The Information Retrieval Experiment Platform (Extended Abstract)*

Maik Fröbe[1], **Jan Heinrich Reimer**[1], **Sean MacAvaney**[2], **Niklas Deckers**[3,4], **Simon Reich**[3], **Janek Bevendorff**[5], **Benno Stein**[5], **Matthias Hagen**[1] and **Martin Potthast**[4,6,7]

[1]Friedrich-Schiller-Universität Jena
[2]University of Glasgow
[3]Leipzig University
[4]ScaDS.AI
[5]Bauhaus-Universität Weimar
[6]University of Kassel
[7]hessian.AI

Abstract

We have built TIREx, the information retrieval experiment platform, to promote standardized, reproducible, scalable, and blinded retrieval experiments. Standardization is achieved through integration with PyTerrier's interfaces and compatibility with `ir_datasets` and `ir_measures`. Reproducibility and scalability are based on the underlying TIRA framework, which runs dockerized software in a cloud-native execution environment. Using Docker images of 50 standard retrieval approaches, we evaluated all of them on 32 tasks (i.e., 1,600 runs) in less than a week on a midsize cluster (1,620 CPU cores and 24 GPUs), demonstrating multi-task scalability. Importantly, TIRA also enables blind evaluation of AI experiments, as the test data can be hidden from public access and the tested approaches run in a sandbox that prevents data leaks. Keeping the test data hidden from public access ensures that it cannot be used by third parties for LLM training, preventing future training–test leaks.

1 Introduction

Research and development in information retrieval (IR) is predominantly experimental. Conducting an IR experiment usually consists of using alternative retrieval approaches to produce rankings of a document collection (called "runs") for a set of topics. Then, reusable relevance judgments are collected for documents retrieved on high ranks, and approach-specific effectiveness scores are computed [Voorhees, 2001]. This basic experimental setup is known in IR as the Cranfield paradigm [Cleverdon, 1967]. Since its introduction, it has become the de facto standard for laboratory experiments in IR as well as for the organization of shared tasks at dedicated conferences such as TREC [Voorhees, 2019] and beyond. Shared tasks have helped to scale collaborative experimentation, and they have also been widely adopted in AI.

Despite the success of shared tasks, there also are shortcomings: (1) even for tasks with diligently archived code repositories, the results are often not reproducible [Arguello *et al.*, 2015; Lin and Zhang, 2020], (2) run submissions require participants to have access to the test data, which may introduce bias [Fuhr, 2020], and (3) several large language models have been trained, by mistake or deliberately, on publicly available test data [Sainz *et al.*, 2023]. Thus, compared to other disciplines, the current best practices for shared tasks do not enforce "blinded experimentation" [1] with sufficient rigor.

To address these shortcomings, we have developed TIREx, the IR experiment platform. Available open source,[2] TIREx combines tools for working with IR data (`ir_datasets` [MacAvaney *et al.*, 2021]), for executing retrieval pipelines (PyTerrier [Macdonald *et al.*, 2021]), and for evaluating IR systems (`ir_measures` [MacAvaney *et al.*, 2022]), with TIRA [Potthast *et al.*, 2019; Fröbe *et al.*, 2023c], a continuous integration service for reproducible shared tasks and experiments. TIREx implements reproducibility by enabling cloud-native experiments with submitted software in shared tasks, where the workload for shared task organizers is kept comparable to that of traditional shared tasks where only submitted software runs are evaluated. As a proof of concept, we conducted an evaluation of 50 "standard" retrieval approaches on 32 shared tasks (15 datasets with a total of 1.9 billion documents); the 1,600 runs finished in less than a week.

2 Background and Related Work

We review ad hoc retrieval experiments in evaluation campaigns, common problems and pitfalls in IR experiments, best practices for leaderboards, existing reproducibility initiatives, and tools to support reproducibility. Insights from all these areas have influenced our implementation decisions for TIREx.

Ad hoc retrieval experiments in evaluation campaigns. Today's shared task-style experiments for ad hoc retrieval evolved from the Cranfield experiments and aim to produce re-usable test collections [Voorhees, 2019]. Therefore, the

*Invited extended abstract of Fröbe *et al.*'s [2023b] SIGIR best paper.

[1]en.wikipedia.org/wiki/Blinded_experiment

[2]github.com/tira-io/ir-experiment-platform

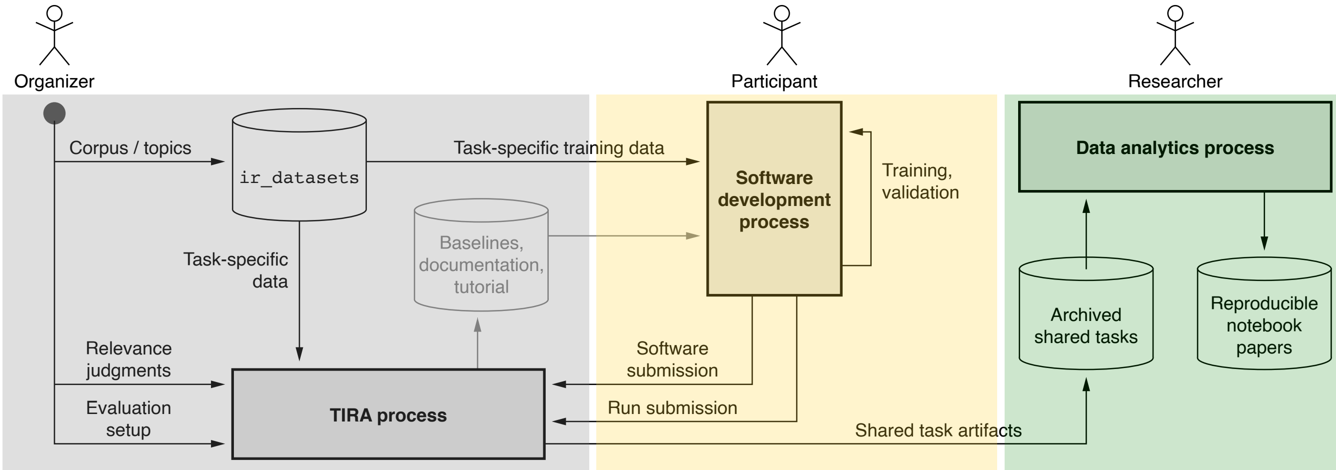

Figure 1: Overview of typical shared task-like IR experiments and how the tools in TIREx support them.

current practice at shared tasks in IR is to assess the relevance of the submitted runs' top-ranked documents, assuming that unjudged documents are non-relevant [Voorhees, 2019], requiring a diverse set of submitted runs pooled at high depth [Voorhees *et al.*, 2022]. Especially for shared tasks that do not attract diverse submissions, TIREx can help to produce a more diverse judgment pool from its dockerized retrieval systems.

Common problems and pitfalls in IR experiments. Even though there is an ongoing discussion on how to conduct IR experiments [Fuhr, 2017; Sakai, 2020; Zobel, 2023; Moffat, 2022], many important characteristics of IR experiments are undisputed. For instance, retrieval studies should be internally valid (conclusions must be supported by the data) and externally valid (repeating an experiment on different but similar data should yield similar observations), where external validity remains an open problem [Fuhr, 2020]. TIREx helps to improve both: the internal validity by archiving all experiments and results, and the external validity by running a submitted software on different data.

Maintaining ongoing leaderboards. Inspired by the observation that many IR studies do not compare against strong baselines [Armstrong *et al.*, 2009b], Armstrong *et al.* [2009a] released EvaluateIR, a public leaderboard for run file submissions. Although the concept was highly valuable for the community to select appropriate baselines, "EvaluateIR never gained traction, and a number of similar efforts following it have also floundered" [Lin, 2018]. Still, certain task-specific leaderboards are quite popular [Zhang *et al.*, 2022; Lin *et al.*, 2022]. Maintaining long-running leaderboards comes with some caveats, as they are conceptually turn-based games where every submission might leak information from the test set [Lin *et al.*, 2022]. With TIREx and its blind evaluation, organizers can choose to blind submissions, supporting the best practices recommended by Lin *et al.* [2022].

Reproducibility initiatives in IR. The IR community makes substantial efforts to foster reproducibility, e.g., with reproducibility tracks at conferences and reproducibility initiatives like OSIRRC [Arguello *et al.*, 2015; Clancy *et al.*, 2019] or CENTRE [Ferro *et al.*, 2018; Ferro *et al.*, 2019; Sakai *et al.*, 2019; Sakai *et al.*, 2020]. Archiving systems for reproducibility is highly challenging, e.g., because external dependencies or platform dependencies might become unavailable. TIREx improves reproducibility because dockerized software is executed in a sandbox (no internet connection), i.e., all dependencies must be already installed.

Tooling for reproducibility. Many tools have been developed to support shared tasks by reducing the workload of organizers and participants while increasing the reproducibility [Yadav *et al.*, 2019; Breuer *et al.*, 2019; Vanschoren *et al.*, 2013; Jagerman *et al.*, 2018; Tsatsaronis *et al.*, 2015; Hopfgartner *et al.*, 2015; Hopfgartner *et al.*, 2018; Fröbe *et al.*, 2023c]. Documentation plays a key role, e.g., with `ir_metadata` [Breuer *et al.*, 2022] implementing the PRIMAD model (platform, research goal, implementation, method, actor, data) [Ferro *et al.*, 2016]. Multiple platforms support organizing and running shared tasks, e.g., CodaLab, EvalAI, STELLA, and TIRA.[3] We use TIRA for TIREx as it supports blinded experimentation based on (private) git repositories hosted on GitLab or GitHub to versionize shared tasks and to distribute the workloads via runners connected to the corresponding repositories.

3 TIREx: The IR Experiment Platform

We have constructed TIREx, the IR experiment platform, to facilitate reproducible, shared task-style IR experiments based on software submissions by integrating `ir_datasets`, `ir_measures`, and PyTerrier into TIRA. IR experiments typically involve intermediate artifacts (like indexes), and retrieval systems involve multi-stage pipelines. Below, we elaborate on how TIREx addresses these requirements and discuss the interaction between integrated tools, provide examples of using available retrieval approaches in TIREx, and demonstrate how TIREx promotes post-experiment replicability and reproducibility through declarative PyTerrier pipelines.

[3] codalab.org, eval.ai, stella-project.org, tira.io

3.1 Experiments in the IR Experiment Platform

As illustrated in Figure 1, TIREx facilitates the entire process of conducting retrieval experiments. It allows shared task organizers and individual experimenters to import data and utilize any pre-existing retrieval software submitted to TIREx as baselines. Following that, submissions of new retrieval approaches for evaluation can be made as software submissions or, if enabled, also as run submissions. To incorporate a new corpus and topics into TIREx, they can be added to `ir_datasets` for automatic import to TIRA. Participants submit their software as Docker images. TIRA ensures their reproducibility and prevents test data leaks by executing them in a sandbox. Among other things, the sandbox disables Internet connectivity for the running software, which ensures that the software and its dependencies are fully installed and no data is sent to unauthorized third parties. Participants can provide additional data their software needs by uploading it to TIRA.

TIREx allows for software submissions to be executed on demand within a cloud-based execution environment, utilizing GitLab or GitHub CI/CD pipelines. In order to meet varying demand, experiment organizers can incorporate additional runners as necessary. TIREx maintains a comprehensive record of every artifact of a retrieval experiment in a specific git repository (Figure 1, right), which can be exported and published, enabling the independent re-execution of approaches with identical or differing data. Consequently, TIREx facilitates "always-on" shared tasks for the IR community, along with an extensive variety of ablation studies.

3.2 Reproducible Shared Tasks with TIRA

Since 2012, TIRA has handled software submissions in shared tasks [Gollub *et al.*, 2012; Potthast *et al.*, 2019]; PAN and Touché are long-running examples.[4] A first version of TIRA did let shared task participants access virtual machines to deploy software. Recently, TIRA was completely redeveloped based on the now industry-standard CI/CD pipelines (continuous integration and deployment) using Git, Docker, and Kubernetes [Fröbe *et al.*, 2023c]. Participants now upload their software implemented in Docker images to a private Docker registry dedicated to their team and the new TIRA runs them on a Kubernetes cluster (1,620 CPU cores, 25.4 TB RAM, 24 GeForce GTX 1080 GPUs); the first users were 71 teams with 647 software submissions in two NLP tasks hosted at SemEval 2023 [Fröbe *et al.*, 2023a; Kiesel *et al.*, 2023]. As previous IR tasks organized in TIRA [Bondarenko *et al.*, 2020; Bondarenko *et al.*, 2021; Bondarenko *et al.*, 2022] were missing standardized data access, yielding non-reusable software submissions, TIRA was also substantially expanded and redeveloped in major parts to integrate `ir_datasets`, `ir_measures`, and PyTerrier.

3.3 Standardized Data Access with `ir_datasets`

The `ir_datasets` toolkit [MacAvaney *et al.*, 2021] provides an interface to access over 200 datasets and over 500 topic sets frequently used in IR experiments. The data is kept up-to-date and processing documents or topics is possible via a single line of Python code. Thus, `ir_datasets` already serves as a common data layer in numerous IR frameworks and tools [Yates *et al.*, 2020; Piwowarski, 2020; Boytsov and Nyberg, 2020; MacAvaney *et al.*, 2020; Costello *et al.*, 2022; Macdonald *et al.*, 2021; Yang *et al.*, 2017; Mallia *et al.*, 2019]. We integrate `ir_datasets` into TIRA via Docker images that can import complete datasets (for full-rank approaches) and that can create re-rankings for any given run file (for re-ranking approaches). We modify `ir_datasets` to include a new field 'default_text' for queries and documents so that the same software can process different datasets.

TIREx aims to support experiments in which components for the individual stages of modularized retrieval pipelines can be easily replaced and compared without having to adapt the complete retrieval software each time. Therefore, TIRA distinguishes between two types of retrieval approaches: (1) full-rank approaches with a document corpus and topics as input, and (2) re-rankers with a re-rank file as input (basically, query–document pairs). From any retrieval software's output, a re-rank file can be automatically created and cached in TIREx by the `ir_datasets` integration. As the structure of these re-rank files always is the same, any re-ranker can easily run on the output of any previous retrieval approach.

3.4 Sanity-checked Evaluation with `ir_measures`

TIRA can automatically evaluate run files (created by software submissions or uploads) via an `ir_measures` evaluator. The evaluator performs sanity checks to test if a run file can be parsed and warns of potential errors. Then, if relevance judgments have been provided, the evaluator derives all specified measures averaged over all queries and per query.

3.5 Reproducible IR Pipelines with TIRA

To improve the efficiency of common IR workflows in TIREx, we redeveloped and extended TIRA's ability to define and run modularized software spanning multiple Docker images. All software in TIRA is immutable so that outputs of one software (e.g., an index) can be cached and reused by another software.

Retrieval software in TIRA can have multiple components that form a sequence similar to UNIX pipes or even a directed acyclic graph (DAG). Each component has a Docker image with a command to be executed and can have none, one, or many preceding components, respectively. Since many different components of a software may use a created artifact like an index, we cache all outputs to make pipelines more efficient (as software submissions are immutable).

3.6 Local Pipeline Reproduction with PyTerrier

When an experiment repository is exported and published by the organizers, by default, the test data is kept private but the run files are published via TIRA and software submissions are uploaded as Docker images to Docker Hub. All possible follow-up studies (e.g., a reproducibility study for a shared task) can be conducted independent of TIRA. To simplify such follow-up studies, we created an PyTerrier integration that allows to re-execute Docker images or inject published outputs (e.g., indices) of software executions in declarative PyTerrier pipelines. Especially the re-use of cached outputs in local pipelines reduces the barrier of entry, because post-

[4] pan.webis.de, touche.webis.de

hoc experiments can build upon outputs of complex software without having to re-execute them.

4 Evaluation

To demonstrate the scalability of TIREx, we conducted an experiment with 50 retrieval approaches on 32 retrieval tasks based on 15 datasets (1.9 billion documents). The resulting leaderboards are public and new submissions can be made at any time.[5] We also describe a `repro_eval`-based [Breuer *et al.*, 2021] case study on system preference reproducibility for different retrieval tasks.

The 15 datasets cover a diverse set of retrieval scenarios, including argument retrieval, general web search, question answering, medical search, news search, etc. (please refer to the original TIREx paper for a full overview of all datasets [Fröbe *et al.*, 2023b]). The 50 retrieval approaches that we imported into TIREx come from 5 retrieval frameworks: BEIR [Thakur *et al.*, 2021], ChatNoir [Bevendorff *et al.*, 2018], Pyserini [Lin *et al.*, 2021], PyGaggle [Lin *et al.*, 2021], PyTerrier [Macdonald *et al.*, 2021]. We ran all retrieval systems on all datasets, see the original TIREx paper for a full evaluation on all datasets [Fröbe *et al.*, 2023b]. Given the reproducibility focus of TIREx, we include a report on a case study on a reproducibility analysis in this extended abstract.

4.1 Case Study: Reproducibility Analysis

As an example of a post-hoc analysis enabled by TIREx, we use `repro_eval` to analyze to which degree system preferences from the TREC Deep Learning 2019 task can be reproduced on other tasks. For each preference between approaches on TREC Deep Learning 2019 (e.g., monoT5 is more effective on TREC DL 2019 than BM25), we set the approach with the lower effectiveness on TREC Deep Learning 2019 as the "baseline" in `repro_eval` and the other approach as the "advanced system". We study the reproducibility of the preferences on two dimensions [Breuer *et al.*, 2020]: (1) the effect ratio of the reproduction, and (2) the delta relative improvement of the reproduction. The effect ratio measures to which degree the advanced system is still better than the baseline on the different task (1 indicates a perfect reproducibility, values between 0 and 1 indicate reproducibility with diminished improvements on the different task, and 0 indicates failed reproducibility), while the delta relative improvement measures the relative effectiveness difference of the advanced system to the baseline (0 indicates perfect reproducibility, values between -1 and 0 indicate an increased relative improvement of the advanced system, values between 0 and 1 indicate a smaller relative improvement, and 1 indicates failed reproducibility).

Table 1 shows the results of the preference reproducibility analysis. Not that surprising, the reproducibility on the very similar TREC Deep Learning 2020 is very good (88.1%) but declines fast for other tasks (e.g., only 57.8% for the Web track 2003 on rank 15). Analyzing the quantiles yields similar observations (e.g., 50% of the system preferences have an almost perfect effect ratio of 0.90 or higher for TREC Deep Learning 2020, while the Web track 2003 on rank 15 has a median effect ratio of 0.04).

[5] github.com/tira-io/ir-experiment-platform#submission

Task	Rank	Succ.	Effect Ratio			Delta Rel. Impr.		
			25%	50%	75%	25%	50%	75%
TREC DL 2020	1	88.1	0.68	0.90	1.11	-0.03	0.02	0.08
Touché 2020	2	77.1	0.12	0.38	0.73	-0.09	0.04	0.17
Web track 2004	3	75.5	0.01	0.29	0.89	-0.07	0.10	0.31
TREC-7	4	73.9	-0.03	0.31	1.11	-0.02	0.12	0.34
Core 2018	5	70.2	-0.05	0.24	0.90	-0.03	0.13	0.35
NFCorpus	10	66.4	-0.06	0.06	0.32	0.02	0.23	0.42
Web track 2003	15	57.8	-0.14	0.04	0.23	-0.08	0.15	0.36
Web track 2009	20	44.1	-0.40	-0.04	0.26	0.00	0.30	0.52
Web track 2010	25	36.3	-0.49	-0.14	0.18	0.03	0.32	0.59
Web track 2013	30	31.0	-0.43	-0.21	0.13	0.06	0.30	0.63

Table 1: Reproducibility of TREC DL 2019 system preferences on other tasks. Success rate in percent (effect ratio > 0; tasks ordered by success rate) and the 25%, 50%, and 75% quantiles for the effect ratio and delta relative improvement.

5 Discussion

We believe that TIREx can have a substantial conceptual impact as we see no alternative to blinded retrieval evaluations in the future (given the practice of training LLMs on basically all available ground truth for IR and NLP tasks [Chung *et al.*, 2022]). Additionally, the platform eases the organization of reproducible IR experiments with software submissions. For shared tasks that run over multiple years on different data, the organizers can automatically re-run approaches submitted to previous editions to track progress. Interesting directions for future development besides including further IR frameworks and libraries are integrations of TIREx with the IR Anthology [Potthast *et al.*, 2021] (e.g., links between entries in the TIREx leaderboards and the corresponding publications) and with DiffIR [Jose *et al.*, 2021] (e.g., rendering runs as search engine result pages to contrast the quantitative evaluations with qualitative evaluations of ranking differences).

6 Conclusion

With TIREx, we aim to substantially ease conducting (blinded) IR experiments and organizing "always-on" reproducible shared tasks on the basis of software submissions. TIREx integrates `ir_datasets`, `ir_measures`, and PyTerrier with TIRA. Retrieval workflows can be executed on-demand via cloud-native orchestration, reducing the effort for reproducing IR experiments since software submitted to TIREx can be re-executed in post-hoc experiments. The platform has no lock-in effect, as archived experiments are fully self-contained, work stand-alone, and are easily exported. By keeping test data private, TIREx promotes further standardization and provenance of IR experiments following the example of, e.g., medicine, where blinded experiments are the norm. TIREx is open to the IR community and ready to include more datasets, shared tasks, and retrieval approaches.

Acknowledgments

This work has been partially supported by the OpenWebSearch.eu project (funded by the EU; GA 101070014).

References

[Arguello *et al.*, 2015] J. Arguello, F. Diaz, J. Lin, and A. Trotman. SIGIR 2015 workshop on reproducibility, inexplicability, and generalizability of results (RIGOR). In *SIGIR 2015*, pages 1147–1148, 2015.

[Armstrong *et al.*, 2009a] T. G. Armstrong, A. Moffat, W. Webber, and J. Zobel. EvaluatIR: An online tool for evaluating and comparing IR systems. In *SIGIR 2009*, page 833, 2009.

[Armstrong *et al.*, 2009b] T. G. Armstrong, A. Moffat, W. Webber, and J. Zobel. Improvements that don't add up: Ad-hoc retrieval results since 1998. In *CIKM 2009*, pages 601–610, 2009.

[Bevendorff *et al.*, 2018] J. Bevendorff, B. Stein, M. Hagen, and M. Potthast. Elastic ChatNoir: Search engine for the ClueWeb and the Common Crawl. In *ECIR 2018*, 2018.

[Bondarenko *et al.*, 2020] A. Bondarenko, M. Fröbe, M. Beloucif, L. Gienapp, Y. Ajjour, A. Panchenko, C. Biemann, B. Stein, H. Wachsmuth, M. Potthast, and M. Hagen. Overview of Touché 2020: Argument retrieval. In *CLEF 2020*, pages 384–395, 2020.

[Bondarenko *et al.*, 2021] A. Bondarenko, L. Gienapp, M. Fröbe, M. Beloucif, Y. Ajjour, A. Panchenko, C. Biemann, B. Stein, H. Wachsmuth, M. Potthast, and M. Hagen. Overview of Touché 2021: Argument retrieval. In *CLEF 2021*, pages 450–467, 2021.

[Bondarenko *et al.*, 2022] A. Bondarenko, M. Fröbe, J. Kiesel, S. Syed, T. Gurcke, M. Beloucif, A. Panchenko, C. Biemann, B. Stein, H. Wachsmuth, M. Potthast, and M. Hagen. Overview of Touché 2022: Argument retrieval. In *CLEF 2022*, 2022.

[Boytsov and Nyberg, 2020] L. Boytsov and E. Nyberg. Flexible retrieval with NMSLIB and FlexNeuART. In *NLP-OSS 2020*, pages 32–43, 2020.

[Breuer *et al.*, 2019] T. Breuer, P. Schaer, N. Tavakolpoursaleh, J. Schaible, B. Wolff, and B. Müller. STELLA: Towards a framework for the reproducibility of online search experiments. In *OSIRRC at SIGIR 2019*, pages 8–11, 2019.

[Breuer *et al.*, 2020] T. Breuer, N. Ferro, N. Fuhr, M. Maistro, T. Sakai, P. Schaer, and I. Soboroff. How to measure the reproducibility of system-oriented IR experiments. In *SIGIR 2020*, pages 349–358, 2020.

[Breuer *et al.*, 2021] T. Breuer, N. Ferro, M. Maistro, and P. Schaer. repro_eval: A python interface to reproducibility measures of system-oriented IR experiments. In *ECIR 2021*, pages 481–486, 2021.

[Breuer *et al.*, 2022] T. Breuer, J. Keller, and P. Schaer. ir_metadata: An extensible metadata schema for IR experiments. In *SIGIR 2022*, pages 3078–3089, 2022.

[Chung *et al.*, 2022] H. W. Chung, L. Hou, S. Longpre, *et al*. Scaling instruction-finetuned language models. arXiv:2210.11416, 2022.

[Clancy *et al.*, 2019] R. Clancy, N. Ferro, C. Hauff, J. Lin, T. Sakai, and Z. Z. Wu. Overview of the 2019 open-source IR replicability challenge (OSIRRC 2019). In *OSIRRC at SIGIR 2019*, pages 1–7, 2019.

[Cleverdon, 1967] C. Cleverdon. The Cranfield tests on index language devices. In *ASLIB Proceedings*, pages 173–192, 1967.

[Costello *et al.*, 2022] C. Costello, E. Yang, D. Lawrie, and J. Mayfield. Patapsco: A python framework for cross-language information retrieval experiments. In *ECIR 2022*, 2022.

[Ferro *et al.*, 2016] N. Ferro, N. Fuhr, K. Järvelin, N. Kando, M. Lippold, and J. Zobel. Increasing reproducibility in IR: Findings from the Dagstuhl seminar on "Reproducibility of data-oriented experiments in e-science". *SIGIR Forum*, 50(1):68–82, 2016.

[Ferro *et al.*, 2018] N. Ferro, M. Maistro, T. Sakai, and I. Soboroff. Overview of Centre@CLEF 2018: A first tale in the systematic reproducibility realm. In *CLEF 2018*, 2018.

[Ferro *et al.*, 2019] N. Ferro, N. Fuhr, M. Maistro, T. Sakai, and I. Soboroff. Overview of Centre@CLEF 2019: Sequel in the systematic reproducibility realm. In *CLEF 2019*, pages 287–300, 2019.

[Fröbe *et al.*, 2023a] M. Fröbe, T. Gollub, B. Stein, M. Hagen, and M. Potthast. SemEval-2023 Task 5: Clickbait spoiling. In *SemEval-2023*, pages 2278–2289, 2023.

[Fröbe *et al.*, 2023b] M. Fröbe, J. H. Reimer, S. MacAvaney, N. Deckers, S. Reich, J. Bevendorff, B. Stein, M. Hagen, and M. Potthast. The information retrieval experiment platform. In *SIGIR 2023*, pages 2826–2836, 2023.

[Fröbe *et al.*, 2023c] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, and M. Potthast. Continuous integration for reproducible shared tasks with TIRA.io. In *ECIR 2023*, 2023.

[Fuhr, 2017] N. Fuhr. Some common mistakes in IR evaluation, and how they can be avoided. *SIGIR Forum*, 51(3):32–41, 2017.

[Fuhr, 2020] N. Fuhr. Proof by experimentation?: Towards better IR research. *SIGIR Forum*, 54(2):2:1–2:4, 2020.

[Gollub *et al.*, 2012] T. Gollub, B. Stein, S. Burrows, and D. Hoppe. TIRA: Configuring, executing, and disseminating information retrieval experiments. In *TIR 2012 at DEXA*, pages 151–155, 2012.

[Hopfgartner *et al.*, 2015] F. Hopfgartner, T. Brodt, J. Seiler, *et al*. Benchmarking news recommendations: The CLEF NewsREEL use case. *SIGIR Forum*, 49(2):129–136, 2015.

[Hopfgartner *et al.*, 2018] F. Hopfgartner, A. Hanbury, H. Müller, *et al*. Evaluation-as-a-service for the computational sciences: Overview and outlook. *Journal of Data and Information Quality*, 10(4):15:1–15:32, 2018.

[Jagerman *et al.*, 2018] R. Jagerman, K. Balog, and M. de Rijke. OpenSearch: Lessons learned from an online evaluation campaign. *Journal of Data and Information Quality*, 10(3):13:1–13:15, 2018.

[Jose *et al.*, 2021] K. M. Jose, T. Nguyen, S. MacAvaney, J. Dalton, and A. Yates. DiffIR: Exploring differences in ranking models' behavior. In *SIGIR 2021*, pages 2595–2599, 2021.

[Kiesel *et al.*, 2023] J. Kiesel, M. Alshomary, N. Mirzakhmedova, M. Heinrich, N. Handke, H. Wachsmuth, and B. Stein. SemEval-2023 Task 4: ValueEval: Identification of human values behind arguments. In *SemEval 2023*, pages 2290–2306, 2023.

[Lin and Zhang, 2020] J. Lin and Q. Zhang. Reproducibility is a process, not an achievement: The replicability of IR reproducibility experiments. In *ECIR 2020*, pages 43–49, 2020.

[Lin *et al.*, 2021] J. Lin, X. Ma, S. Lin, J. Yang, R. Pradeep, and R. Nogueira. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *SIGIR 2021*, pages 2356–2362, 2021.

[Lin *et al.*, 2022] J. Lin, D. Campos, N. Craswell, B. Mitra, and E. Yilmaz. Fostering coopetition while plugging leaks: The design and implementation of the MS MARCO leaderboards. In *SIGIR 2022*, pages 2939–2948, 2022.

[Lin, 2018] J. Lin. The neural hype and comparisons against weak baselines. *SIGIR Forum*, 52(2):40–51, 2018.

[MacAvaney *et al.*, 2020] S. MacAvaney, A. Yates, S. Feldman, D. Downey, A. Cohan, and N. Goharian. OpenNIR: A complete neural ad-hoc ranking pipeline. In *WSDM 2020*, pages 845–848, 2020.

[MacAvaney *et al.*, 2021] S. MacAvaney, A. Yates, S. Feldman, D. Downey, A. Cohan, and N. Goharian. Simplified data wrangling with ir_datasets. In *SIGIR 2021*, pages 2429–2436, 2021.

[MacAvaney *et al.*, 2022] S. MacAvaney, C. Macdonald, and I. Ounis. Streamlining evaluation with ir-measures. In *ECIR 2022*, pages 305–310, 2022.

[Macdonald *et al.*, 2021] C. Macdonald, N. Tonellotto, S. MacAvaney, and I. Ounis. Pyterrier: Declarative experimentation in python from BM25 to dense retrieval. In *CIKM 2021*, pages 4526–4533, 2021.

[Mallia *et al.*, 2019] A. Mallia, M. Siedlaczek, J. M. Mackenzie, and T. Suel. PISA: Performant indexes and search for academia. In *OSIRRC at SIGIR 2019*, pages 50–56, 2019.

[Moffat, 2022] A. Moffat. Batch evaluation metrics in information retrieval: Measures, scales, and meaning. *IEEE Access*, 10:105564–105577, 2022.

[Piwowarski, 2020] B. Piwowarski. Experimaestro and Datamaestro: Experiment and dataset managers (for IR). In *SIGIR 2020*, pages 2173–2176, 2020.

[Potthast *et al.*, 2019] M. Potthast, T. Gollub, M. Wiegmann, and B. Stein. TIRA integrated research architecture. In *Information Retrieval Evaluation in a Changing World*. 2019.

[Potthast *et al.*, 2021] M. Potthast, S. Günther, J. Bevendorff, J. P. Bittner, A. Bondarenko, M. Fröbe, C. Kahmann, A. Niekler, M. Völske, B. Stein, and M. Hagen. The information retrieval anthology. In *SIGIR 2021*, pages 2550–2555, 2021.

[Sainz *et al.*, 2023] O. Sainz, J. A. Campos, I. García-Ferrero, *et al*. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *EMNLP 2023*, pages 10776–10787, 2023.

[Sakai *et al.*, 2019] T. Sakai, N. Ferro, I. Soboroff, Z. Zeng, P. Xiao, and M. Maistro. Overview of the NTCIR-14 Centre task. In *NTCIR 2019*, 2019.

[Sakai *et al.*, 2020] T. Sakai, S. Tao, Z. Zeng, Y. Zheng, J. Mao, Z. Chu, Y. Liu, M. Maistro, Z. Dou, N. Ferro, et al. Overview of the NTCIR-15 We Want Web with Centre (WWW-3) task. *NTCIR 2020*, 2020.

[Sakai, 2020] T. Sakai. On Fuhr's guideline for IR evaluation. *SIGIR Forum*, 54(1):12:1–12:8, 2020.

[Thakur *et al.*, 2021] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *NeurIPS Datasets and Benchmarks 2021*, 2021.

[Tsatsaronis *et al.*, 2015] G. Tsatsaronis, G. Balikas, P. Malakasiotis, *et al*. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16:138:1–138:28, 2015.

[Vanschoren *et al.*, 2013] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo. OpenML: Networked science in machine learning. *SIGKDD Explor.*, 15(2):49–60, 2013.

[Voorhees *et al.*, 2022] E. M. Voorhees, I. Soboroff, and J. Lin. Can old TREC collections reliably evaluate modern neural retrieval models? arXiv:2201.11086, 2022.

[Voorhees, 2001] E. M. Voorhees. The philosophy of information retrieval evaluation. In *CLEF 2001*, pages 355–370, 2001.

[Voorhees, 2019] E. M. Voorhees. The evolution of Cranfield. In *Information Retrieval Evaluation in a Changing World*, pages 45–69. 2019.

[Yadav *et al.*, 2019] D. Yadav, R. Jain, H. Agrawal, *et al*. EvalAI: Towards better evaluation systems for AI agents. arXiv:1902.03570, 2019.

[Yang *et al.*, 2017] P. Yang, H. Fang, and J. Lin. Anserini: Enabling the use of Lucene for information retrieval research. In *SIGIR 2017*, pages 1253–1256, 2017.

[Yates *et al.*, 2020] A. Yates, S. Arora, X. Zhang, W. Yang, K. M. Jose, and J. Lin. Capreolus: A toolkit for end-to-end neural ad hoc retrieval. In *WSDM 2020*, pages 861–864, 2020.

[Zhang *et al.*, 2022] X. Zhang, N. Thakur, O. Ogundepo, *et al*. Making a MIRACL: Multilingual information retrieval across a continuum of languages. arXiv:2210.09984, 2022.

[Zobel, 2023] J. Zobel. When measurement misleads: The limits of batch assessment of retrieval systems. In *ACM SIGIR Forum*, volume 56, pages 1–20, 2023.