The Information Retrieval Experiment Platform

Extended Abstract

Maik Fröbe¹, Jan Heinrich Reimer², Sean MacAvaney³, Niklas Deckers⁴, Janek Bevendorff⁵, Benno Stein⁶, Matthias Hagen⁷ and Martin Potthast⁸

¹Friedrich-Schiller-Universität Jena
²Leipzig University and ScaDS.AI
³Bauhaus-Universität Weimar

Abstract
In this extended abstract,¹ we present the Information Retrieval Experiment Platform (TIREx) that integrates ir_datasets, ir_measures, PyTerrier, and TIRA for standardized, reproducible, collaborative, scalable, and blinded experiments in IR. Information retrieval experiments face potential problems concerning (1) internal validity, (2) external validity, (3) leakage by large pre-trained models, and (4) a high barrier of entry to built on top of state-of-the-art approaches. TIREx aims to support IR experiments to mitigate those issues. We will focus our talk on collaborations enabled by TIREx that lower the barrier to entry for students to shared tasks in IR.

1. Introduction

Research and development in information retrieval (IR) has been predominantly experimental. In its early days in the 1960s, the IR community saw the need to develop and validate experimental procedures, giving rise to the Cranfield paradigm [2], which became the de facto standard for shared tasks hosted at TREC [3] and many spin-off evaluations. Organizers of typical IR-shared tasks provide a task definition, a document collection, and topics. Participants implement retrieval approaches for the task, and run them on the topics. They then submit the resulting document rankings (“runs”) to the organizers. The task organizers pool all submitted runs, and evaluate them [4], to produce a reusable set of relevance assessments. Finally, participants share a written description of their runs (a “notebook” paper) to disseminate their methodology and findings. This division of labor allowed the community to scale up collaborative experiments. With many research labs working independently on the same task, the community descends on a “wisdom of the crowd”, while ensuring a rigorous comparative evaluation.

¹Condensed version of a resource paper at SIGIR 2023 [1]

© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
Despite their lasting success, this way of organizing shared tasks also has shortcomings. First, as with many other disciplines in computer science and beyond, approaches described in a given notebook underlying a given run submission might not be reproducible. There are well-documented cases where reproductions failed, despite putting much effort into it, even for approaches with diligently archived code repositories [5, 6]. Second, run submissions require that participants have access to the test data, which has severe implications [7], such as informing (biasing) the research hypothesis or approach, unless researchers make a point of not looking at the test topics during development. Third, it cannot be ruled out that not a single one of future large language models has been trained, by mistake or deliberately, on publicly available test data to maximize its ability, or that a usage warning that states not to use the data for training would go unnoticed. In any case, the current best practices for shared tasks do not enforce “blinded experimentation” with sufficient rigor, compared to other empirical disciplines.

We develop the IR Experiment Platform [1] to address those problems. Its key features include full integration of open source tools for working with IR data (ir_datasets [8]), for executing retrieval pipelines (PyTerrier [9]), and for evaluating IR systems (ir_measures [10]) with TIRA [11], a continuous integration platform for reproducible shared tasks. The IR Experiment Platform aims to promote the standardization of IR experiments and to enable the submission of working software rather than runs.

2. New Perspectives on Cooperations and Teaching Initiatives

The main focus of the presentation at the FGIR workshop will be on how we can lower the barrier of entry to shared tasks and on how TIREx can promote new types of cooperation among participants of shared tasks. We specifically want to address teaching initiatives where students participate in shared tasks as part of their coursework. One main difficulty in this setting is that organizers become part of the debug cycle. To mitigate this problem, we aim to reduce the gap between the development environment and the submission environment. For Docker submissions, we often observed that participants dockerized their software only as an afterthought (the instructions to set up the development environment did not match the setup in the Docker image; submitted images often had missing libraries or wrong versions), and we currently try to address this by promoting procedures where participants directly develop in the docker image that they will submit (e.g., via dev-containers). Furthermore, TIREx allows to execute pipelines where the output of one pipeline component can be used as input to subsequent components. These components can be software submissions or manual data uploads. Cooperations enabled by this are, for instance, if one team creates user query variants that can be used as additional input in pipelines by other teams. We also envision similar cooperations for standard components of retrieval pipelines. For instance, the team behind the query performance prediction toolkit qpptk [12] currently dockerizes their framework so that it can be used as a pipeline component in TIREx, which enables other teams to directly use query performance prediction without the need to learn a new framework. We are currently in the process of advertising this type of cooperation and try to get more such retrieval components into TIREx (e.g., the Splade [13] and REL [14] teams agreed to submit their systems to TIREx). TIREx supports caching of results, so each component is executed only once, aiming at GreenIR [15].
Acknowledgments

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

References


