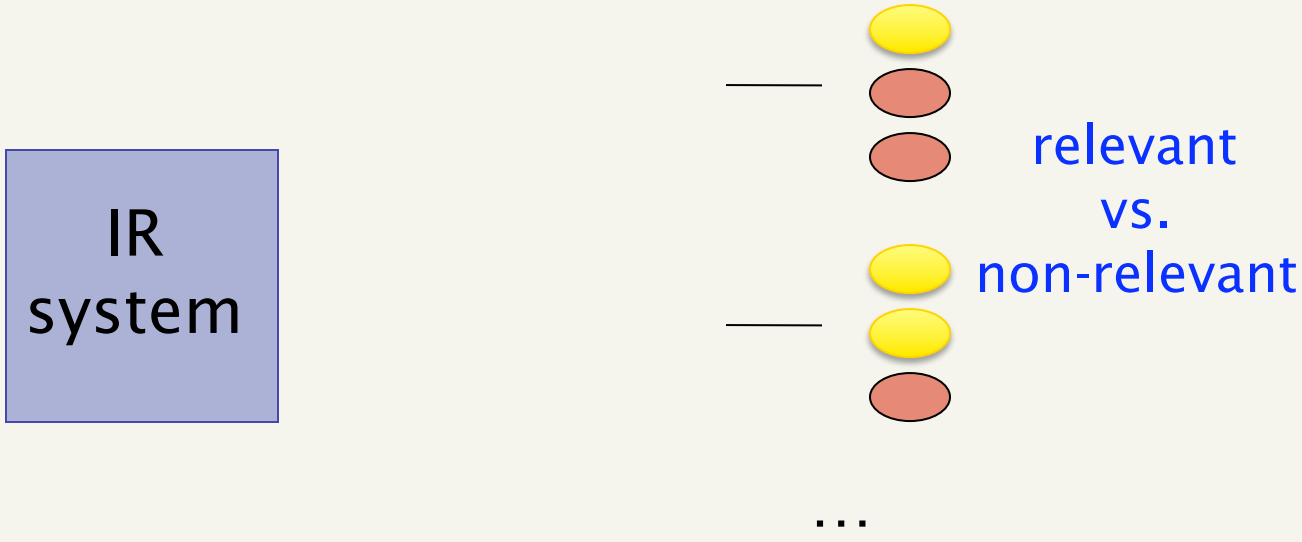
# Data for evaluation: option 2

- In many domains, finding ALL relevant documents is infeasible (think the web)
- Instead, evaluate a few sets of results for a few systems, and assume these are all the relevant documents



# How can we quantify the results?

- We want a numerical score to quantify how well our system is doing
- Allows us to compare systems
- To start with, let's just talk about boolean retrieval



# Accuracy?

- The search engine divides ALL of the documents into two sets:  relevant and nonrelevant
- The **accuracy** of a search engine is the proportion of these that it got right
- **Accuracy** is a commonly used evaluation measure in machine learning classification work

- Is this a good approach for IR?

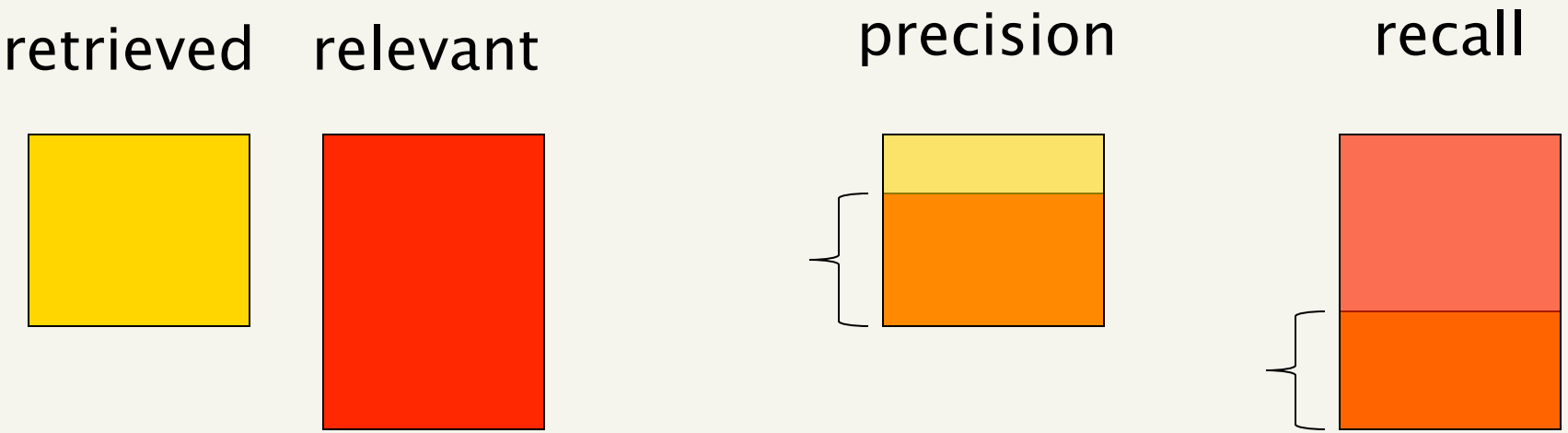
# Accuracy?

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


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

# 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)



# Precision/Recall tradeoff

- Often a trade-off  between better precision and better recall

- How can we increase recall?
  - Increase the number of documents retrieved (for example, return all documents)
- What impact will this likely have on precision?
  - Generally, retrieving more documents will result in a decrease in precision`

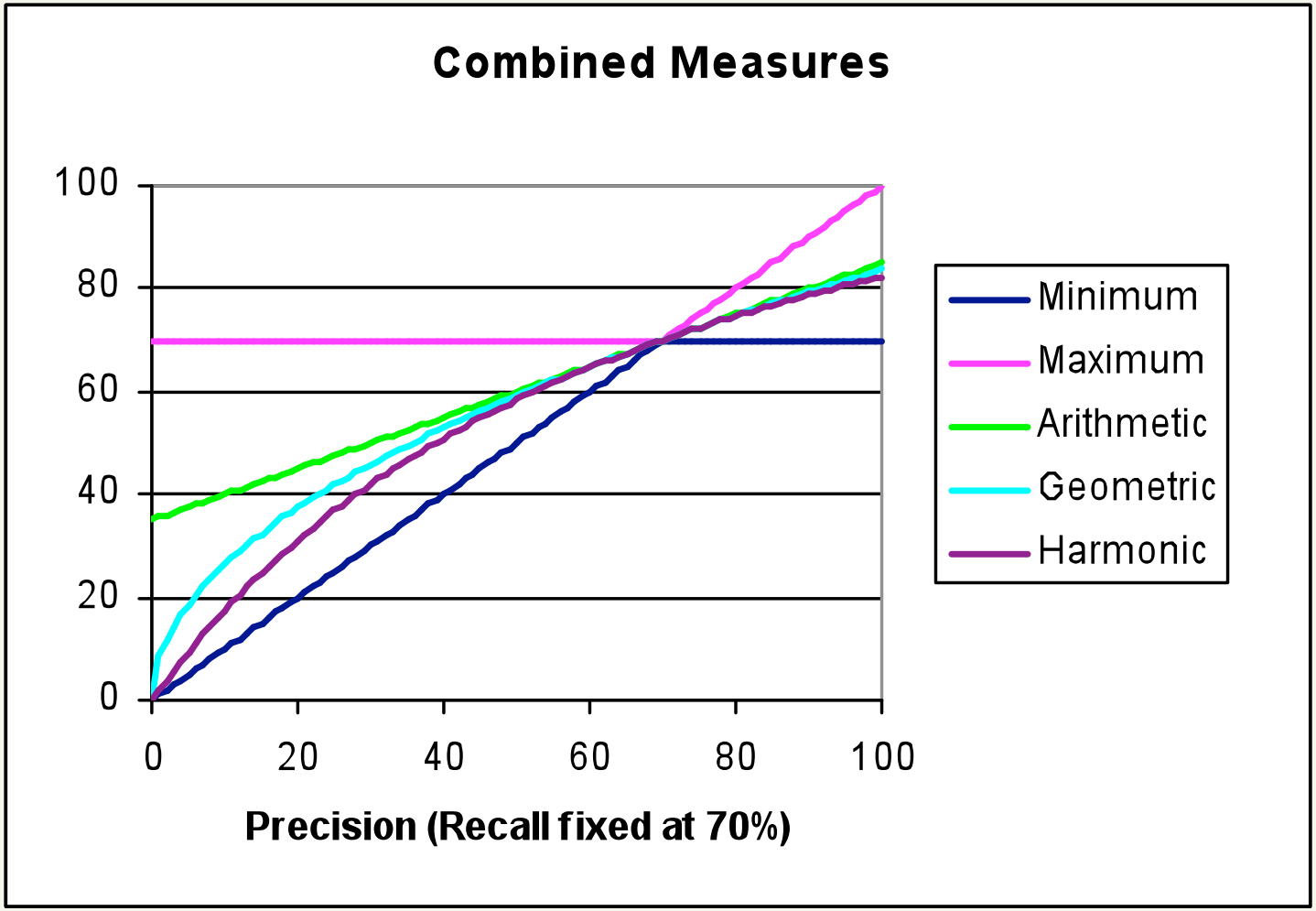
# A combined measure: *F*

- 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 $\alpha = ½$
- Harmonic mean is a conservative average

# $F_1$ and other averages

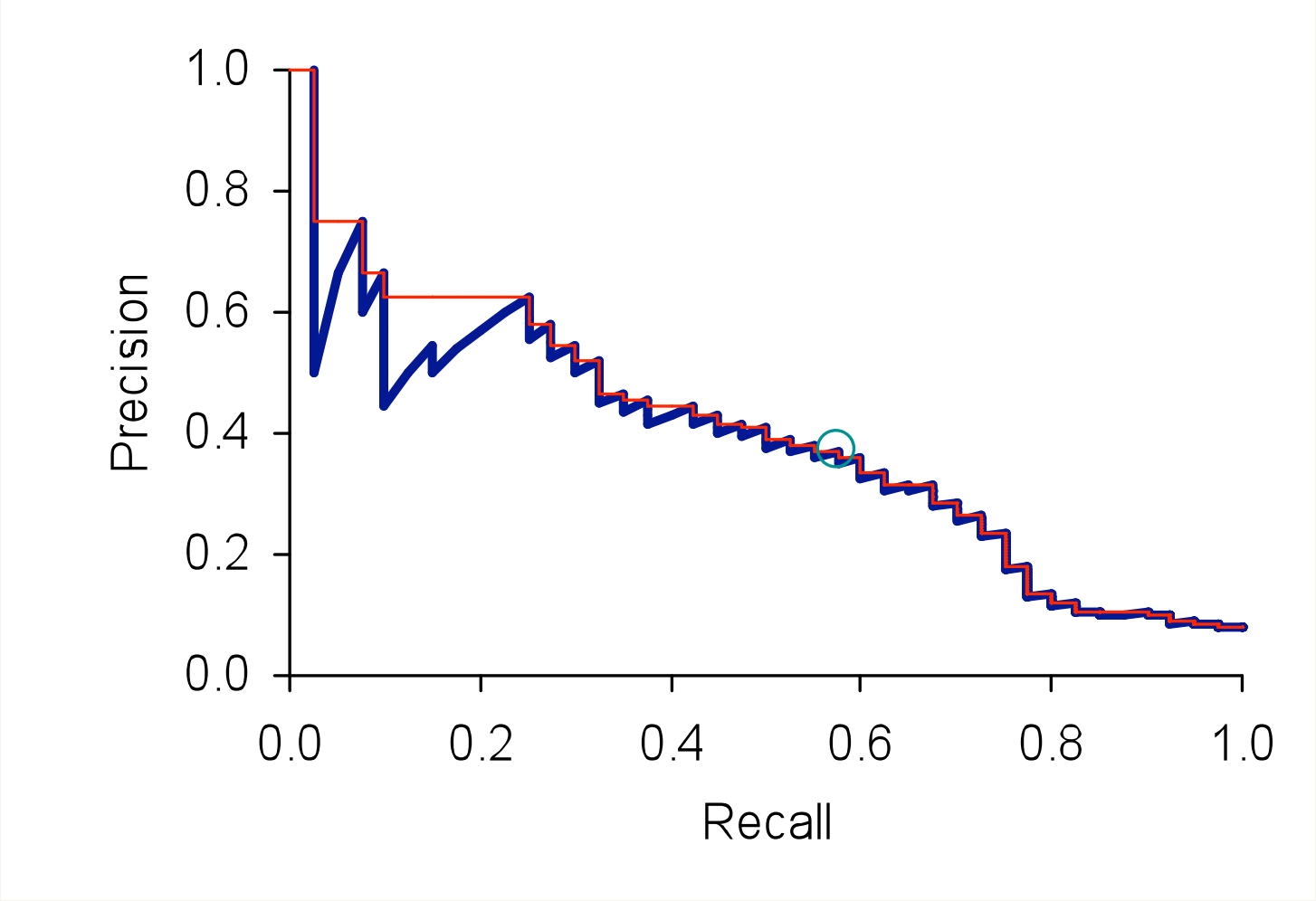


# Evaluating ranked results

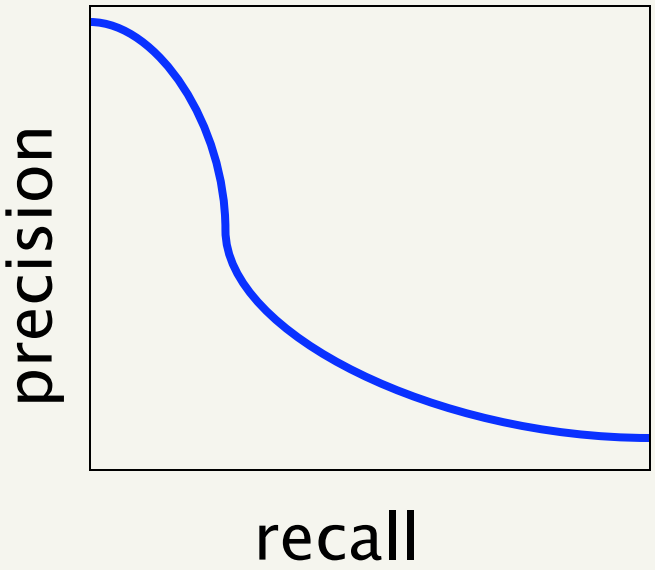
- Most IR systems are ranked systems
- We want to evaluate the systems based on their ranking of the documents
- What might we do?
- With a ranked system, we can look at the precision/ recall for the top K results

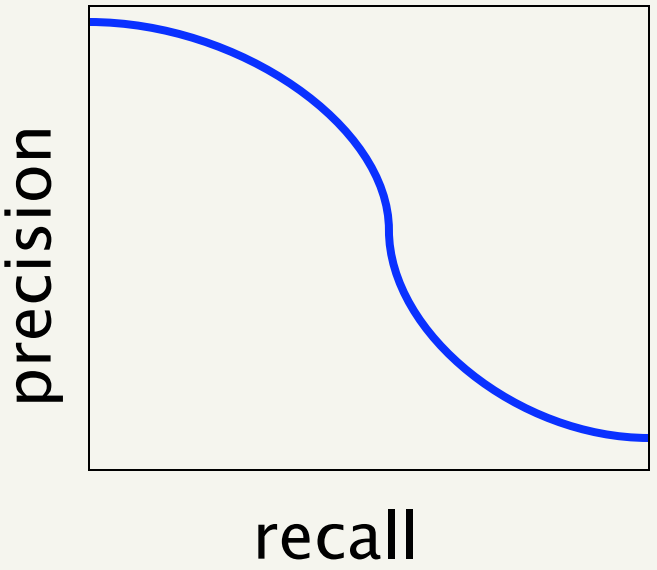
- Plotting this over K, gives us the precision-recall curve

# A precision-recall curve



# Which is system is better?



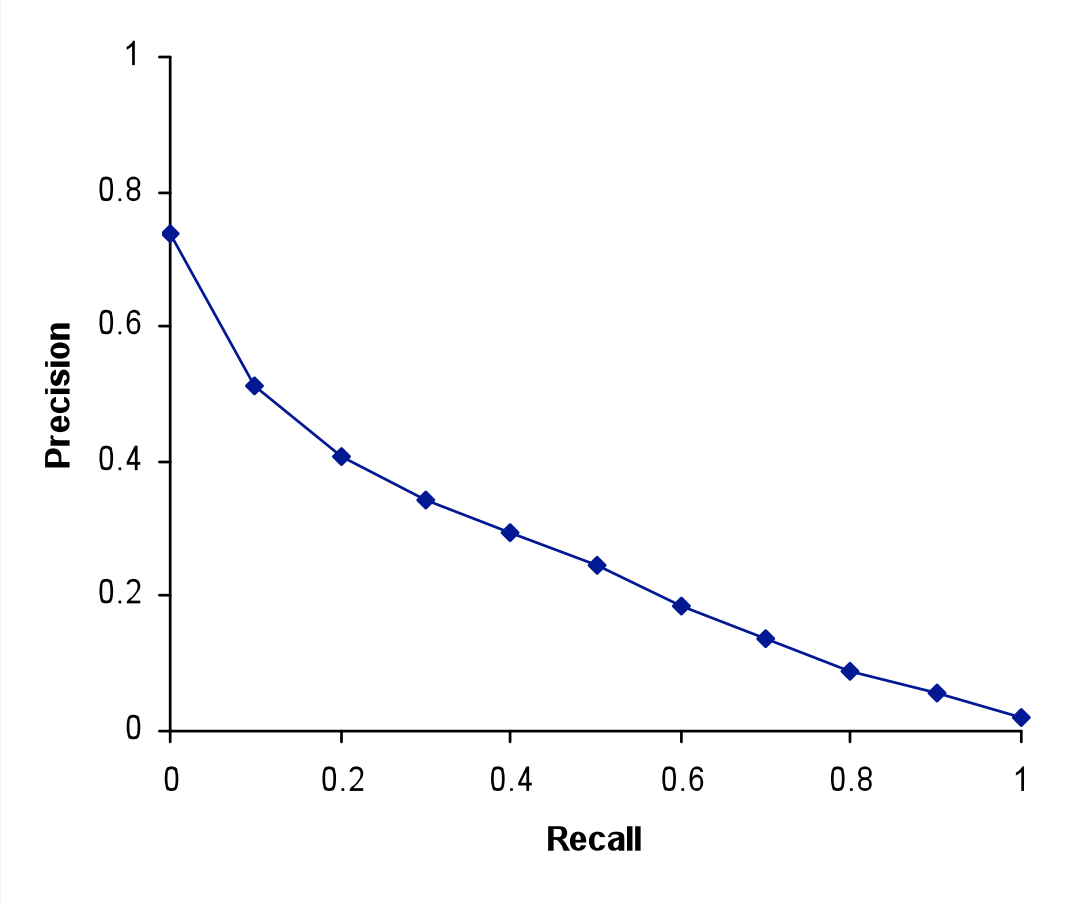



# Evaluation

- 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*
- Any way to capture more of the graph?
  - 11-point average precision
    - Take the precision at 11 levels of recall varying from 0 to 1 by tenths of the documents and average them
    - Evaluates performance at all recall levels (which may be good or bad)

# Typical (good) 11 point precisions

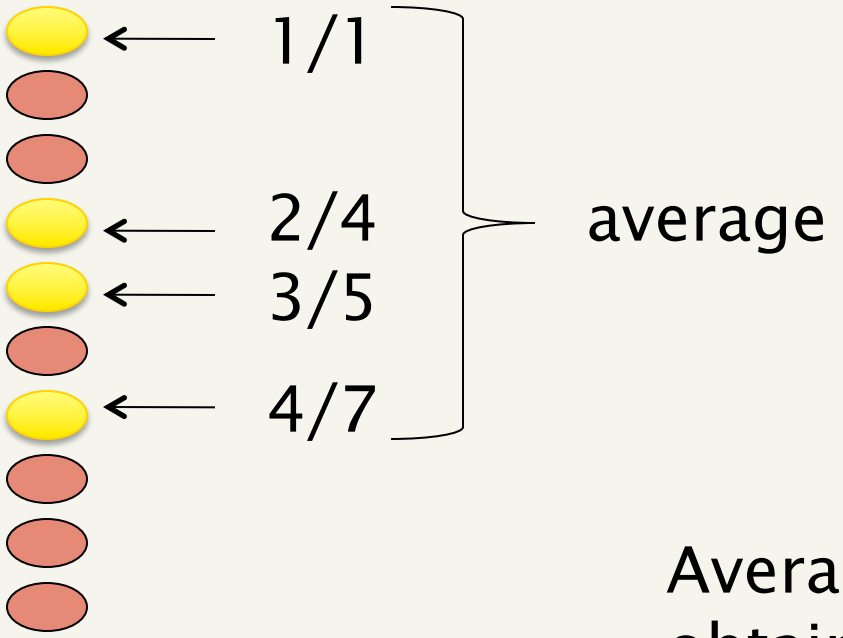
- SabIR/Cornell 8A1 11pt precision from TREC 8 (1999)



# 11 point is somewhat arbitrary…

- What are we really interested in?
  - How high up are the relevant results
- How might we measure this?
  - Average position in list
- Any issue with this?
  - Query dependent, i.e. if there are more relevant documents, will be higher (worse)
- Mean average precision (MAP)
  - Average of the precision value obtained for the top *k* documents**, each time** a relevant doc is retrieved

# MAP




Average of the precision value obtained for the top *k* documents, **each time** a relevant doc is retrieved

# Other issues: human evaluations

- Humans are not perfect or consistent
- Often want multiple people to evaluate the results

| Number of docs | Judge 1 | Judge 2 |
|---|---|---|
| 300 | Relevant | Relevant |
| 70 | Nonrelevant | Nonrelevant |
| 20 | Relevant | Nonrelevant |
| 10 | Nonrelevant | relevant |

# Multiple human labelers

- Can we trust the data?
- How do we use multiple judges?

| Number of docs | Judge 1 | Judge 2 |
|---|---|---|
| 300 | Relevant | Relevant |
| 70 | Nonrelevant | Nonrelevant |
| 20 | Relevant | Nonrelevant |
| 10 | Nonrelevant | relevant |

| Number of docs | Judge 1 | Judge 2 |
|---|---|---|
| 100 | Relevant | Relevant |
| 30 | Nonrelevant | Nonrelevant |
| 200 | Relevant | Nonrelevant |
| 70 | Nonrelevant | relevant |

# Measuring inter-judge agreement

Is there any problem with this?

370/400 = 92.5%

| Number of docs | Judge 1 | Judge 2 |
|---|---|---|
| 300 | Relevant | Relevant |
| 70 | Nonrelevant | Nonrelevant |
| 20 | Relevant | Nonrelevant |
| 10 | Nonrelevant | relevant |

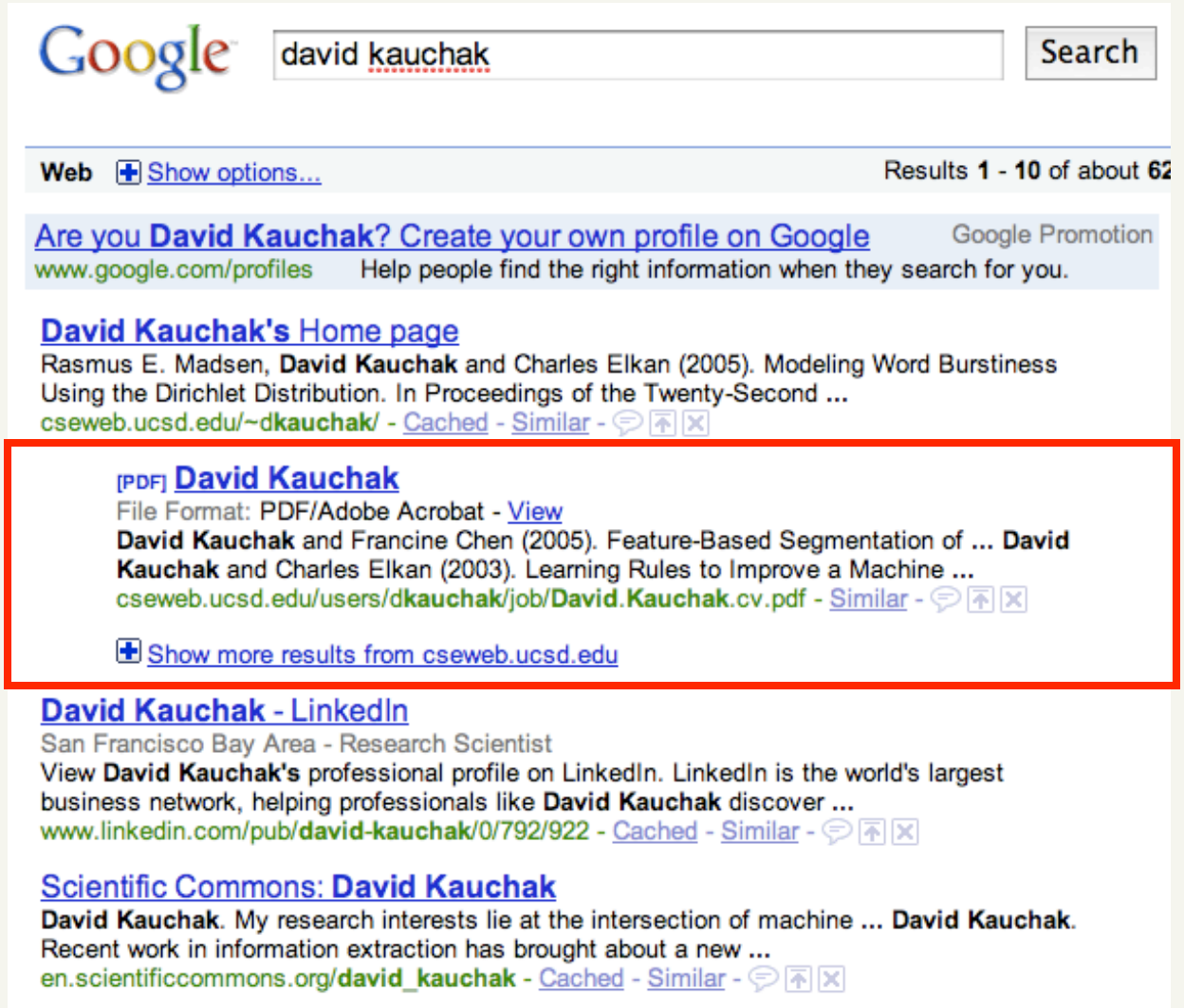
130/400 = 32.5%

| Number of docs | Judge 1 | Judge 2 |
|---|---|---|
| 100 | Relevant | Relevant |
| 30 | Nonrelevant | Nonrelevant |
| 200 | Relevant | Nonrelevant |
| 70 | Nonrelevant | relevant |

# Measuring inter-judge (dis)agreement

- Kappa measure
  - Agreement measure among judges
  - Designed for categorical judgments
  - Corrects for chance agreement
- Kappa = [ P(A) – P(E) ] / [ 1 – P(E) ]
- P(A) – proportion of time judges agree
- P(E) – what agreement would be by chance
- Kappa = 0 for chance agreement, 1 for total agreement
- Kappa above 0.7 is usually considered good enough

# Other issues: pure relevance




Why does Google do this?

# Other issues: 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
- Measuring marginal relevance can be challenging, but search engines still attempt to tackle the problem

# Evaluation at large search engines

- Search engines have test collections of queries and hand-ranked results
- 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

# A/B Testing

- Google wants to test the variants below to see what the impact of the two variants is
- How can they do it?

# A/B testing

- 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

# Guest speaker today

- Ron Kohavi
- http://videolectures.net/kdd07_kohavi_pctce/