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 Potthast4,6,7

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 build The Information Retrieval Experiment Platform (TIREx) to promote more 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 Integrated Research Architecture, which runs dockerized software in a cloud-native execution environment. This enables blind evaluation, since the test data and ground truth are hidden from public access, and the software processes the data in a sandbox that prevents data leaks. More generally, TIRA facilitates AI experiments on test data that cannot be used for LLM training, preventing training–test leaks. Using Docker images of 50 standard retrieval approaches, we evaluated all of them on 32 tasks (totaling 1,600 runs) in less than a week on a midsize cluster (1,620 CPU cores and 24 GPUs), demonstrating multi-task scalability.

1 Introduction

Research and development in information retrieval (IR) has been predominantly experimental. The so-called Cranfield paradigm [Cleverdon, 1967] is the de facto standard for experiments in IR and shared tasks hosted at TREC [Voorhees, 2019] and beyond. An experiment (a shared task) consists of a document corpus for which retrieval approaches are implemented, and topics for which the approaches are run to produce document rankings (so-called “runs”). The runs are then pooled and (reusable) relevance judgments are gathered to calculate the evaluation scores [Voorhees, 2001]. Organizing retrieval experiments as shared tasks allowed the community to scale up collaborative laboratory experiments. Shared tasks of this kind, or leaderboards for benchmarks, have also become the de facto standard for evaluating AI approaches in general.

Despite the lasting success of shared tasks in computer science, they also have shortcomings: (1) Even for approaches with diligently archived code repositories, they are often found not to be reproducible [Arguello et al., 2015; Lin and Zhang, 2020]. (2) The typical submission of runs requires participants to have access to the test data, which may, for instance, introduce bias [Fuhr, 2020a]. (3) It has been shown that well-known large language models have been trained, by mistake or deliberately, on publicly available test data (and its ground truth after a shared task) [Sainz et al., 2023]. The current best practices for shared tasks do not enforce “blinded experimentation” with sufficient rigor, compared to other empirical disciplines.

To address all of these shortcomings, we have developed the IR Experiment Platform (TIREx; cf. Figure 1 for an overview). Available as open source, a key feature of TIREx is the full integration of 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 is designed for reproducibility through software submissions while keeping an experimenter’s or task organizer’s workload comparable to run file submissions. As a proof of concept, we conducted a large-scale evaluation of 50 “standard” retrieval approaches on 32 shared retrieval tasks (15 corpora with a total of 1.9 billion documents). This experiment consists of 1,600 runs and 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 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

---

*Invited abstract of our SIGIR 2023 best paper [Fröbe et al., 2023b].
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], there is a consensus on many important characteristics of IR experiments. For instance, it is rather undisputed that 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) [Fuhr, 2020b]. Still, external validity of IR experiments remains an open problem [Fuhr, 2020a]. TIREx can help to further improve both: the internal validity via archiving all experiments and results, and the external validity via simplifying to run 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; cla, 2019] or CENTRE [Ferro et al., 2018, 2019; Sakai et al., 2019, 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, 2018; Fröbe et al., 2023c]. Documentation plays a key role, e.g., with ir_metadata [Breuer et al., 2022] implementing the PRIMAD model [Ferro et al., 2016] (platform, research goal, implementation, method, actor, data). Multiple platforms support organizing and running shared tasks, e.g., CodaLab, EvalAI, STELLA, and TIRA. 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 the IR Experiment Platform (TIREx) 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 address 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.

3.1 **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

---

3 codalab.org, eval.ai, stella-project.org, tira.io
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 \texttt{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

TIRA has been used to handle software submissions in shared tasks since 2012 [Gollub et al., 2012; Potthast et al., 2019]—the CLEF labs PAN and Touche being two long-running examples. A first version of TIRA provided participants with access to virtual machines to deploy their software. Meanwhile, Docker has gained maturity and widespread adoption and is now supported by many cluster computing frameworks such as Kubernetes. Hence, 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]. In the new version of TIRA, participants upload their software implemented in Docker images to a private Docker registry dedicated to their team. For on-demand execution, TIRA presently runs the software on our Kubernetes cluster (1,620 CPU cores, 25.4 TB RAM, 24 GeForce GTX 1080 GPUs). This version of TIRA was first used in two NLP tasks hosted at SemEval 2023, to which 71 of 170 registered teams submitted 647 runs based on software submissions [Fröbe et al., 2023a; Kiesel et al., 2023]. Still, previous IR tasks organized in TIRA [Bondarenko et al., 2020, 2021, 2022] were missing standardized data access, causing that software submissions could not be re-used, hence, we substantially expanded TIRA and redeveloped major parts to integrate \texttt{ir\_datasets}, \texttt{ir\_measures}, and PyTerrier.

3.3 Standardized Data Access with \texttt{ir\_datasets}

The \texttt{ir\_datasets} toolkit [MacAvaney et al., 2021] provides a standard interface to access over 200 corpora 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, \texttt{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 \texttt{ir\_datasets} into TIRA via Docker images that can import complete corpora (for full-rank approaches) and that can create re-rankings for any given run file (for re-ranking approaches). We modify \texttt{ir\_datasets} to include a new ‘default\_text’ field for queries and documents so that the same software can run on many 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 \texttt{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 \texttt{ir\_measures}

TIRA can automatically evaluate run files (created by software submissions or uploads) via an \texttt{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 a 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-
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 corpora (1.9 billion documents). The resulting leaderboards are public and new submissions can be made at any time. We also describe a repro_eval-based [Breuer et al., 2021] case study on system preference reproducibility for different retrieval tasks.

The 15 corpora 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], PyGagger [Lin et al., 2021], PyTerrier [Macdonald et al., 2021]. We ran all retrieval systems on all corpora, 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 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 corpora, shared tasks, and retrieval approaches.

Acknowledgements

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

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.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TREC DL 2020</td>
<td>1</td>
<td>88.1</td>
<td>0.68</td>
<td>0.90 1.11 -0.03 0.02 0.08</td>
</tr>
<tr>
<td>Touché 2020</td>
<td>2</td>
<td>77.1</td>
<td>0.12</td>
<td>0.38 0.73 -0.09 0.04 0.17</td>
</tr>
<tr>
<td>Web track 2004</td>
<td>3</td>
<td>75.5</td>
<td>0.01</td>
<td>0.29 0.89 -0.07 0.10 0.31</td>
</tr>
<tr>
<td>TREC-7</td>
<td>4</td>
<td>73.9</td>
<td>-0.03</td>
<td>0.31 1.11 -0.02 0.12 0.34</td>
</tr>
<tr>
<td>Core 2018</td>
<td>5</td>
<td>70.2</td>
<td>-0.05</td>
<td>0.24 0.90 -0.03 0.13 0.35</td>
</tr>
<tr>
<td>NFCorpus</td>
<td>10</td>
<td>66.4</td>
<td>-0.06</td>
<td>0.06 0.32 0.02 0.23 0.42</td>
</tr>
<tr>
<td>Web track 2003</td>
<td>15</td>
<td>57.8</td>
<td>-0.14</td>
<td>0.04 0.23 -0.08 0.15 0.36</td>
</tr>
<tr>
<td>Web track 2009</td>
<td>20</td>
<td>44.1</td>
<td>-0.40</td>
<td>-0.04 0.26 0.00 0.30 0.52</td>
</tr>
<tr>
<td>Web track 2010</td>
<td>25</td>
<td>36.3</td>
<td>-0.49</td>
<td>-0.14 0.18 0.03 0.32 0.59</td>
</tr>
<tr>
<td>Web track 2013</td>
<td>30</td>
<td>31.0</td>
<td>-0.43</td>
<td>-0.21 0.13 0.06 0.30 0.63</td>
</tr>
</tbody>
</table>

3 github.com/tira-io/ir-experiment-platform#submission
References


