Building Test Collections

(without running a community evaluation)

Ian Soboroff
NIST

Schedule

9-9:30	introductions
9:30-10:30	How to build a test collection
(10:30-11 break)	
11-12:15	How to know if a collection is any good
(12:15-13:30 lunch)	
13:30-15:30	Questions, extras, set up group activity
(3:30-4 break)	
4-5:30	continue group activity

Goals and anti-goals

- Talk about how to build a test collection.
- (Without having your own TREC track.)
- Test collections for user tasks relating to accessing information. (Not generic datasets.)
- A gentle dive into the critical questions.
- A good bibliography to explore afterwards.
- Talk through actual test collection building issues together.

NOT...

- to teach the basics of information retrieval,
- or about IR evaluation or how to use test collections,
- or how to start your own evaluation conference series.

Tutorial in One Slide

- Determine the *task*.
- Identify a *document collection*.
- Build *topics*.
- Make *relevance judgments*.
- Conduct experiments to *measure* the collection.

https://isoboroff.github.io/Test-Colls-Tutorial/

The task

- What is the user trying to do?
- Central to test collection construction.
- Abstracted from the real world to something measurable.
- Drives...
  - the choice of a document collection
  - topic development
  - relevance assessment
  - measures

TREC ad hoc task

- User task: user is searching for any and all information about a subject. They will prepare a report describing all information found.
- Abstracted task
  - single query search with ranked retrieval.
  - applies to nearly any document collection.
  - relevance is defined *minimally* and *independently*.
  - recall is important but high precision is desired.
  - implies AP as a primary measure.

Task abstraction

In the real world...

- user has context
- searches to accomplish a larger goal
- searches many times
- reads a few documents, jumps around.
- consumes information in a variety of ways
- goals change over time

In the abstract world...

- user has no context
- searches occur in isolation
- searches once
- reads linearly through the ranked list.
- reading counts for relevance
- goal is abstract

Abstraction

- Really, these are all criticisms that apply to the Cranfield experiments from the 1960s.
- They are not inherent to test collections.
- The "Cranfield paradigm" is extensible in lots of ways.
  - reading pattern / "interaction" model
  - relevance definition
  - context
  - goal

    ...
- However, many things are hard or impossible to implement in a test collection given typical resources.
  - relevance feedback
  - novelty

In-class task exercise

- User task: re-finding on the web.
- The person is looking for a specific web page that they found two weeks ago, but they don't remember exactly what it was called or where it was.
- Abstract this task:
  - operationalize the task,
  - define relevance,
  - define measures.

Exercise 2

- User task: tweet filtering.
- The person follows thousands of people but only wants to see really useful tweets, as quickly as possible.
- Abstract this task:
  - operationalize the task,
  - define relevance,
  - define measures.

Exercise 3

- User task: background citation search.
- Given a paper, suggest the best introductory materials that the person should read in order to understand that paper.
- Abstract this task:
  - operationalize the task,
  - define relevance,
  - define measures.

Does the task make sense?

- It's easy to imagine tasks, especially if we begin from an available document set.
- ... or an older task we know how to evaluate.
- Is the TREC ad hoc task sensible in any document collection? Can you think of one where it isn't?
- Even if the task is sensible, is it what users want to do with that data?
- We want improvement on the abstract task to predict improvement in actual systems with real users and tasks.

Document collections

- Now that we have a task, let's identify a document collection.
- Some tasks imply certain kinds of documents.
- But others can apply with any sort of information.
- The documents themselves affect how systems can search, so systems will perform differently on the same task with different documents. (Seems obvious)
  - what does it mean for a task to be "intuitive"?
  - to be "real"? "natural"?
  - the toolmaker's paradox: tool users learn to use the tool in the way that works for them.

Don’t he look natural?

- Some collections are *opportunistic*.
  - Opportunity samples (my email)
  - Easily available data (the university’s website)
- Some collections are *constructed*.
  - Tweets containing a #hashtag.
  - Web pages retrieved in response to these queries.
- Others aim to be *naturalistic*.
  - A very large web crawl.
  - A year’s worth of news.
  - All tweets for three months.

What are the implications of these choices?

Can you share?

- Ask yourself if the document collection you are working with is something you can distribute to other people.
- Reproducibility is becoming an important factor for publication.
- Just a few of the issues:
  - copyrighted? public? fair use?
  - terms of service, licensing restrictions?
  - private, proprietary, personally identifiable information?
- It is important for others to be able to reproduce your work.
- It is also important not to get sued.
- It is also important not to annoy your data source.

Does the collection have to be static?

- Static document collections are simpler methodologically.
- I can easily compare two systems on one collection.
- What if the collection changes in between?
- How can we use the live web usefully as a document set in a test collection?
- Or, pointers to web pages that might change?
- (Soboroff, SIGIR 2006, "Dynamic test collections" gives some ideas, and there are lots of others...)

Topics

- Articulation of user's information need.
- Not a query:
  - Context and interpretation of the need.
  - Allows experiments on query formulation.
  - Documentation.
- An experimental observation.
  - Variance due to topics is greater than variance due to systems.
  - Averaged, or a rich source for analysis.
  - Can embody experimental conditions.

Building topics

- (Some) Methods used in TREC
  - Collection exploration
  - Log-driven
- Observation of real users

Collection exploration

- Allow the assessor to explore the collection.
  - Search, browse, etc.
- Assessor invents the information need.
  - Unnatural in that topics have an artificial context.
  - The assessor is the real user, however.

Basic exploration process

- Assessor is given a search interface to the collection.
  - Sometimes allow browsing within web collections.
  - Result documents are sandboxed.
- Use a single search.
  - Really, just use a few searches to arrive at a “final” query.
- Judge the top 25 documents for relevance.
- If none are relevant, or 20+ are relevant, discard the topic.
- Record final query, topic title, description, and additional comments.
- (Topic selection process happens.)
- Assessors compose final topic statements.

Log-driven development

- Start with a query log.
- (indirect observations of user behaviors.)
- Assessor extrapolates those observations to a topic.
  - Simplistic: "backfit" a topic to a single query.
  - What problems could arise there?
  - How could they be solved?
- More "natural" because the topic starts from a "real query".
- Interpretation process could involve exploration.

Observational

- Observe users performing their own searches.
- These searches may arise from user's normal tasks, or be prompted.

Bias and balance

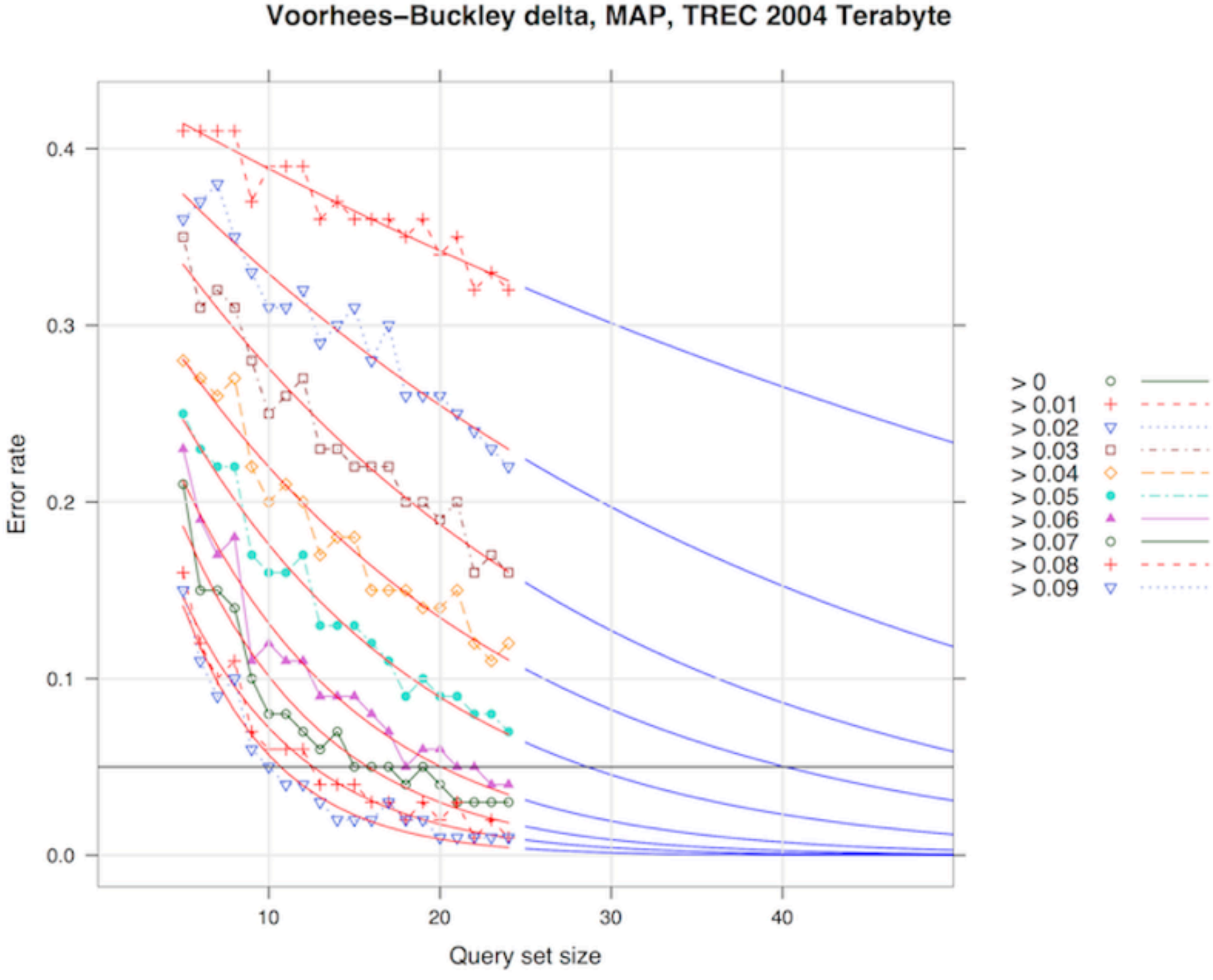
- With collection exploration, the topic set reflects the abilities of the assessors with the tools provided.
- Query logs also exhibit the tools used to create them.
- Topics can be intrinsically hard or easy for many reasons.
  - Some are system-independent, but most are not.
- We strive to build a balanced topic set, because systems exhibit high variance on individual topics.

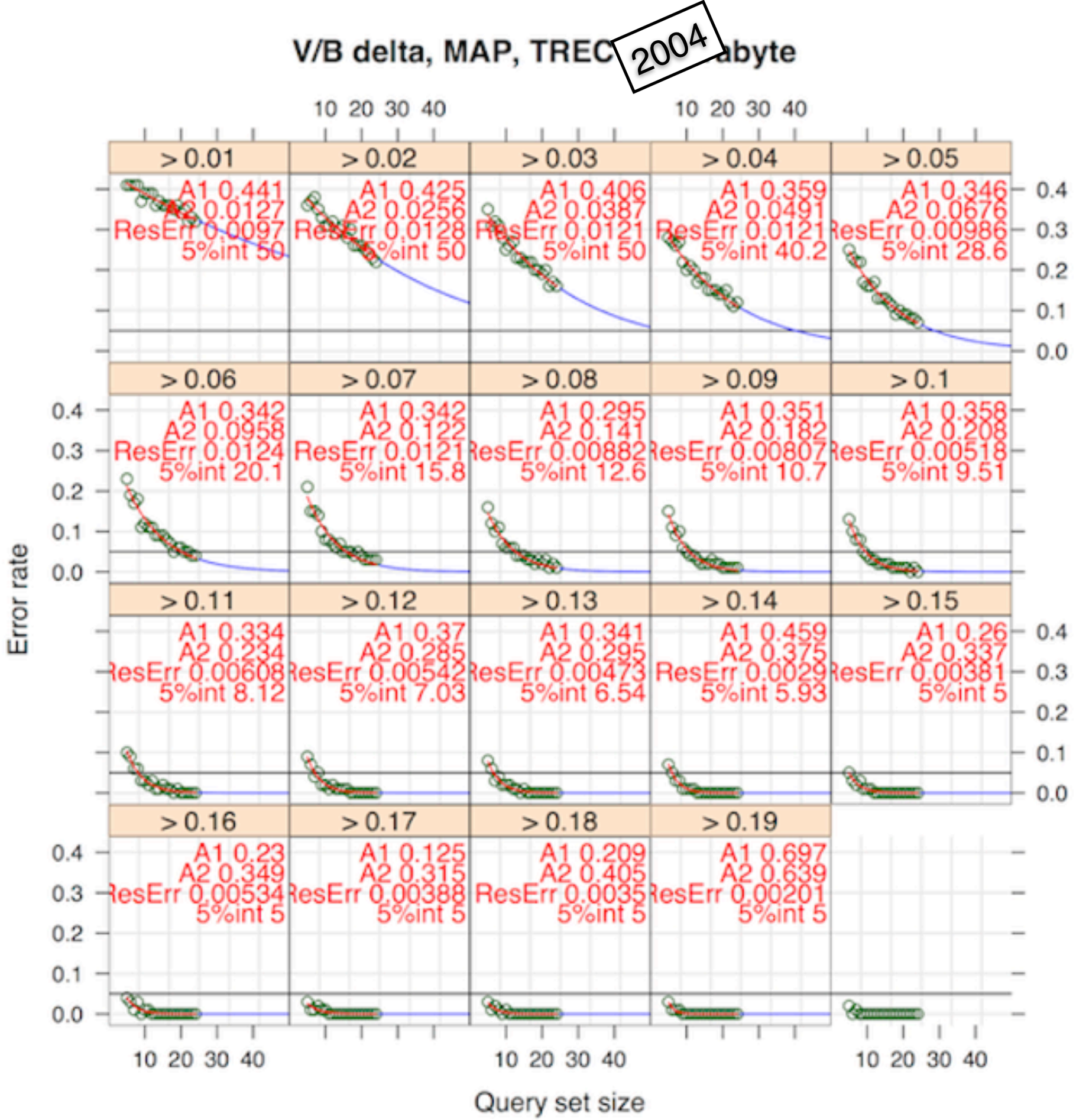
How many topics do you need?

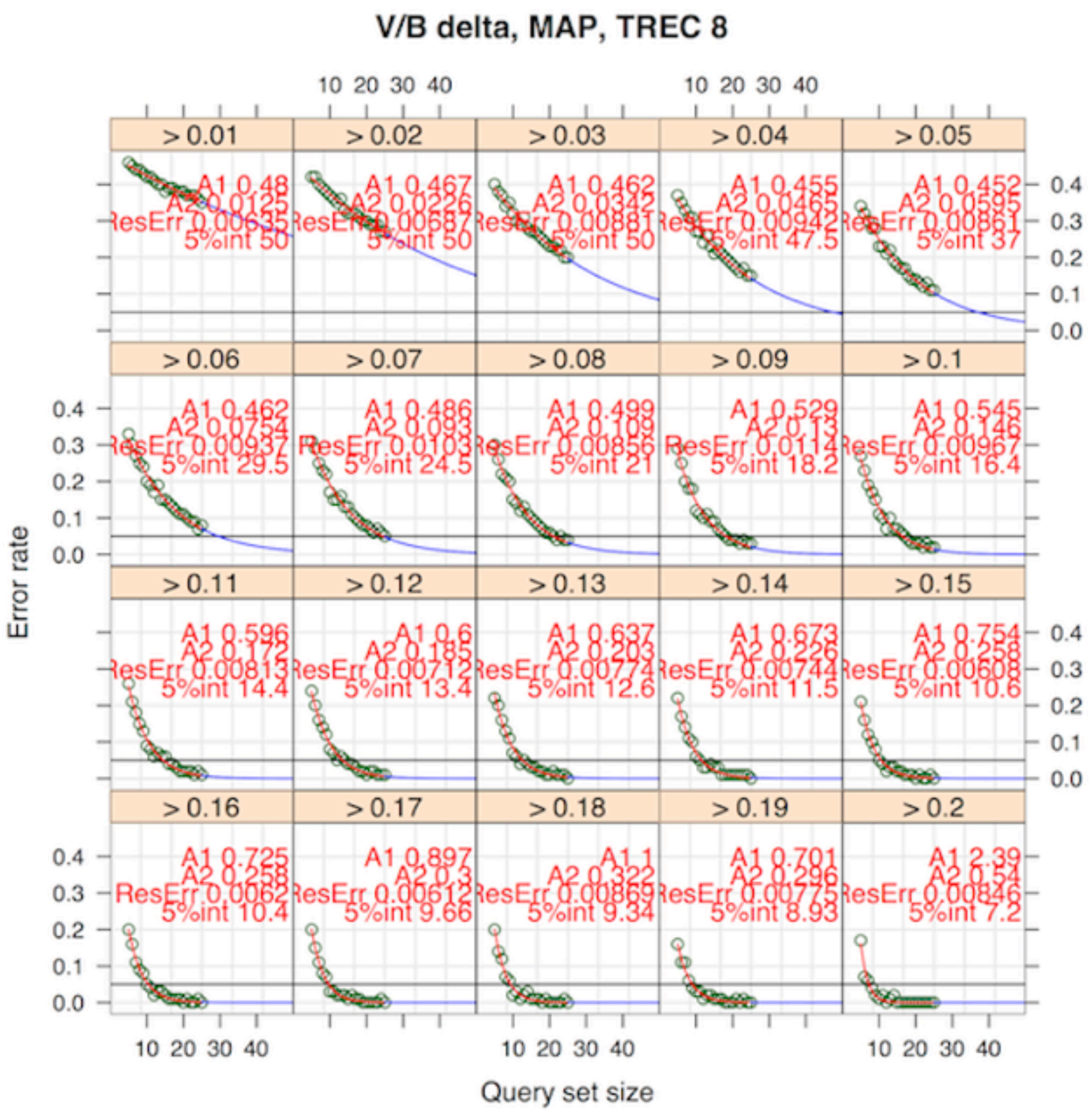
- Having more topics ...
  - smooths out variability across topics.
  - increases the discriminative power of the collection.
  - makes room for having subsets of topics for different experimental conditions.
  - costs more.

Stability

- Buckley and Voorhees (SIGIR 2000)
- Ingredients: a test collection, a set of system runs.
- Choose two random disjoint subsets of the topics.
- Rank the runs according to each subset using some measure.
- Count the number of times systems swap their position in the ranking.
- Repeat.
- Findings:
  - High probability of a swap => collection is unstable.
  - Stability increases with more topics.
  - Some measures (P@10) are less stable than others (MAP).
  - Differences of less than 0.05 are not usually detectable.

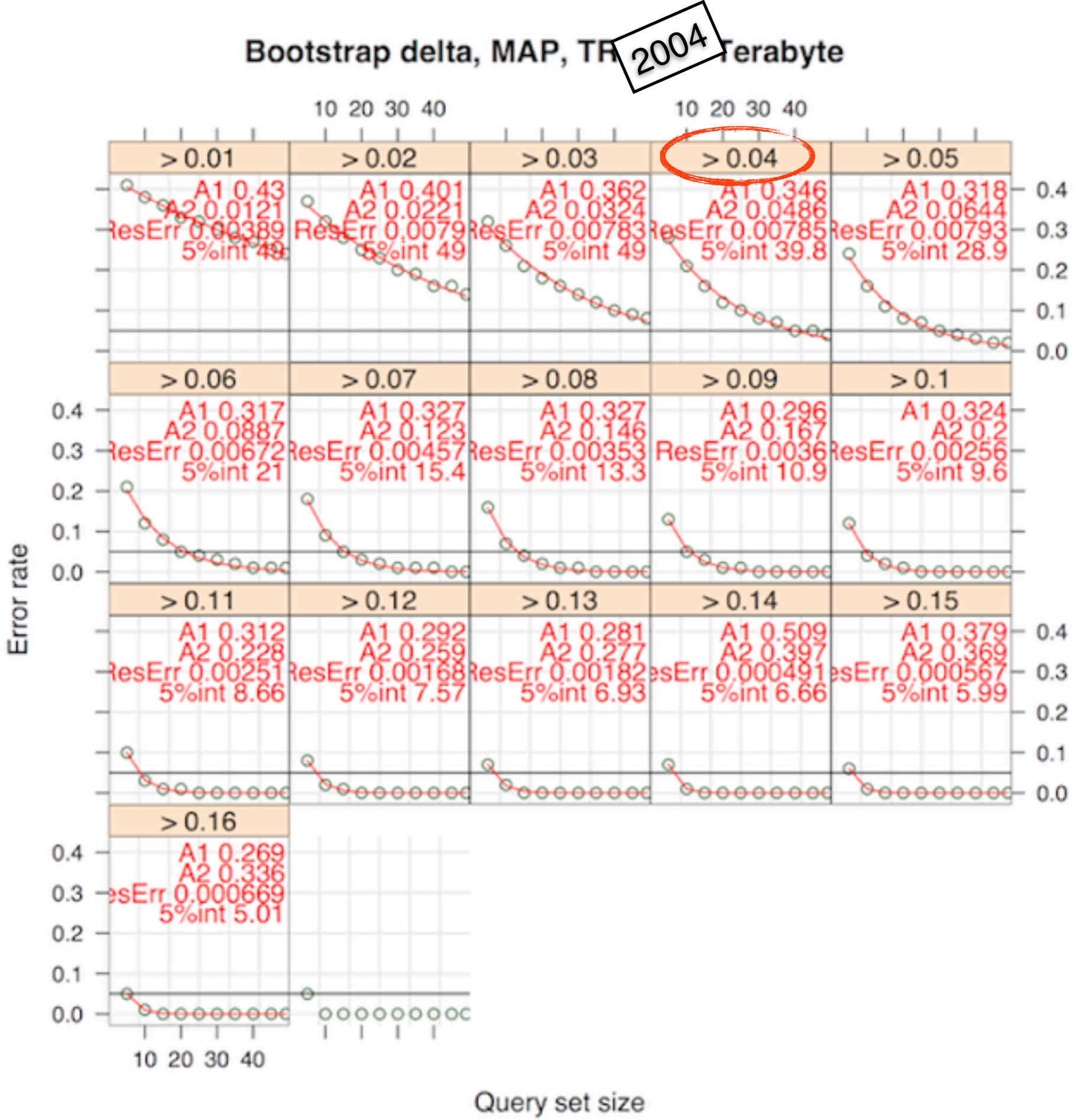

Voorhees–Buckley delta, MAP, TREC 2004 Terabyte
Error rate
0.0
0.1
0.2
0.3
0.4
10
20
30
40
Query set size
> 0
> 0.01
> 0.02
> 0.03
> 0.04
> 0.05
> 0.06
> 0.07
> 0.08
> 0.09

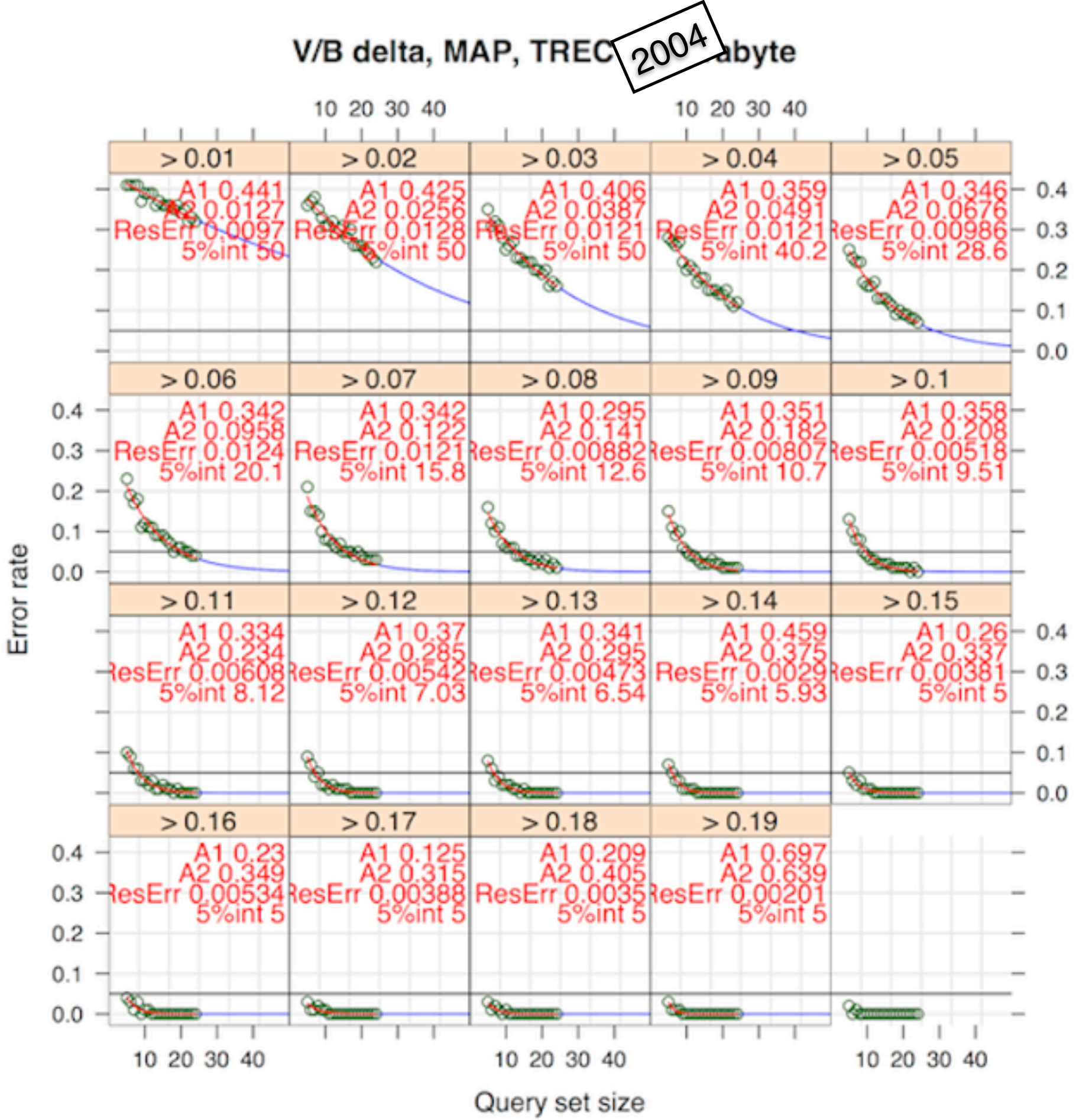

V/B delta, MAP, TREC 2004 Terabyte
> 0.01
A1 0.441
A2 0.0127
ResErr 0.0097
5%int 50
> 0.02
A1 0.425
A2 0.0256
ResErr 0.0128
5%int 50
> 0.03
A1 0.406
A2 0.0387
ResErr 0.0121
5%int 50
> 0.04
A1 0.359
A2 0.0491
ResErr 0.0121
5%int 40.2
> 0.05
A1 0.346
A2 0.0676
ResErr 0.00986
5%int 28.6
> 0.06
A1 0.342
A2 0.0958
ResErr 0.0124
5%int 20.1
> 0.07
A1 0.342
A2 0.122
ResErr 0.0121
5%int 15.8
> 0.08
A1 0.295
A2 0.141
ResErr 0.00882
5%int 12.6
> 0.09
A1 0.351
A2 0.182
ResErr 0.00807
5%int 10.7
> 0.1
A1 0.358
A2 0.208
ResErr 0.00518
5%int 9.51
> 0.11
A1 0.334
A2 0.234
ResErr 0.00608
5%int 8.12
> 0.12
A1 0.37
A2 0.285
ResErr 0.00542
5%int 7.03
> 0.13
A1 0.341
A2 0.295
ResErr 0.00473
5%int 6.54
> 0.14
A1 0.459
A2 0.375
ResErr 0.0029
5%int 5.93
> 0.15
A1 0.26
A2 0.337
ResErr 0.00381
5%int 5
> 0.16
A1 0.23
A2 0.349
ResErr 0.00534
5%int 5
> 0.17
A1 0.125
A2 0.315
ResErr 0.00388
5%int 5
> 0.18
A1 0.209
A2 0.405
ResErr 0.0035
5%int 5
> 0.19
A1 0.697
A2 0.639
ResErr 0.00201
5%int 5
Error rate
Query set size
10 20 30 40
0.0 0.1 0.2 0.3 0.4


V/B delta, MAP, TREC 8
> 0.01
A1 0.48
A2 0.0125
ResErr 0.00635
5%int 50
> 0.02
A1 0.467
A2 0.0226
ResErr 0.00687
5%int 50
> 0.03
A1 0.462
A2 0.0342
ResErr 0.00881
5%int 50
> 0.04
A1 0.455
A2 0.0465
ResErr 0.00942
5%int 47.5
> 0.05
A1 0.452
A2 0.0595
ResErr 0.00861
5%int 37
> 0.06
A1 0.462
A2 0.0754
ResErr 0.00937
5%int 29.5
> 0.07
A1 0.486
A2 0.093
ResErr 0.0103
5%int 24.5
> 0.08
A1 0.499
A2 0.109
ResErr 0.00856
5%int 21
> 0.09
A1 0.529
A2 0.13
ResErr 0.0114
5%int 18.2
> 0.1
A1 0.545
A2 0.146
ResErr 0.00967
5%int 16.4
> 0.11
A1 0.596
A2 0.172
ResErr 0.00813
5%int 14.4
> 0.12
A1 0.6
A2 0.185
ResErr 0.00712
5%int 13.4
> 0.13
A1 0.637
A2 0.203
ResErr 0.00774
5%int 12.6
> 0.14
A1 0.673
A2 0.226
ResErr 0.00744
5%int 11.5
> 0.15
A1 0.754
A2 0.258
ResErr 0.00608
5%int 10.6
> 0.16
A1 0.725
A2 0.258
ResErr 0.0062
5%int 10.4
> 0.17
A1 0.897
A2 0.3
ResErr 0.00612
5%int 9.66
> 0.18
A1 1.1
A2 0.322
ResErr 0.00869
5%int 9.34
> 0.19
A1 0.701
A2 0.296
ResErr 0.00775
5%int 8.93
> 0.2
A1 2.39
A2 0.54
ResErr 0.00846
5%int 7.2
Error rate
Query set size
0.0
0.1
0.2
0.3
0.4
10 20 30 40

Discriminative Power

- Sakai (2006)
- Modified the "stability" method to use bootstrap sampling.
- Changed the focus of the method to determining the discriminative power of a test collection for a given measure.
- DP = the smallest absolute difference in score that yields a swap rate of less than 5% using all the topics in the collection.
- Note that this calibration is still dependent on the set of runs we sample from.


Bootstrap delta, MAP, TR 2004 Terabyte
> 0.01
A1 0.43
A2 0.0121
ResErr 0.00389
5%int 49
> 0.02
A1 0.401
A2 0.0221
ResErr 0.0079
5%int 49
> 0.03
A1 0.362
A2 0.0324
ResErr 0.00783
5%int 49
> 0.04
A1 0.346
A2 0.0486
ResErr 0.00785
5%int 39.8
> 0.05
A1 0.318
A2 0.0644
ResErr 0.00793
5%int 28.9
> 0.06
A1 0.317
A2 0.0887
ResErr 0.00672
5%int 21
> 0.07
A1 0.327
A2 0.123
ResErr 0.00457
5%int 15.4
> 0.08
A1 0.327
A2 0.146
ResErr 0.00353
5%int 13.3
> 0.09
A1 0.296
A2 0.167
ResErr 0.0036
5%int 10.9
> 0.1
A1 0.324
A2 0.2
ResErr 0.00256
5%int 9.6
> 0.11
A1 0.312
A2 0.228
ResErr 0.00251
5%int 8.66
> 0.12
A1 0.292
A2 0.259
ResErr 0.00168
5%int 7.57
> 0.13
A1 0.281
A2 0.277
ResErr 0.00182
5%int 6.93
> 0.14
A1 0.509
A2 0.397
ResErr 0.000491
5%int 6.66
> 0.15
A1 0.379
A2 0.369
ResErr 0.000567
5%int 5.99
> 0.16
A1 0.269
A2 0.336
ResErr 0.000669
5%int 5.01
Error rate
Query set size
10 20 30 40
0.0 0.1 0.2 0.3 0.4


V/B delta, MAP, TREC 2004 Terabyte
> 0.01
A1 0.441
A2 0.0127
ResErr 0.0097
5%int 50
> 0.02
A1 0.425
A2 0.0256
ResErr 0.0128
5%int 50
> 0.03
A1 0.406
A2 0.0387
ResErr 0.0121
5%int 50
> 0.04
A1 0.359
A2 0.0491
ResErr 0.0121
5%int 40.2
> 0.05
A1 0.346
A2 0.0676
ResErr 0.00986
5%int 28.6
> 0.06
A1 0.342
A2 0.0958
ResErr 0.0124
5%int 20.1
> 0.07
A1 0.342
A2 0.122
ResErr 0.0121
5%int 15.8
> 0.08
A1 0.295
A2 0.141
ResErr 0.00882
5%int 12.6
> 0.09
A1 0.351
A2 0.182
ResErr 0.00807
5%int 10.7
> 0.1
A1 0.358
A2 0.208
ResErr 0.00518
5%int 9.51
> 0.11
A1 0.334
A2 0.234
ResErr 0.00608
5%int 8.12
> 0.12
A1 0.37
A2 0.285
ResErr 0.00542
5%int 7.03
> 0.13
A1 0.341
A2 0.295
ResErr 0.00473
5%int 6.54
> 0.14
A1 0.459
A2 0.375
ResErr 0.0029
5%int 5.93
> 0.15
A1 0.26
A2 0.337
ResErr 0.00381
5%int 5
> 0.16
A1 0.23
A2 0.349
ResErr 0.00534
5%int 5
> 0.17
A1 0.125
A2 0.315
ResErr 0.00388
5%int 5
> 0.18
A1 0.209
A2 0.405
ResErr 0.0035
5%int 5
> 0.19
A1 0.697
A2 0.639
ResErr 0.00201
5%int 5
Error rate
Query set size
10 20 30 40
0.0 0.1 0.2 0.3 0.4

Where does stability come from?

- "Stability" or "discriminative power" is a function of
  - the hardness of the topics,
  - the definition of relevance,
  - the document collection,
  - the effectiveness of the systems used to build it!
- Balance is therefore a tricky target, because it is affected by how you observe it.

Abstracted relevance

- In TREC ad hoc (and many related tasks), relevance is defined *minimally:*

  A document is relevant if any part of the document is relevant, even a single sentence.

- ... and *independently:*

  A document is relevant independent of all other documents the user has already seen. The user has no outside context aside from basic world knowledge.

- What are the implications of these relevance limits?

Eliciting relevance

- A minimal, independent threshold for relevance eases the job of the assessor (the person making relevance judgments).
- If relevance is not minimal, the assessor has to make a decision about whether the relevant information they have found is “relevant enough”.
- If relevance is not independent, the assessor has to remember what they’ve read before. This creates a high cognitive load and limits how much they can assess.
- Both situations get in the way of consistency.

Consistency in relevance

- The assessor is the user with this information need.
- Most documents will be obviously not relevant.
- Some will be obviously relevant.
- It doesn't matter how they decide the documents on the boundary, *as long as they do so consistently*.
- Think about duplicate documents, near-duplicates, repeated information.
- The measures do not differentiate documents with the same relevance judgment, so neither should the assessor.

Assessor agreement

- Consistency is different than agreement.
- Consistency: is the assessor being consistent with themselves?
- Agreement: are two (or more) assessors making consistent judgments between them?
- We can measure agreement...
  - Cohen's kappa, with many variations...
  - Overlap and Jaccard coefficients...
- ... but it gets expensive!

Voorhees (SIGIR 1998)

- TREC-4: two sets of relevance judgments for ad hoc topics done by NIST.
- TREC-6: additional relevance judgments done by Waterloo.
- Questions: do assessors disagree? If so, what is the effect on the *measures?*
- Agreement was around 40% (overlap).
- Absolute values of MAP change.
- Rank ordering of systems was essentially identical.

★ assessors do disagree, but differences either have a small effect, or affect all systems equally.

Scholer et al (SPIRE 2004)

- Computed near-duplicate equivalence classes of judged documents in a TREC web task, using shingling.
- Found many instances of judgment differences between members of the same near-duplicate class.
- Seemed to imply that assessors are not consistent!
- Soboroff (unpub) repeated their experiment, and found that assessment differences occurred within nearly identical pages from the same website. Only one page was actually relevant. The assessor was consistent.
- How does this imply we might improve consistency?

Should you care?

- Is agreement important?
- Is consistency important?
- Is there a relationship between agreement and consistency?

- We can design assessment guidelines to promote both agreement and consistency.
  - e.g. minimizing relevance complexity
  - e.g. minimizing cognitive load
- We can also design interfaces to support agreement and consistency.
  - Reliable, repeatable document display.
  - Order similar documents together for assessment.

Making relevance judgments

- Which documents should we make relevance judgments for?
- Original Cranfield method required all documents to be judged with respect to all queries.
- It turns out that this is not necessary.
- We only need to judge enough documents that
  - we have an unbiased sample of the relevant documents.
  - we have an unbiased estimate of the number of relevant documents that exist in the collection.

Pooling

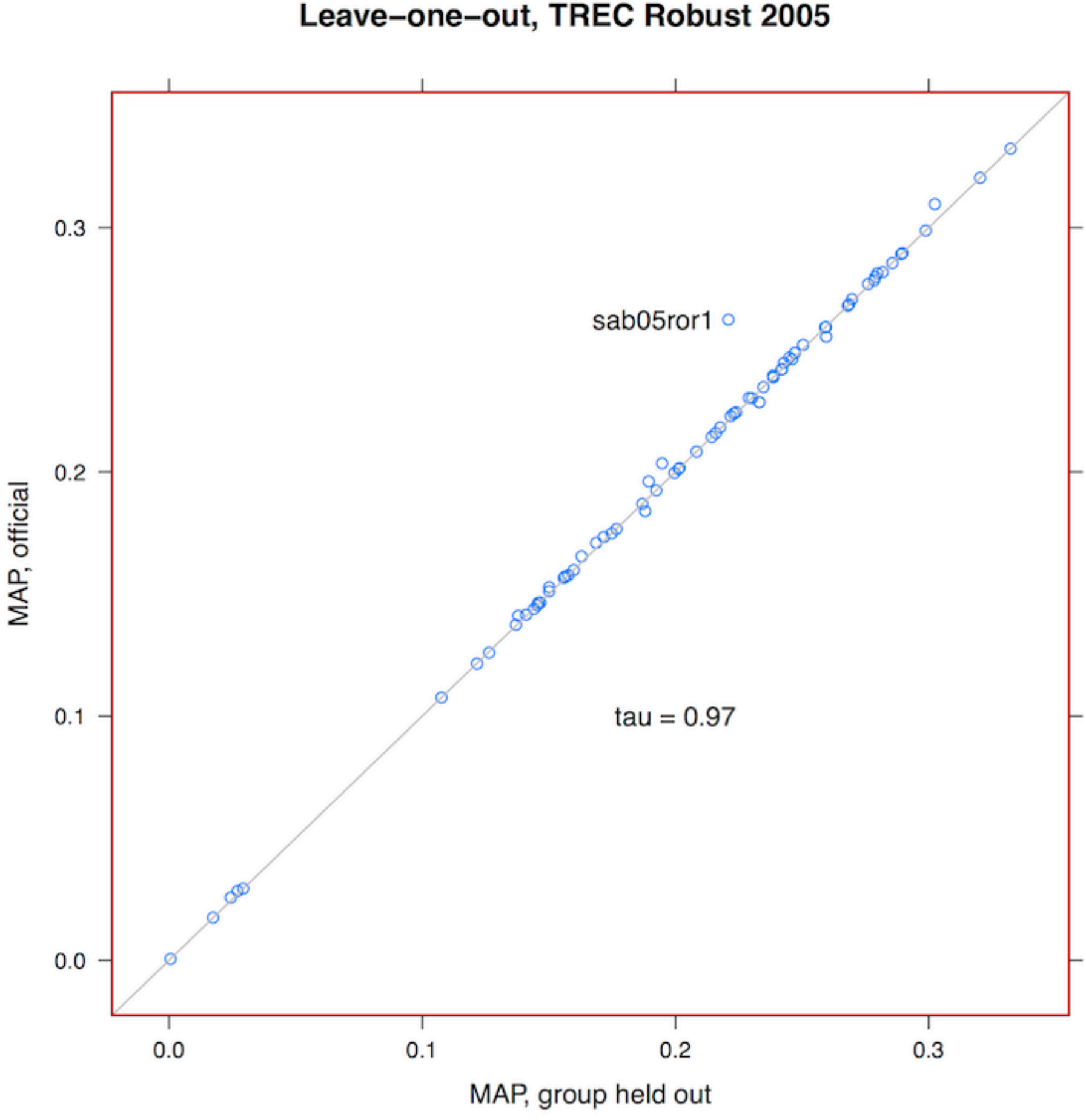
- Pooling is a strategy to avoid reading all the documents in the collection.
- Originally proposed and analyzed by Karen Spärck Jones, Keith van Rijsbergen, and Stephen Robertson in the 1970s.
  - three classic "British Library reports" can be read at http://www.sigir.org/museum/allcontents.html
- Operationalized in TREC.

TREC Pooling

- In TREC, topics are released to the world without relevance judgments.
- Participants in TREC generate a ranked list of the top *k* documents for each topic, using any method (a "run").
  - k is typically 1,000 or 10,000.
- TREC organizers combine the top *j* << *k* documents from some or all runs to form a "pool".
  - j is typically 100 in the ad hoc collections.
  - However, it varies due to measurement goals and budget considerations.
  - A document only goes into the pool once, so for *r* runs and a pool depth of *j*, the pool size $s \leq r \times j$.
- The documents in the pool are judged for relevance.
- All unpooled documents are assumed to be not relevant.

Does pooling work?

- Leave-one-out experiment (Zobel, SIGIR 1998)
- For each group contributing runs to the pool...
  - Hold out that group's runs from the pool.
  - (This removes their unique contributions, creating a pool biased somewhat against that group, as if they had not participated.)
  - Measure those systems again using the biased pool.
  - Compare the difference in scores between the biased pool and the "complete" pool.
- For the collections and runs Zobel examined, the bias due to being left out was not large enough to be a concern.



Why does pooling work?

- The systems contributing to the pools are *trying* to rank relevant documents above nonrelevant ones.
- The systems are *diverse* enough to uncover an unbiased sample of the relevant documents.
- The pools are *deep* enough to discover an unbiased sample of the relevant documents.
- There are few enough relevant documents that a *sufficient* sample is contained in the pool.

Pooling bias

- Sometimes, pooling *doesn't* work.
- Systems reasons:
  - scale prevents interesting retrieval approaches.
  - everyone uses the same baseline system.
  - the state of the art is very immature.
- Collection reasons:
  - the topics are very large w.r.t. the pooling depth
- Can you think of other reasons?

Getting Systems to Pool

- In TREC and similar venues, participating groups share their attempts to find relevant documents using their own systems.
- Those systems are typically not "stock" but come out of the research of those participating groups.
- Furthermore, groups choose to participate in an evaluation because they are interested in making systems that can effectively solve the task in the evaluation.
- Outside of an evaluation (or even inside!) we need to consider how to achieve pool diversity.

Iterative Search and Judge

- (Cormack et al, SIGIR 1998)
- Manually search the collection to find relevant documents.
- Goal: find as many relevant documents as possible.
- Use multiple searches, different queries.
- Single search tool, or multiple.
- Relevance feedback might be useful.

ISJ with Feedback

- (Soboroff and Robertson, SIGIR 2003)
- Initial search: one query, relevance judgments on top 100 retrieved documents.
- Judgments used in six feedback mechanisms:
  - Rocchio feedback with SMART
  - BM25 with R/SJ feedback using PRISE
  - Relevance models with YARI (a pre-Indri system)
  - Three classifiers in from McCallum's BOW toolbox: SVM, kNN, naive Bayes

Pool diversity

- Saturation point idea
- For 1 random system, expected relevant document loss is x
- For 2 systems, it's x' < x.
- Measure both mean and variance of loss.
- At a certain point, additional systems do not recover more relevant documents.

Varying the pool depth

- Shallow pools find fewer relevant documents.
- How deep to the pools need to be to estimate R?
- Probability of relevance at rank *k* decreases.

Move-to-front pooling

- Cormack et al, SIGIR 1998.
- Arrange runs randomly in a queue.
- Draw the first run from the queue and examine its first unexamined retrieved document.
- If the document is not relevant, put the run at the rear of the queue.
- Otherwise, examine the next document from that run.
- Stop when no more relevant documents are found.
- Given the pooling assumptions, MTF pooling will find the same relevant documents while examining fewer documents.

Pooling as sampling

- Pooling is a sampling method.
- Gilbert and Sparck Jones (1979) explored the implications of this, given several assumptions, in the third BL report.
- Their assumptions:
  - all (or nearly all) relevant documents would be in the pool.
  - there are not that many relevant documents.
  - a random sample of the pool is assessed for relevance.

Random sampling

- In practice, TREC pooling looks at the entire pool.
- If draw a sample of the pool, we can then compute estimates of precision, recall, etc.
- Specific techniques explored by Aslam et al (SIGIR 2006), Yilmaz and Aslam (CIKM 2006), Yilmaz et al (SIGIR 2008)
- These techniques were used to judge small samples of the pools in the TREC Terabyte track, producing mostly reasonable rankings but overestimating the actual values.

Stratified sampling

- Small samples are a problem.
- Yilmaz and Aslam (2008) proposed a method for taking a sample stratified by pool depth.
- Pavlu proposed a sampling strategy based on accurately estimating average precision (used in the TREC MQ track)

Sampling in the Legal track

- TREC legal track: identify documents responsive to a request in a legal discovery scenario.
- Frequently the search needs yield lots of responsive documents.
  - Some TREC cases: 10% of the collection!
- Stratified sampling very deep in the collection continued to find yet more responsive documents.

Sampling: the bottom line

- Small samples are a problem.
- Especially when you have to pick the sampling rate before making any relevance judgments.
- And even more so when you don't have any ground truth.
- Modern web-scale collections mean every sample is a small sample.
- Scoping the topics is one approach.
- Scoping the task is another.
- Recall considerations make topics and task design a series of tradeoffs.

Assessors

- NIST uses paid contractors as assessors.
- We hire people specifically who can read lots of documents/web pages/what have you and focus critically on them.
- This is not the typical skill set of a university student (or IR researcher!)
- Keep in mind that you will be asking (or paying) people to do a long, repetitive task that requires focus. This has implications for software design, remuneration.
- Do your assessors have the incentive to do a good job?

Crowdsourcing

- Crowdsourcing is the opposite approach:
  - get people to do a tiny amount of work,
  - with a tiny amount of training,
  - for a tiny sum of money.
- That creates some obvious issues!
- Omar Alonso (Bing) has taught an excellent tutorial on crowdsourcing for IR.
- Read everything written by Omar on this subject.
- Read everything written by Panos Iperiotis (NYU) too.