Experimental Methods for Information Retrieval

Donald Metzler (Google)
Oren Kurland (Technion)

Originally presented as a half-day tutorial at SIGIR '12.

Who We Are

Don

Oren

Tutorial Outline

- Introduction [15 minutes]
- A Bird's-Eye Overview [30 minutes]
- Deep Dive [120 minutes]
- Summary [5 minutes]

Motivation

- Information retrieval (IR) is a highly applied *scientific* discipline
- Experimentation is a critical component of the scientific method
- Poor experimental methodologies are not scientifically sound and should be avoided
- IR has become plagued with weak experimentation, causing:
  - Outsiders to think of IR as non-scientific
  - A plethora of minor improvements vs. weak baselines
  - Difficulty in defining the "state-of-the-art"

Experimentation in IR

- Three fundamental types of IR research:
  - Systems (efficiency)
  - Methods (effectiveness)
  - Applications (user utility)
- Empirical evaluation plays a critical role across all three types of research
- Our primary focus is on "methods" research, but the same principles can be applied elsewhere (including other disciplines)

Why is Empirical Evaluation Important…

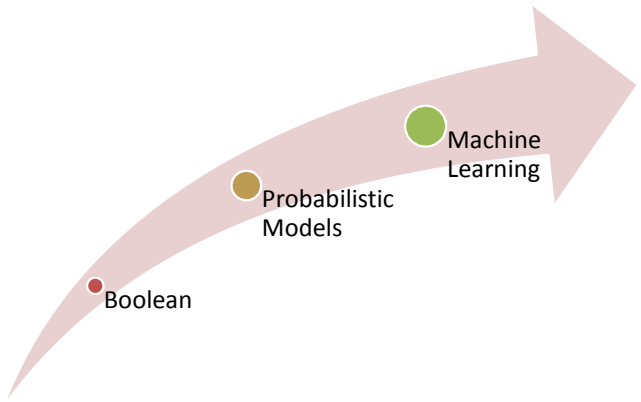
To Researchers?

- It allows you to convince others (e.g., reviewers, other researchers, funding agencies) that your work is meaningful
- Without a strong evaluation, your paper will (probably) be rejected
- Empirical evaluation helps guide meaningful research directions

To Practitioners?

- It allows you to convince others (e.g., company VPs, investors, clients) that your work is meaningful
- Without a strong evaluation, your code will (probably) not be deployed
- Empirical evaluation helps guide meaningful development directions

Up, Up, and Away?



Fundamental Question

- How much, if at all, has information retrieval advanced over the past X years?
- Important question for the entire field
- There is unfortunately no clear answer
  - Moffat and Zobel. [WISE '04]
  - Armstrong et al. [CIKM '09]
  - TREC Economic Impact Study
- With rigorous experimental methodologies, these questions become easier to answer

Tutorial Goals

- Primary goals:
  - Highlight importance of experimental evaluation in IR
  - Provide an in-depth overview of the fundamental IR evaluation paradigm
- Secondary goal:
  - Help develop and sustain a culture of (even) strong(er) experimental evaluations
- Important: This tutorial is *not* about devising evaluation measures or building test collections

Empirical Evaluation: A Bird's-Eye View

- Brief high-level overview of empirical evaluation
- Topics covered:
  - The fundamental evaluation paradigm
  - The core components of (solid) empirical evaluation
  - 3 high-level examples: ad hoc retrieval, text categorization, and question answering

The Fundamental Evaluation Paradigm

- The scientific method from an empirical point of view:
  - *Objective* and *reproducible test* of a well-defined *hypothesis*
  - Most often carried out as an *experiment*
- IR example:
  - The cluster hypothesis (Rijsbergen '79)
    - Jardine and Rijsbergen '71, Voorhees '85, Smucker and Allan '09

Core Steps of An Empirical Evaluation

- **What** do we evaluate?
- Before writing any code, running any experiments, etc., it is essential to:
  - Clearly define a *task*
  - Formulate a set of *testable hypotheses*

Defining Tasks

- It is very difficult to solve ill-defined problems/tasks
- Are you solving a well-known existing task? Or something that is similar to an existing task?
  - E.g., ad hoc (query-based) retrieval, categorization, question answering, clustering, filtering, etc.
- Is your task novel and unlike anything that has been done before?
  - How is the system used?
  - What are the inputs? outputs?
  - How do you define success?

Important: No matter how good your research is, or how good your results are, it is of little value if nobody understands what problem you're trying to solve.

Formulating Hypotheses

- Statement of the hypothesis to be tested
  - Concise
    - e.g., contrast the algorithm and system points of view
  - The hypothesis either holds or does not, with some caveats:
    - Scope: with respect to the data we use for experiments
    - Extent: might hold to a limited extent and/or under certain conditions
  - Important questions to be asked:
    - What specific component of our proposed method actually concerns the hypothesis? Can we test this component *in isolation*?

Always remember: Methods are not devised in an arbitrary manner. We always have a hypothesis (whether implicit or explicit) for why our work should improve upon existing research.

Core Steps of An Empirical Evaluation

- **How** do we evaluate?
- Key components necessary to carry out a meaningful evaluation:
  - Experimental *methodology*
  - *Analysis* of results

Experimental Methodology

- Provide full details of the implementation for reproducibility
  - Free parameters, data processing (e.g., stopword removal, stemming), toolkits used, etc.
- Basic ingredients
  - Benchmarks
    - Variety, representativeness, and more to be discussed later
    - Publicly available vs. proprietary
  - Reference comparisons (baselines)
    - Pertaining to the task (and hypothesis) at hand, strong enough
    - Re-implementation vs. quoting numbers
  - Evaluation metrics
    - Variety, connection to the task/hypothesis
  - Parameter tuning
    - Tune parameters of new approach *and* reference systems
    - Consider other factors that affect performance

Key takeaway: If you're lucky, someone else may want to re-run your experiment in the future. It is up to you to make sure this is possible by using a good methodology and describing it adequately.

Analysis of Results

- How well did our methods do (on average) with respect to the reference comparisons?
  - Are the improvements consistent across benchmarks?
  - Anecdotal differences vs. substantial differences vs. statistically significant differences
- Analysis of factors affecting the performance
  - Sensitivity to values of free parameters
  - Analyzing the performance of specific cases (instances) of our method (those that enable us to understand whether the performance differences can be attributed to the tested hypothesis)
  - Failure analysis

Bottom line: Do the results provide substantial support to the stated hypothesis?

Using the Fundamental Evaluation Paradigm: Ad Hoc Retrieval

- The task: ad hoc (query-based) retrieval
- Hypothesis (1):
  - Accounting for term-proximity information when representing texts results in retrieval performance that transcends that of using a bag-of-terms representation
- Hypothesis (2):
  - Using estimates of type X that use term-proximity information yields improved retrieval performance with respect to previously proposed methods that utilize term proximities with estimates of type Y
- Testing the hypothesis:
  - A novel retrieval method that utilizes term-proximity information

Testing Hypothesis #1

- Experimental setup
  - Publicly available TREC benchmarks for ad hoc retrieval
    - Queries, documents, relevance judgments
  - Data pre-processing (more details later)
  - Reference comparisons
    - The language-modeling approach, Okapi BM25, DFR, etc.
      - (strong) baselines that use a bag-of terms representation
      - freely available toolkits
  - Evaluation metrics: MAP, P@k, NDCG, ERR,…
  - Tuning our method and the reference comparisons
    - Cross validation over queries for setting free-parameter values
- Experimental results
  - Our method outperforms the reference comparisons in a consistent, substantial, and statistically significant manner?

Using the Fundamental Evaluation Paradigm: Text Classification

- The task: topical text classification
- The hypothesis: Modeling term proximities in the text improves upon using a bag-of-terms representation
- Testing the hypothesis
  - Using support vector machines (SVMs) with polynomial kernels of a second degree

Testing the Hypothesis

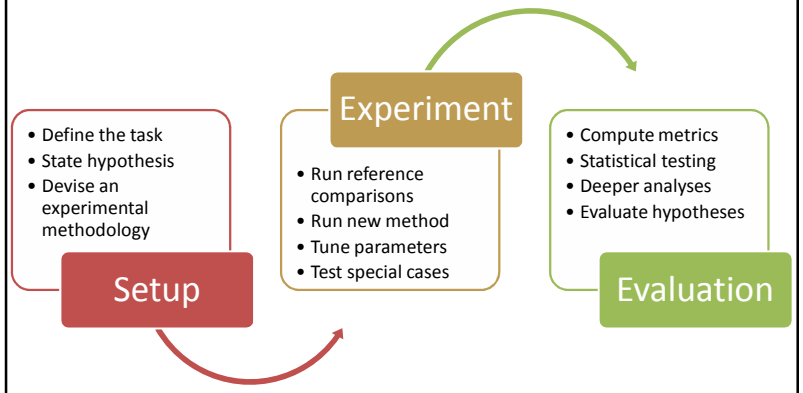
- Experimental setup
  - The publicly available Reuters benchmarks
  - Data pre-processing
  - Reference comparisons
  - Evaluation metrics: (Macro/micro) Precision/Recall/F1
  - Training our method and the reference comparison
- Experimental results
  - Does our method outperform the baselines in a consistent, substantial, and statistically significant manner?
  - Example for a solid evaluation of text categorization methods: Yang and Liu '99
  - Example of contradicting results in the literature:
    - Joachims '98 vs. Dumais et al. '98 (Rocchio vs. Naïve Bayes)

Using the Fundamental Evaluation Paradigm: Question Answering

- The task: passage retrieval for a QA system
  - Answers are subsequently extracted from the retrieved passages
- Hypothesis: improved passage retrieval results in improved overall QA performance
  - cf., Tellex et al. '03, Collins Thompson et al. '04
- Testing the hypothesis
  - A novel passage retrieval method

Testing the Hypothesis

- Experimental setup:
  - The TREC QA benchmarks
    - Questions, relevance judgments (for documents), answer-extraction patterns
  - Data pre-processing (more details later)
  - Reference comparisons
    - A previously proposed effective passage retrieval method
  - Evaluation metrics: MAP, MRR
  - Tuning our method and the reference comparisons
    - Cross validation over questions
- Experimental results
  - Does our method outperform the baseline in a consistent, substantial, and statistically significant manner?



Empirical Evaluation: Deep Dive

- The remainder of the tutorial fleshes out the details of the broad topics just covered
- Not meant to be an exhaustive treatment
- We cover the most important aspects in detail and briefly touch upon a number of secondary (but still important) issues

Running Example

- The task: Ad hoc retrieval
- The hypothesis: Emphasizing the original query terms in pseudo-feedback-based query expansion methods is important (to ameliorate query drift)
- The method: A pseudo-feedback-based query expansion approach that performs *query anchoring* (RM3, Abdul Jaleel et al. '04)

Running Example (contd.)

- q - a query
- D_q - an initial list (the "pseudo feedback list") of n documents retrieved in response to q by some search method
  - e.g., the query likelihood method (Song&Croft '99)

    $Score_{QL}(d;q) \triangleq p(q|d) \triangleq \prod_{q_i \in q} p(q_i|d)$;

    q_i is a query term

Running Example (contd.)

R - a relevance language model, which can be thought of as an expanded query form

$p(w|R_{RM1}) \triangleq \sum_{d \in D_q} p(w|d)p(d|q)$; Lavrenko&Croft '01

$p(w|R_{RM3}) \triangleq \lambda \boxed{p(w|q)} + (1-\lambda)\, p(w|R_{RM1})$; Abdul Jaleel et al. '04

query anchoring

- We "clip" R so as to use the m terms to which it assigns the highest probability, and then normalize to have a probability distribution
- Scoring document d in the corpus with respect to R is by their "similarity" (e.g., the KL divergence)

"Our" Method (RM3)

- Uses language models
  - Implies to a must-have reference comparison – the query likelihood method used to create the initial ranking
  - The query likelihood model is a special case of our approach (λ=1)
  - **Hypothesis**: our query expansion method outperforms the approach of not using query expansion
- Builds upon a previous method (RM1) that constitutes as a specific case of our approach (λ=0)
  - Implies to a must-have reference comparison – RM1
  - **Hypothesis**: query anchoring is important
- Incorporates 3 free parameters
  - *n* – the number of top-retrieved documents used to construct the relevance model
  - *m* – the number of terms used by the relevance model
  - λ – the query anchoring parameter

Setting Up An Experiment

- Recall the basic formula:
  - Full specification for reproducibility
  - Benchmarks
    - The "Cranfield paradigm" (Cleverdon and Kean '68)
  - Implementation details (e.g., data processing steps, toolkits)
  - Reference comparisons
  - Evaluation metrics
  - Tuning free-parameter values

Reproducibility

- We must specify each and every detail needed for re-implementing our method and the experimental setup
  - Benchmarks and data pre-processing steps
  - Toolkit used
  - More later…

Effectiveness vs. Efficiency

- There are different ways of evaluating a system
- Modeling papers, as ours here, typically care mostly about effectiveness, while systems papers focus on efficiency
  - In practice, both are important
- Although here we care about effectiveness, efficiency issues should also be discussed
  - Two retrievals are performed, the second is based on using a "long" query

Open Source Software

- Don't reinvent the wheel!
- Some open source toolkits that are commonly used for ad hoc IR experiments
  - Indri/Lemur/Galago (www.lemurproject.org)
    - Language models, Okapi, Markov Random Fields (MRF)
  - Terrier (http://terrier.org/)
    - Language models, divergence from randomness (DFR) methods, and MRFs
  - Lucene (http://lucene.apache.org/)
    - Serves as a platform to commercial products
    - Very basic retrieval methods, no built-in support for pseudo-feedback-based retrieval
    - Some support for basic language modeling approaches and Okapi BM25

Choosing Your Corpora

- Availability (public vs. proprietary)
  - Whenever possible, use publicly available datasets for (i) reproducibility, and (ii) sanity checks with previously reported numbers
- Stick to main stream whenever possible
  - TREC vs. CLEF vs. NTCIR etc. as the main benchmarks
- Appropriateness for the task
  - Basic ad hoc retrieval (yes, in our case) vs. patent retrieval (can be an additional benchmark)
- Size
  - Incurs not only scale issues (e.g., potential for query drift)
  - ClueWeb category A: ~500,000,000 documents
  - AP (TREC disks 1-3): ~240,000 documents
- Age
  - Use newly released corpora when possible (e.g., ClueWeb)
- Properties
  - Newswire vs. Web corpora
  - Homogeneous vs. heterogeneous
  - "Clean" vs. "noisy"

Example TREC corpora

Corpus	Data	# docs	Properties
AP	TREC disks 1-3	242,918	Newswire, homogeneous
WSJ	TREC disks 1-2	173,252	Newswire, homogeneous
FR	TREC disks 1-2	45,820	Federal register, long documents
ROBUST	TREC disks 4,5 –{CR}	528,155	Mainly newswire, somewhat heterogeneous
WT10g	WT10g	1,692,096	Noisy web corpus
GOV2	GOV2	25,205,179	"Clean" web corpus
ClueWeb Cat A	ClueWeb	503,903,810	Noisy web corpus
ClueWeb Cat B	ClueWeb	50,220,423	Noisy web corpus

Selecting Your Queries

- Similar considerations as corpora
- The corpora are often accompanied with queries in the benchmarks
  - We need a really good reason to select only a subset of queries from a given benchmark
- Use as many queries as possible per benchmark (at least 50)
  - Provides a broad picture and may better attest to the potential generalization of the results
  - Enables analysis of performance variance and proper statistical significance testing
  - Enables informative failure analysis
- Also consider properties of the queries in the context of your task, model, and goals
  - Short vs. verbose (titles of TREC topics vs. descriptions)
  - Long tail vs. short tail (query logs)
- A common "pain point"
  - Coming up with our own queries and judgments

Relevance Judgments

- Binary vs. Graded
- Crowdsourced
  - Many challenges; out of the scope of this tutorial
- Inter-annotator agreement
- Fair, unbiased annotators (e.g., not the authors)

Data Pre-Processing

- No standard technique
- One of the biggest hindrances of reproducibility
- Use standard tokenization, stemming/lemmatization (e.g., Porter or Krovetz), stopword removal (e.g., the INQUERY list) practices
  - Relatively easy if using an open source toolkit
  - Can affect performance to a substantial and statistically significant degree
  - Over the Web, spam removal can be an important pre-processing step (Cormack et al. '11)
- Justify your choices
  - Why (not to) remove stopwords (robustness?)
- **Always** test your methods, and the reference comparisons, with data that was processed in exactly the same way
  - Citing performance numbers reported in published papers is not acceptable for performance comparisons (but can be used for sanity checks); unless, we are comparing systems and not methods

Free Parameters

- Determining the free parameters of your model
- Which parameters need to be tuned?
  - If not sure, tune them all, but don't "go crazy"
  - In our case: n (number of top-retrieved documents), λ (the query anchoring parameter), and m (number of terms used)
- Which can stay fixed?
  - Those for which either (i) the performance is known to be robust across corpora and queries, or (ii) widely "accepted" default values
  - In our case, the language model Dirichlet smoothing parameter (μ=1000; Zhai and Lafferty '01), but with caveats
- How should they be tuned?
  - Often a crucial question
  - One of *the* major obstacles for papers to get in…
  - Importance of training/test splits, cross-validation, etc.
  - Optimize for the right metric

Setting Free-Parameter Values

- Per-query over fitting
  - Enables to explore whether the "information sources" our method uses (i.e., query anchoring) on top of those previously used (RM1) are of merit
  - Completely neutralizes the effects of free-parameter values
  - Highly uncommon
  - Can we come up with a technique for setting free-parameter values on a per-query basis? (cf., adaptive relevance feedback; Lv and Zhai '09)
- Selecting values that yield optimal average performance for the set of queries in a benchmark
  - Ameliorates the effect of free-parameter values, but still a form of overfitting
  - Enables us to explore the relative performance patterns of methods when used with free-parameter values that are effective on average
- Train/test split; cross validation
  - Find optimal free-parameter values for a train set (of queries), and apply these for the test set of queries
  - As the number of queries is often small, use cross validation

The Train/Test Paradigm for Setting Free-Parameter Values

- Preferably, the train and test queries should be for the same corpus
- If possible, avoid non-random splits of the query set
- Metric divergence issues (Azzopardi et al. '03, Morgan et al. '04, Metzler and Croft '05)
  - The metric for which performance is optimized in the training phase should be the same as that used for evaluation over the test set
  - Do not optimize "unstable" metrics such as p@k or MRR; use MAP/NDCG instead
- How to find optimal free-parameter values with respect to a metric at hand?
  - Line/exhaustive search
  - Coordinate ascent (Metzler and Croft '05)
  - Other learning to rank methods (Liu '09)

Metric Divergence is Real

	ROBUST				WT10G			
	MAP	p@5	NDCG@5	p@10	MAP	p@5	NDCG@5	p@10
Initial	24.8	47.9	50.0	42.8	19.6	33.4	32.8	28.2
RM1 (OptAQ)	23.4	48.6	50.9	41.6	16.6^{i}	33.6	31.2	28.0
RM3 (OptAQ)	28.1^{i}_{r}	49.8	51.9	43.7^{i}_{r}	21.7^{i}_{r}	34.2	33.0	28.6
RM1 (OptPQ)	28.7^{i}	56.5^{i}	59.5^{i}	48.8^{i}	23.2^{i}	43.1^{i}	41.4^{i}	36.4^{i}
RM3 (OptPQ)	$\mathbf{34.4}^{i}_{r}$	$\mathbf{59.6}^{i}_{r}$	$\mathbf{62.5}^{i}_{r}$	$\mathbf{52.2}^{i}_{r}$	$\mathbf{30.3}^{i}_{r}$	$\mathbf{49.1}^{i}_{r}$	$\mathbf{47.6}^{i}_{r}$	$\mathbf{42.4}^{i}_{r}$
RM1 (LOO)	23.4	48.6	50.9	41.6	16.6^{i}	33.6	31.2	28.0
RM3 (LOO)	27.3^{i}_{r}	48.5	50.5	43.0	21.2_{r}	33.8	32.2	28.5_{r}
RM1 (10Fold)	23.4	48.6	50.9	41.6	15.7^{i}	32.0	29.4	25.9
RM3 (10Fold)	27.9^{i}_{r}	49.2	51.3	43.5_{r}	20.4_{r}	34.0	32.1	28.6

Table 1: MAP optimization.

	ROBUST				WT10G			
	MAP	p@5	NDCG@5	p@10	MAP	p@5	NDCG@5	p@10
Initial	24.8	47.9	50.0	42.8	19.6	33.4	32.8	28.2
RM1 (OptAQ)	23.4	48.6	50.9	41.6	16.6^{i}	33.6	31.2	28.0
RM3 (OptAQ)	27.2^{i}_{r}	48.0	50.5	44.6^{i}_{r}	20.8_{r}	35.1	32.7	31.0^{i}
RM1 (OptPQ)	24.4	55.6^{i}	57.0^{i}	54.3^{i}	19.0	43.5^{i}	38.0^{i}	41.9^{i}
RM3 (OptPQ)	$\mathbf{27.7}^{i}_{r}$	$\mathbf{56.9}^{i}$	$\mathbf{57.8}^{i}$	$\mathbf{59.5}^{i}_{r}$	$\mathbf{23.6}^{i}_{r}$	$\mathbf{47.4}^{i}_{r}$	$\mathbf{41.5}^{i}_{r}$	$\mathbf{49.5}^{i}_{r}$
RM1 (LOO)	23.4	48.6	50.9	41.6	15.5^{i}	33.6	31.3	26.3
RM3 (LOO)	26.9^{i}_{r}	46.6	48.7	41.1^{i}	20.6_{r}	34.2	32.3	29.9_{r}
RM1 (10Fold)	23.4	48.6	50.9	41.6	15.7^{i}	33.8	31.1	27.3
RM3 (10Fold)	27.2^{i}_{r}	48.8	50.2	43.1	20.5_{r}	34.0	32.0	29.5

Table 2: p@10 optimization

'i' and 'r' mark statistically significant differences with the initial ranking (QL) and RM1, respectively.

Reference Comparisons

- How to choose the right baseline?
  - Uses principles similar to those used by our method
    - In our case: Mixture model of Zhai and Lafferty '01, which also performs query anchoring
  - Can shed light on the tested hypothesis
    - In our case: RM1 which does not use query anchoring and which is a specific case of our RM3 method; and, the QL method that does not perform query expansion and which is also a specific case of our method
  - Try to stay "within the same framework"
    - In our case, is the comparison with Rochhio's (vector space) method ('71) informative?
    - Is the comparison with term-proximity-based query expansion methods informative?
  - Known to yield strong performance; preferably, considered state-of-the-art
    - In our case: Mixture model of Zhai and Lafferty '01
- What to do if no baseline exists (e.g., for new tasks)
  - Try to adapt an existing method proposed for a related task
- The "musts":
  - Implement the reference comparison yourself or get the code; do not cite numbers
  - Tune the reference comparisons with respect to free-parameter values using the same approach employed for our methods

Evaluating Experimental Results

- Now that you have some results, what do you do with them?
- Evaluation involves *measuring*, *comparing*, and *analyzing* your results
  - Helps prove (or disprove) your hypotheses
  - Demonstrates how your methods or systems compare against the existing state-of-the-art
  - Provides fundamental insights into the underlying research problems being addressed

Evaluation Measures

- Summary of most commonly used measures for different tasks on the next slide
- Measures should be chosen based on properties of the system being developed
  - E.g., is it a recall-oriented system (legal domain) or a precision oriented system (Web search, entry-page finding)?
- Deep analysis is often more interesting than tables full of numbers (don't overdo it!)

Commonly used evaluation measures

Measure	Binary/Graded	Properties
MAP (mean non-interpolated average precision)	Binary	Widely used, stable; variants: GMAP, infAP, statAP
p@k	Binary	Precision at top ranks, unstable
MRR (mean reciprocal rank)	Binary	Highly unstable, often used in passage retrieval for QA
Recall	Binary	Not used much nowadays, except for specific applications such as legal search
NDCG (normalized discounted cumulative gain; Jarvelin and Kekalainen '02)	Graded	Widely used for Web search
ERR (expected reciprocal rank; Chapelle et al. '09)	Graded	Became a standard in TREC for Web search

Statistical Significance Testing

- The performance of retrieval methods can substantially vary across queries
  - NRRC RIA workshop '03, Robust track '04
- Big changes of performance don't necessarily mean the delta is meaningful
  - Might be the result of large improvements for very few queries, while (slightly) degrading performance for many other queries
  - Pseudo-feedback-based query expansion methods are known to improve retrieval effectiveness on average, but to hurt performance for many specific queries
  - Take extra care when using (very) small query samples
    - E.g., when using a single train-test split of the query set
- Statistical significance testing
  - Paired, two-tailed, tests; p-value=0.01, 0.05, …
  - Recommended tests: paired t-test, permutation test (Smucker et al. '09)
  - Approach: contrast two vectors of performance numbers for a set of queries
- Other measures/analyses of performance "robustness"
  - RI (robustness index; Sakai et al. '05)
  - Histograms (Metzler and Croft '07, Collins-Thompson '08)

Presenting Your Results (Example)

	ROBUST			WT10G		
	MAP	p@5	NDCG@5	MAP	p@5	NDCG@5
Initial	24.8	47.9	50.0	19.6	33.4	32.8
RM1 (OptAQ)	23.4	48.6	50.9	16.6^{i}	33.6	31.2
RM3 (OptAQ)	28.1^{i}_{r}	49.8	51.9	21.7^{i}_{r}	34.2	33.0
RM1 (OptPQ)	28.7^{i}	56.5^{i}	59.5^{i}	23.2^{i}	43.1^{i}	41.4^{i}
RM3 (OptPQ)	$\mathbf{34.4}^{i}_{r}$	$\mathbf{59.6}^{i}_{r}$	$\mathbf{62.5}^{i}_{r}$	$\mathbf{30.3}^{i}_{r}$	$\mathbf{49.1}^{i}_{r}$	$\mathbf{47.6}^{i}_{r}$
RM1 (LOO)	23.4	48.6	50.9	16.6^{i}	33.6	31.2
RM3 (LOO)	27.3^{i}_{r}	48.5	50.5	21.2_{r}	33.8	32.2
RM1 (10Fold)	23.4	48.6	50.9	15.7^{i}	32.0	29.4
RM3 (10Fold)	27.9^{i}_{r}	49.2	51.3	20.4_{r}	34.0	32.1

Table 1: OptAQ– optimizing MAP over all queries, OptPQ– optimizing MAP per query, LOO– leave-one-out using MAP for optimization, 10Fold– 10-fold cross validation. i,r - statistically significant differences with Initial and RM1.

	ROBUST			WT10G		
	Docs	Terms	OrigQueryWeight	Docs	Terms	OrigQueryWeight
RM1 (OptAQ)	10	500	0	10	500	0
RM3 (OptAQ)	10	100	0.1	5	50	0.2

Table 2: Free parameter values for OptAQ.

"Meaningful" Performance Differences

- Performance measures should be taken with a grain of salt
- Just because method A is (statistically significantly) better than B according to some metric, doesn't mean A is fundamentally better than B
- Important to be able to understand the difference between incremental and meaningful performance differences
- What makes a difference "meaningful"?
  - Depends on the problem, domain, application, etc.
  - In some applications, small gains in performance can yield great reward
  - In other applications, gains must be substantial to make a difference

Deeper Analysis

- Up until this point, we've discussed how to measure and compare performance
- It is tempting to stop here and declare victory (or defeat)
- By analyzing your empirical results deeper, you can often gain more insights into the problems being studied
- What are some deeper analysis techniques?

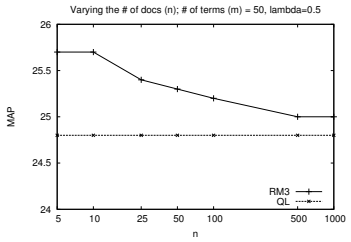
Parameter Sensitivity

- Most methods and systems have some set of free parameters, as described earlier
- It is often interesting to understand how sensitive performance is to the setting of these parameters
- If performance is highly sensitive, then the method may not generalize well across data sets and careful tuning may be necessary
- If performance is independent of the parameter setting, then using some "default" value may be fine
- To demonstrate the sensitivity of performance
  - Hold all other variables fixed at "reasonable" values or tune them to optimal values
  - Compute performance across the entire range of "reasonable" values for the free parameter of interest
  - Plot performance vs. parameter setting value

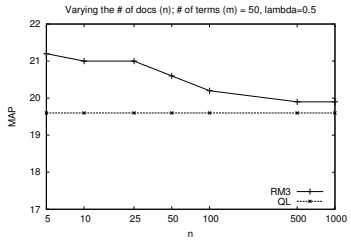
Parameter Sensitivity Example

(Varying # of docs (n), fixing other parameters' values)

ROBUST



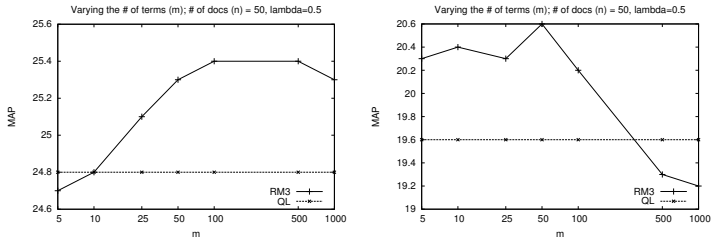
WT10G



Parameter Sensitivity Example

(Varying # of terms, fixing other parameters' values)

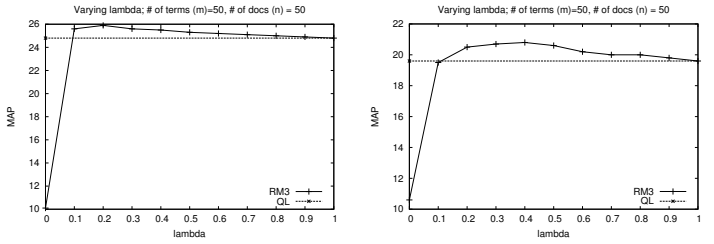
ROBUST **WT10G**



Parameter Sensitivity Example

(Varying lambda, fixing other parameters' values)

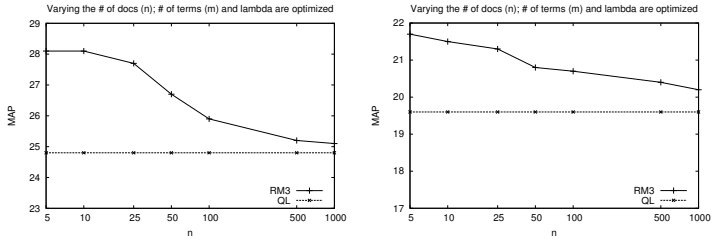
ROBUST **WT10G**



Parameter Sensitivity Example

(Varying # of docs (n), optimizing other parameters' values)

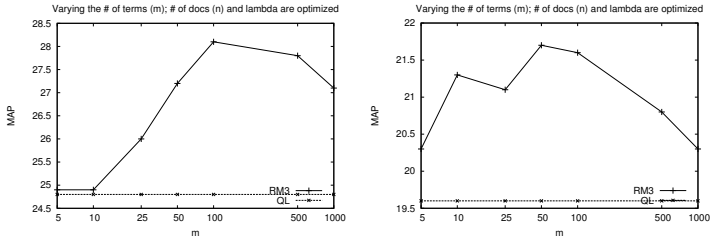
ROBUST **WT10G**



Parameter Sensitivity Example

(Varying # of terms (m), optimizing other parameters' values)

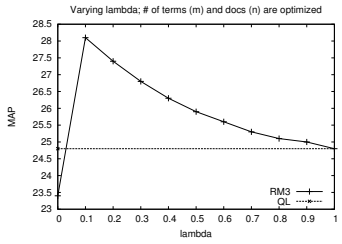
ROBUST **WT10G**



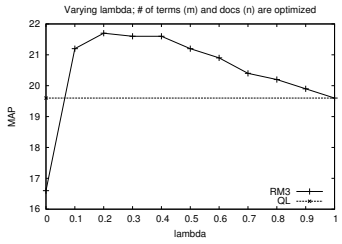
Parameter Sensitivity Example

(Varying lambda, optimizing other parameters' values)

ROBUST



WT10G



Special Cases

- Many systems and methods are built on top of, or generalize, existing approaches
- There are often *special cases* where some setting of the parameters gives rise to an existing approach
  - $\lambda = 0$ in our example model (RM3) is the same as RM1
  - $\lambda = 1$ in our example model (RM3) is the same as QL
  - More common than you may think!
- Importance of special cases
  - Theoretical: useful to draw connections between new and existing approaches
  - Practical: it is a good sanity check (for yourself and the reviewers) to show that the performance of the special case is as expected

Detailed (Micro) Analysis

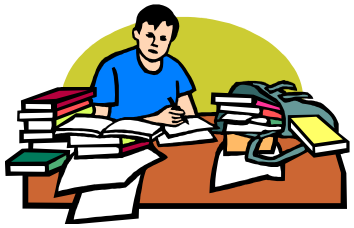
- Aggregate measures are good for analyzing average performance
  - Can mask some interesting phenomena
- It is always worthwhile to do more fine-grained analysis
- Such analysis can provide deep insights into the strengths (and weaknesses) of your approach and help hypothesize how they may be overcome (e.g., as future work)
- Examples
  - Looking at the specific queries (or query types) that were most helped or hurt as the result of your method
  - Analyzing the best and worst performing categories for a text classification task
  - Studying the importance of different features in a learning to rank system

Recap

- You should now be able to:
  - Devise strong experimental methodologies
  - Convince yourself and others that your new system or method does (or does not) advance the state-of-the-art
  - Evaluate other researchers' empirical evaluations
- Your homework

Your Homework

- Identify papers with exceptionally strong/weak empirical evaluations
- Think of how you would use the toolbox developed here in your own work



Key Takeaways

- Empirical evaluation drives information retrieval innovation
- Experimental methodologies should be carefully devised with the scientific method in mind
- Generalization is very important, so parameters should be carefully tuned using held-out data
- Experiments should be reproducible, so experiments should use standard, publicly available data sets and open source IR toolkits whenever possible
- Reference comparisons are critical for convincing yourself and others of the utility of your research
- Detailed analyses often reveal deeper insights into results than average-case analyses

Pointers

- Email:
  - Don Metzler (metzler@google.com)
  - Oren Kurland (kurland@ie.technion.ac.il)
- Slides:
  - http://iew3.technion.ac.il/~kurland/sigir12-tutorial.pdf

Acknowledgments

- We thank Fiana Raiber for running experiments and generating the tables used in this tutorial

Thank you!

References

- Abdul Jaleel, N., Allan, J, Croft, W. B., Diaz, F., Larkey, L., Li, X., Smucker, M., Wade, C. UMass at TREC 2004: Novelty and HARD. TREC, 2004.
- Armstrong, T., Moffat, A., Webber, W., and Zobel, J. Improvements That Don't Add Up: Ad-hoc Retrieval Results Since 1998, CIKM, 2009.
- Azzopardi, L, Girolami, M, van Rijsbergen, C. J. Investigating the Relationship Between Language Model Perplexity and IR Precision-recall Measures. SIGIR, 2003.
- Chapelle, O., Metzler, D., Zhang, Y., and Grinspan, P. Expected Reciprocal Rank for Graded Relevance. CIKM, 2009.
- Cleverdon, C. and Kean, M. Factors Determining the Performance of Indexing Systems. Technical Report, Aslib Cranfield Research Project, 1968.
- Collins-Thompson, K., Callan, J., Terra, E., Clarke, C. The Effect of Document Retrieval Quality on Factoid Question Answering Performance. SIGIR 2004:574-575
- Collins-Thompson, K. Estimating Robust Query Models with Convex Optimization. NIPS, 2008.
- Cormack, G., Smucker, M., Clarke, C. Efficient and Effective Spam Filtering and Re-ranking for Large Web Datasets. Information Retrieval, 2011.
- Dumais, S., Platt, J., Heckerman, D., and Sahami, M. Inductive Learning Algorithms and Representations for Text Categorization. CIKM, 1998.

References

- Harman, D. and Buckley, C. The NRRC Reliable Information Access (RIA) Workshop. SIGIR, 2004.
- Jardine, N, van Rijsbergen, C. J. The Use of Hierarchic Clustering in Information Retrieval. Information Storage and Retrieval, 1971.
- Jarvelin, K and Kekalainen, K. Cumulated Gain-based Evaluation of IR Techniques. TOIS, 2002.
- Joachims, T. Text Categorization with Suport Vector Machines: Learning with Many Relevant Features. ECML, 1998.
- Lavrenko, V. and Croft, W. B. Relevance-Based Language Models. SIGIR, 2001.
- Liu, T-Y. Learning to Rank for Information Retrieval. Foundations and Trends in Information Retrieval, 2009.
- Lv, Y., Zhai, C. Adaptive Relevance Feedback in Information Retrieval. CIKM, 2009.
- Metzler, D. and Croft, W.B. A Markov Random Field Model for Term Dependencies. SIGIR, 2005.
- Metzler, D. and Croft, W. B. Latent Concept Expansion using Markov Random Fields. SIGIR, 2007.
- Moffat, A. and Zobel, J. What Does It Mean to "Measure Performance"?, WISE, 2004.
- Morgan, W., Greiff, W., and Henderson, J. Direct Maximization of Average Precision by Hill-Climbing, with a Comparison to a Maximum Entropy Approach. HLT-NAACL, 2004.

References

- van Rijsbergen, C. J. Information Retrieval. Butterworth, 1979.
- Rocchio, J. Relevance Feedback in Information Retrieval. In Salton: The SMART Retrieval System: Experiments in Automatic Document Processing, 1971.
- Rowe, B., Wood, D., Link, A., and Simoni, D. Economic Impact Assessment of NIST's Text REtrieval Conference (TREC) Program. Technical Report, National Institute of Standards and Technology, 2010.
- Sakai, T., Manabe T., Koyama M. Flexible Pseudo-Relevance Feedback via Selective Sampling. Transactions on Asian Lang. Inf. Process. , 2005.
- Sakai, T. Comparing Metrics Across TREC and NTCIR. The Robustness to System Bias. CIKM, 2008.
- Song, F., Croft, W. B. A General Language Model for Information Retrieval. CIKM, 1999.
- Smucker, M. and Allan, J. A New Measure of the Cluster Hypothesis. ICTIR, 2009.
- Smucker, M., Allan, J., Carterette, B. A Comparison of Statistical Significance for Information Retrieval Evaluation. CIKM, 2009.
- Smucker, M, Allan, J, and Carterette, B. Agreement Among Statistical Significance Tests for Information Retrieval Evaluation at Varying Sample Sizes. SIGIR, 2009.
- Tellex, S., Katz, B., Lin, J., Fernandes, J, and Marton, G. Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering. SIGIR, 2003.
- Voorhees, E. The Cluster Hypothesis Revisited. SIGIR, 1985.
- Voorhees, E. Overview of the TREC 2004 Robust Track. TREC, 2004.
- Yang, Y. and Liu, X. A Re-Examination of Text Categorization Methods. SIGIR, 1999.
- Zhai, C., and Lafferty, J. Model-based Feedback in the Language Modeling Approach to Information Retrieval. CIKM, 2001.