Toward Reproducible Baselines: The Open-Source IR Reproducibility Challenge

Jimmy Lin, **Matt Crane**, Andrew Trotman, Jaime Callan,
Ishan Chattopadhyaya, John Foley, Grant Ingersoll,
Craig Macdonald, and Sebastiano Vigna

Introduction I

- Reproducibility is good
- Strong baselines are good

- How hard can it possibly be?

Introduction II

- “we used <ranking function> as a baseline”
  - Some cases (BM25, QL, …) variants have statistically significantly different results
  - Parameter settings?

Introduction III

- Open-source search engines are a good first step
- However:
    - version?
    - configuration?
    - document cleaning & pre-processing?
    - etc.?

The Challenge I

- Organized as part of RIGOR workshop at SIGIR2015
- Bring developers together to provide reproducible baselines in a common execution environment (Amazon EC2)
- Gather everything necessary into a repository, such that anyone can replicate results by running a single script

The Challenge II

- Long term goals:
    - Understand various aspects of retrieval pipeline (tokenization, document processing, stop words, …) impact effectiveness
    - Understand how different query evaluation strategies impact efficiency

The Challengers I

- Solicited contributions from seven open-source search engines:
    - ATIRE, Galago, Indri, JASS, Lucene, MG4J, Terrier
- Yielding:
    - 13 different indexes
    - 17 different search configurations

Methodology I

- Discussed on a mailing list
- Collection had to be large enough to be interesting, but not so large as to be unwieldy:
  - GOV2, 25M documents, 150 queries for eval

Methodology II

- “Baseline”
  - Depends on techniques being studied
  - Pushed choice to developers with guideline:
    - “If you read a paper that used your system, what would you like to have seen as a baseline?”

Methodology III

- Parameter tuning
- Proper settings critical to effectiveness
- Could not converge on “fair” and feasible given workshop deadline
    - Compromised on “out of the box”
    - What a naive user might use after downloading

Methodology IV

- EC2 instance started and credentials handed out
    - Instance: r3.4xlarge, 16 vCPUs, 122GB ram, Ubuntu Server 14.04 LTS (HVM)
- Configured with union of needed packages, software etc.
- Collection stored on Amazon EBS, mounted at specific location

Methodology V

- Each team agreed on directory structure & naming conventions

- Wrote their script, and committed it to the repo

  - Scripts generally: downloaded, compiled, indexed, searched, printed evaluations

- Repo contains topics, qrels, eval. tools (`trec_eval`)

Methodology VI

- Instance shut down and restarted to match schedules
- Two rounds: first initial results, second fixing issues
- Results were executed on a new “clean” instance by someone not involved in writing the script

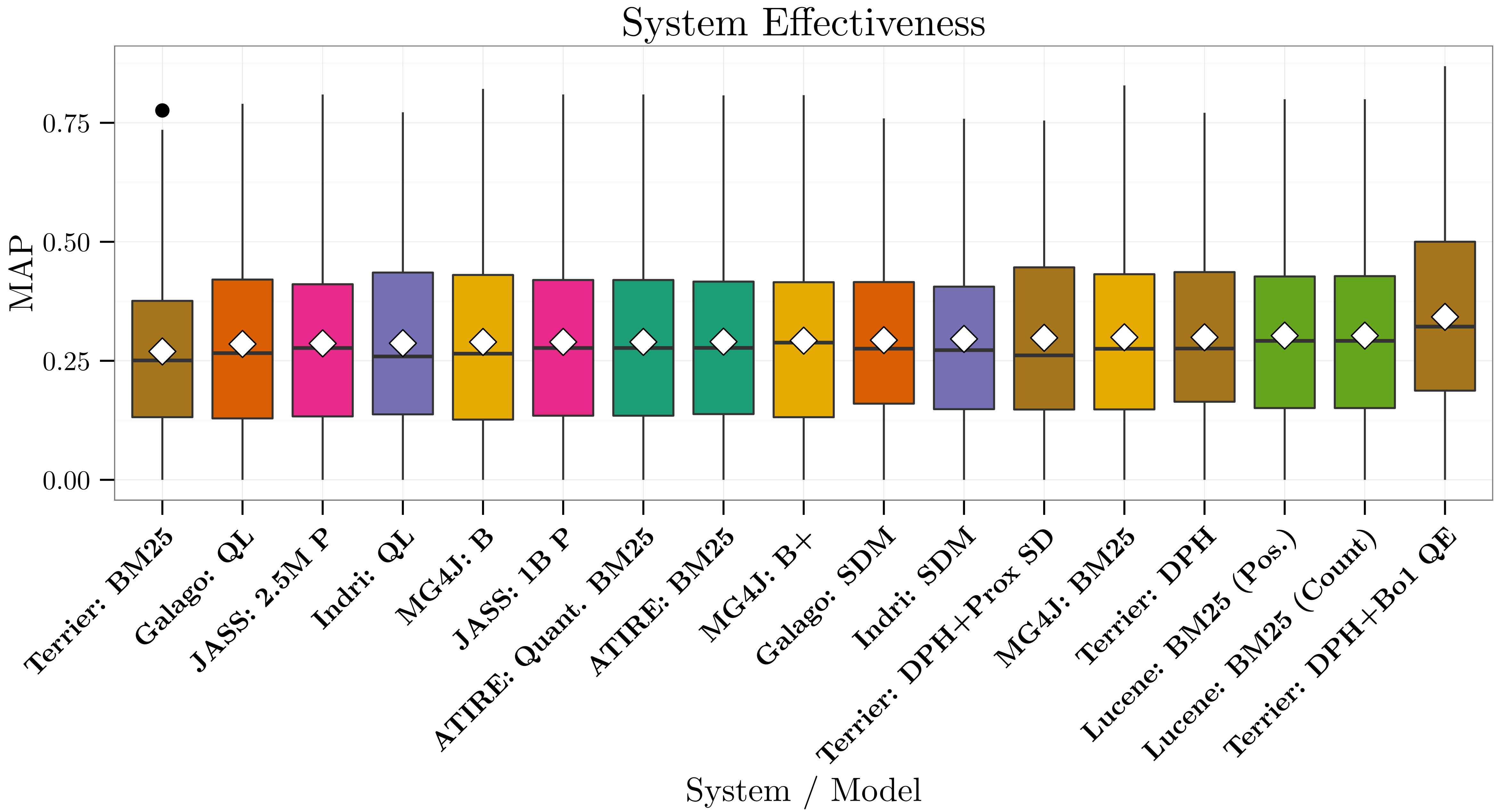
Indexing Results I

System	Type	Size	Time	Threading
ATIRE	Count	12GB	41m	Multi
ATIRE	Count + Quantized	15GB	59m	Multi
Galago	Count	15GB	6h 32m	Multi
Galago	Positions	48GB	26h 23m	Multi
Indri	Positions	92GB	6h 42m	Multi
JASS	ATIRE Quantized	21GB	1h 03m	Multi
Lucene	Count	11GB	1h 36m	Multi
Lucene	Positions	40GB	2h 00m	Multi
MG4J	Count	8GB	1h 46m	Multi
MG4J	Positions	37GB	2h 11m	Multi
Terrier	Count	10GB	8h 06m	Single
Terrier	Count + Direct	18GB	18h 13m	Single
Terrier	Positions	36GB	9h 44m	Single

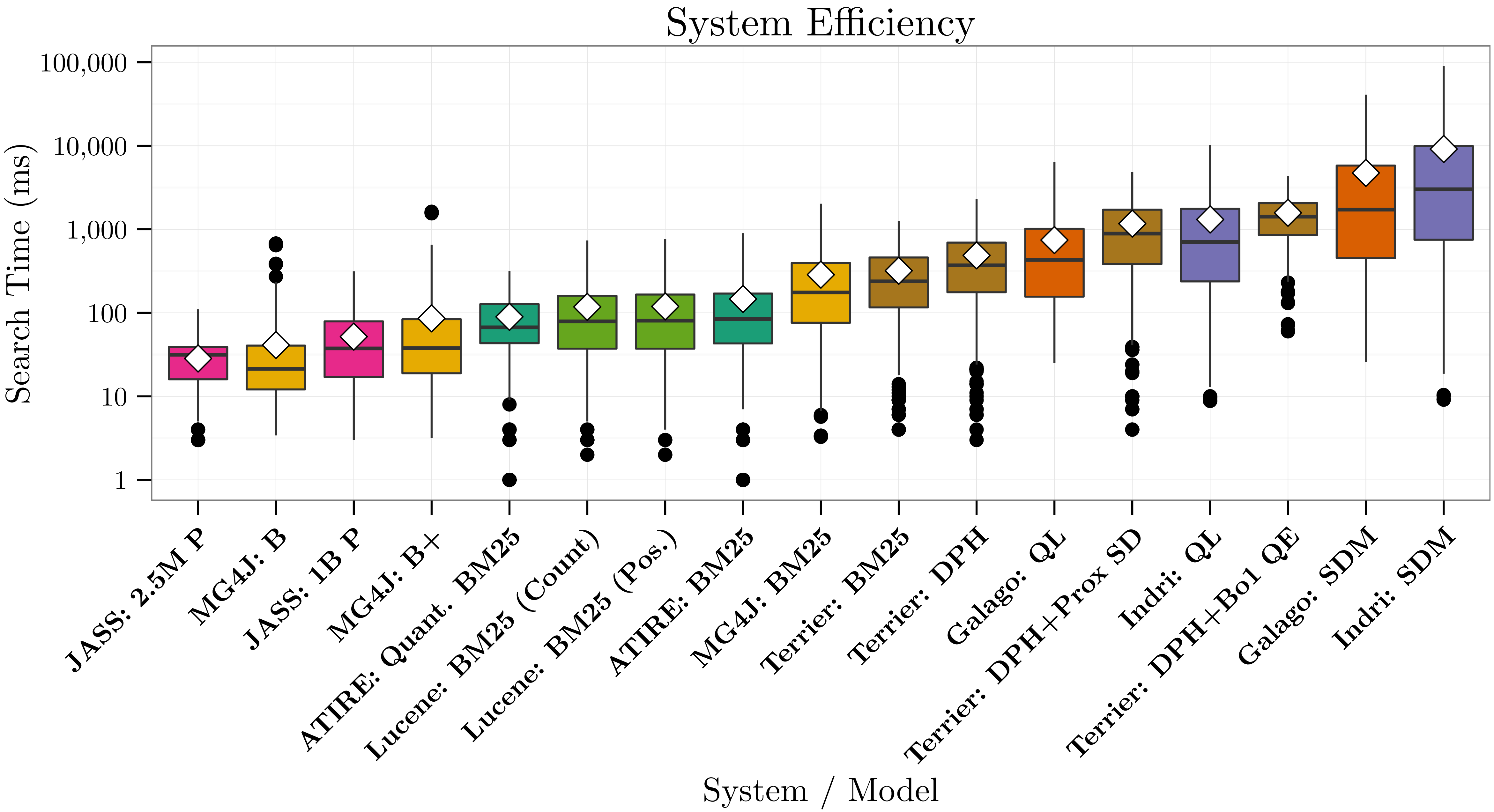
Indexing Results II

System	Type	Terms	Postings	Tokens
ATIRE	Count	39.9M	7.0B	26.5B
ATIRE	Count + Quantized	39.9M	7.0B	26.5B
Galago	Count	36.0M	5.7B	
Galago	Positions	36.0M	5.7B	22.3B
Indri	Positions	39.2M		23.5B
JASS	ATIRE Quantized	39.9M	7.0B	26.5B
Lucene	Count	72.9M	5.5B	
Lucene	Positions	72.9M	5.5B	17.8B
MG4J	Count	34.9M	5.5B	
MG4J	Positions	34.9M	5.5B	23.1B
Terrier	Count	15.3M	4.6B	
Terrier	Count + Direct	15.3M	4.6B	
Terrier	Positions	15.3M	4.6B	16.2B

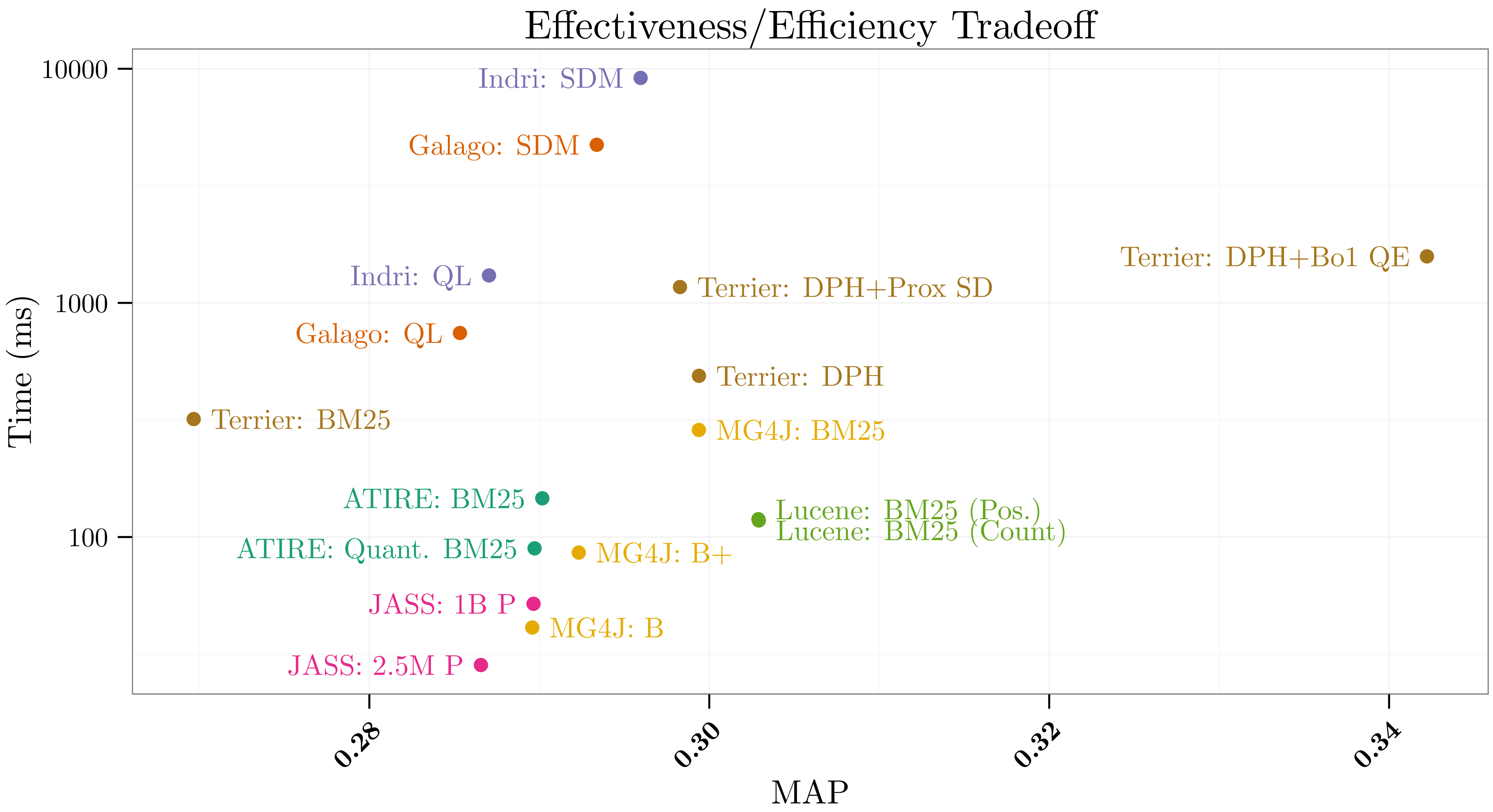
Searching Results I



Searching Results II



Searching Results III



Lessons I

- Challenge was a modest success
- A lot more involved than it would appear, and collective effort much higher than expected
- Global collaboration = difficulty matching schedules
- Surprisingly, for a standard collection it generally took longer than estimated for scripts to be written

Lessons II

- Reproducibility proved more difficult than imagined:
  - A least one case a pre-processing script was required that had never been publicly released
  - At least two cases bugs were exposed in systems that were subsequently fixed
  - EC2 represents a different computing environment than otherwise assumed

Lessons III

- Unintended consequence — serves as a useful teaching resources for students new to information retrieval
- Scripts in repo can serve as an introduction to the basics of working with a test collection

Future I

- Relatively modest maintenance cost
  - Update as new baselines become published
  - Hope sufficient investment in project so far
    - Developers want their systems used “properly”
- We’ll see if we succeed long term

Future II

- Most obvious steps
  - More collections, there are some scripts for ClueWeb09 Cat. B and ClueWeb12 B13
  - More systems, at least two more have made murmurings, others are invited :)

Future III

- Training: from simple parameter tuning, to a complete learning-to-rank setup
  - LTR would provide useful baselines for state of the art in retrieval model
  - Have not yet converted on methodology for “trained” models

Future IV

- External resources
  - Many models take advantage of sources such as anchor text, PageRank, spam score, etc.
  - Some can be derived from the collection
  - Should these resources be included in the repo?
    - Impractical, but introducing external dependencies increases chance of errors

Future V

- Finally, we suspect that much of the differences we observed are down to relatively uninteresting differences (tokenization, stemming, stop words)
- Could create a derived collection that every system ingests to normalize this
- Similar to Buccio *et al.*, but evolved interfaces rather than prescribed
- Perhaps fanciful, but mix and matching different components could greatly accelerate research progress

Conclusion I

- Open-Source IR Reproducibility Challenge represents an ambitious effort to build reproducible baselines for use by the community
- Sincerely encourage participation from the community: both developers contributing additional systems, and adopting our baselines in their work.

Questions?
// Comments

github.com/lintool/IR-Reproducibility