Resources for Combining Teaching and Research in Information Retrieval Courses

Maik Fröbe
Friedrich-Schiller-Universität Jena
Jena, Germany

Harrisen Scells
Leipzig University
Leipzig, Germany

Theresa Elstner
Leipzig University
Leipzig, Germany

Christopher Akiki
Leipzig University
Leipzig, Germany

Lukas Gienapp
Leipzig University
Leipzig, Germany

Jan Heinrich Reimer
Friedrich-Schiller-Universität Jena
Jena, Germany

Sean MacAvaney
University of Glasgow
Glasgow, United Kingdom

Benno Stein
Bauhaus-Universität Weimar
Weimar, Germany

Matthias Hagen
Friedrich-Schiller-Universität Jena
Jena, Germany

Martin Potthast
University of Kassel, hessian.AI, and ScaDS.AI
Kassel, Germany

ABSTRACT

A recent study showed that students in IR courses are especially motivated and learn more effectively when they participate in shared tasks as part of their coursework. To support teachers in integrating such activities, we present Web IDE-based applications and tutorials that employ TIREx and `ir_datasets` to cover the process of a typical shared task in IR: from creating test collections over developing retrieval systems to making relevance judgments and finally statistically analyzing the results. Using our tools, students can gain hands-on experience with empirical IR research by working on some current shared task, by working on earlier collections, or by creating new ones. Our experiences in implementing the corresponding teaching concept in four IR courses at two universities confirm that students are very motivated to conduct research, and we also find that some of the resulting artifacts (e.g., students' test collections and retrieval approaches) are of really good quality.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; • **Education**;

KEYWORDS

Teaching IR, Shared tasks, Test collections, Retrieval evaluation

ACM Reference Format:
Maik Fröbe, Harrisen Scells, Theresa Elstner, Christopher Akiki, Lukas Gienapp, Jan Heinrich Reimer, Sean MacAvaney, Benno Stein, Matthias Hagen, and Martin Potthast. 2024. Resources for Combining Teaching and Research in Information Retrieval Courses. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3626772.3657886

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657886

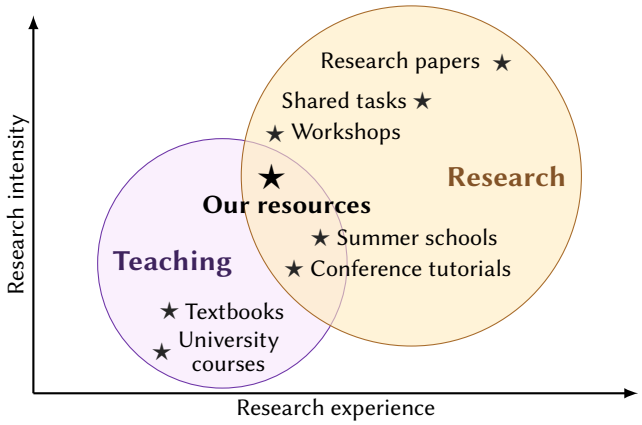


Figure 1: Conceptual overview of how our new resources complement existing teaching or research resources / events.

1 INTRODUCTION

Wilhelm von Humboldt, the "father of the modern university" [60], argued that good teaching and cutting-edge research go hand in hand [58]. As a lot of the research in information retrieval (IR) is empirical and depends on test collections or shared tasks at evaluation campaigns like CLEF, FIRE, NTCIR, or TREC [14, 65], two previous ideas on integrating research and teaching suggested to let students construct test collections as part of their coursework [62] or to let students participate in shared tasks [19]. Still, only a few IR courses adopted such ideas. Potential entry barriers may be that teachers deem research-oriented retrieval software too "difficult" for students and that shared task deadlines often do not align with course schedules. Therefore, the field of information retrieval is still far from Humboldt's Ideal of integrating teaching and research.

In this paper, we present resources[1] that support IR course instructors who want to integrate empirical IR research in their teaching. Our individual components reduce the entry barriers

[1] http://github.com/tira-io/teaching-ir-with-shared-tasks

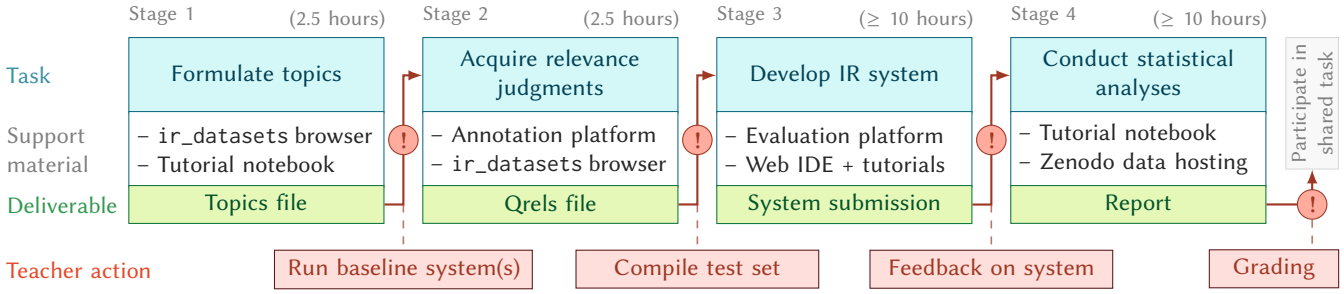


Figure 2: The course workflow supported by our resources with estimated student effort, deliverables, and actions of teachers, organized into four stages: (1) topic formulation, (2) relevance judgments, (3) system development, and (4) statistical analyses.

for shared task participation with students, and ease constructing and assessing IR test collections—the assessment being particularly important to verify the quality of student-created collections [55, 62]. When combining our resources' components for collection construction with the components for system development in TIRA [25] / TIREx [24], an IR course can even complement the research landscape (shown in Figure 1) by internally mimicking a shared task setup that reduces dependencies to external deadlines.

Our resources include tutorials (Section 3.4), Zenodo-hosted datasets (Section 3.5), a Python API unifying access to the datasets and to components submitted to TIREx (Section 3.6), and customizable web applications that guide students and teachers while providing easy access to documents, topics, relevance judgments, and runs (Section 3.2). The overall intended course workflow has four stages with teacher actions as milestones (Figure 2): topic formulation (Stage 1), relevance judgments (Stage 2), system development (Stage 3), and statistical analyses (Stage 4); afterwards, participation in a fitting shared task is possible. In Stage 3, the students should formulate a research question / hypothesis to encourage focusing their system development on insights and not on leaderboard chasing [9]. The developed systems can be submitted to TIREx [24] to test the hypotheses. In TIREx, system runs usually are blinded until the teacher unblinds them to let the students access their results and test their hypothesis. Overall, the process covers the typical steps of experimental IR research and our respective tutorials guide students so that they learn from and motivate each other.

The tutorials are implemented in the dev container standard[2] that enables development in prepared Docker images with all requirements pre-installed. GitHub Codespaces[3] automatically boot into dev containers, so students can work instantly in the cloud without installation. We streamline this process with a new code submission feature in TIRA (Section 3.7) that creates prepared and correctly configured Git repositories, including a GitHub Actions workflow that builds a Docker image from the submission and uploads the image to TIREx with all metadata, including the repository, branch, and commit to cover the full reproducibility cycle. Compared to Google Colab, this has the advantage that teachers have access to the Git repository and that professional versioning simplifies teamwork. The web applications that guide through the tutorials are hosted on GitHub Pages and Zenodo to ensure high robustness and availability (Section 3.2).

[2]https://containers.dev/
[3]https://github.com/features/codespaces

We have instantiated the workflow in four courses at two universities and report on our experiences in Section 4. Conceptually, our teaching approach can enrich shared tasks—traditionally focused on improving effectiveness [56]—by studying aspects of the corpus creation in collaborative teaching environments. Still, our resources are also usable without the intention to participate in a shared task.

2 RELATED WORK

Bauer et al. [8] argue that improving teaching improves future research. We review research in this space, covering tools and demos to support teaching in IR, curricula for IR courses, including lab projects, shared tasks, tutorials, and building test collections.

Tools and Demos to Support Teaching in IR. The IR community has a strong background in creating tools and demos to support teaching. Trotman and Lilly [61] developed JASSjr, a retrieval system for teaching that acts as a baseline and reference against which students compare their implementations during a course. Pyserini [37] and PyTerrier [43] have also been used in teaching university and summer school courses [70]. Macdonald et al. [45] reported higher student satisfaction, engagement, and attainment when switching from "compile-and-execute" experimentation to interactive notebooks. This development-oriented teaching concept can be complemented by easy-to-use web applications, e.g., Wilhelm-Stein et al. [68] proposed a gamified teaching environment for IR courses. We build upon those approaches by combining interactive notebooks with easy-to-use web applications that also link to previous resources and extend the scope (previously on retrieval algorithms and experimentation) to cover the complete spectrum of experimental IR research, especially integrating dataset construction.

Information Retrieval Curriculum. Discussions about the design and the learning objectives of IR courses have been ongoing for more than a decade. For instance, Fernández-Luna et al. [23] proposed to categorize IR courses on a spectrum from system-oriented (e.g., focused on computer science) to user-oriented (e.g., focused on psychology and behavior) with three main education goals: (1) knowledge of IR foundations, (2) training in search formulation, and (3) knowledge of IR processes and components. Blank et al. [10] then developed a detailed course outline from the survey by tailoring the content to four groups: IR system user, management, administration, and development. At the same time, Thornley [59] highlighted that IR has unique opportunities allowing students to work in problem-based learning frameworks on issues relevant to

the real world. Later, Markov and de Rijke [46] surveyed five of the most widely used IR textbooks to determine the types of content that likely is taught in courses that use the textbooks.

The general theme of all these studies is that (1) students should learn about a spectrum of technical and non-technical IR content (i.e., covering many areas of IR), (2) different groups of students should learn about IR (i.e., not just computer science, but also library science, digital humanities, etc.), and (3) the course content should allow students to build and interact with IR systems (i.e., letting students implement IR components). López-García and Cacheda [38] examplarily design course projects that cover these three themes of teaching IR—which inspired our resources.

Supporting students in practical projects inevitably involves programming, preferably in interactive Python notebooks [45]. However, collecting, maintaining, and enabling comparisons of practical projects is challenging, as only recently highlighted by ActivePapers [33] that aimed to improve reproducibility:[4]

> As of 2024, this project is archived and unmaintained. While it has achieved its mission of demonstrating that unifying computational reproducibility and provenance tracking is doable and useful, it has also demonstrated that Python is not a suitable platform to build on for reproducible research. Breaking changes at all layers of the software stack are too frequent.

Following their learnings and recommendations, we center our resources around Docker images that run in a Web IDE to support the longevity of the resources and simplified submissions to TIREx.

Tutorials for IR. Scientific conferences in IR often include tutorials and workshops. Macdonald et al. [44] ran several tutorials to demonstrate how to use IR components from a fundamental level all the way to state-of-the-art retrieval approaches, using declarative PyTerrier pipelines [45].[5] Meanwhile, Azzopardi et al. [5] ran a workshop that gave an in-depth overview of how to use Lucene for IR research, focusing on connecting the industry usage of Lucene with its use in academia. Our focus is instead on university-level teaching, especially connected to shared tasks for which tutorials rarely exist, so we have to reduce the assumed prior knowledge for the tutorials to a minimum. Our vision, however, is to link existing tutorials/resources as a follow-up to our beginner-level material, e.g., that interested students can subsequently continue with research conference tutorials, such as the PyTerrier tutorials.

Teaching with Shared Tasks. We focus our efforts on teaching IR through the lens of shared tasks, which can be especially motivating for students [19]. Shared tasks provide a test collection and information needs for which teams develop systems that are pooled for relevance judgments [65]. Since teams are blind to results, those participating in shared tasks are generally more focused on answering a specific hypothesis, undertaking statistical analyses on the data they have, and not concerned with 'leaderboard chasing' [9]. This environment encourages a focus on collecting insights about why systems are effective (or not) on the test collection. We use the traits of shared tasks to cover the full cycle of experimental IR research for students (i.e., creation of test collections, system development, and statistical analysis).

Building Corpora as Coursework. In line with Humboldtian principles, the construction, annotation, and subsequent public release of natural language processing corpora as a classroom exercise has been conducted since at least the mid-2000's [39, 71]. In IR, test collections are widely used for rapid development of research prototypes and offline evaluations. Test collections form the backbone for rapid experimentation and scalable evaluation in IR [14, 65]. However, creating them is difficult and requires costly manual annotation, e.g., between 30 to 60 seconds per query-document pair [65]. Ideally, a test collection's topics and relevance judgments are created by domain experts (i.e., gold annotators [53]). However, this is time-consuming and costly. To reduce costs, several alternative approaches have been proposed, mainly revolving around so-called bronze annotators [53]. Sakai et al. [53] provide a comprehensive overview of different annotator quality levels (i.e., differences between gold and bronze annotators). For our purposes, bronze annotators may include crowdworkers [6, 30, 34, 57], large language models [20, 41], and even students in courses. For example, Althammer et al. [4] developed relevance judgments for TripClick [50]. Others have also expanded existing test collections or built new ones as part of IR coursework [55, 62].

We expand the previous ideas so students experience all parts of a shared task, ultimately aiming to combine research and teaching. Our method can be used to develop new test collections and to deepen the pool of relevance judgments for existing ones. We analyze the results of using our method in two semesters of university courses run concurrently at two universities and discuss its future prospects and scalability to include more universities.

3 TEACHING RESOURCES FOR TIREX

This section describes the resources we present to support shared task style teaching in the hands-on parts of IR courses. We tightly integrate our resources with TIREx [24] and `ir_datasets` [42] to support all four stages (see Figure 2) of our course concept: (1) topic creation, (2) relevance judgments, (3) system development, and (4) statistical analysis. Table 1 lists all resources and their corresponding stages. Each resource is either ongoing or archival: Ongoing resources are intended to be reused in subsequent iterations of courses and might be adapted when the course changes. These include a dashboard (Section 3.1), a `ir_datasets` browser (Section 3.2), and a range of tutorials (Section 3.4 and 3.7). All ongoing resources aim to provide students with a complete, easily accessible picture of experimental IR research and to support them in brainstorming ideas for subsequent practical parts. As all ongoing resources are Git repositories, from the teaching perspective, different branches or forks can reflect teachers' different flavors, opinions, or priorities. Archival resources make the results of each course easily accessible for long-term use so that the knowledge gained in the courses is not "lost" but preserved for further research.

Our teaching resources are, technically, grounded on static file hosting (for dashboards and long-term archival) and containerized IDEs (to lower the entry barrier for student submissions).

[4] https://github.com/activepapers/activepapers-python
[5] https://github.com/terrier-org/cikm2021tutorial

Table 1: Overview of our resources, their intended scope and stage during the course, hosting details, and links to each resource.

	Resource	Scope	Stage	Hosting	Link
Ongoing	Dashboard	brainstorming	1 to 4	Static★	https://tira-io.github.io/teaching-ir-with-shared-tasks
	`ir_datasets` browser	brainstorming	1 to 2	Static★	https://tira-io.github.io/ir-dataset-browser
	Tutorials	brainstorming	1 to 4	Dev containers★	https://github.com/tira-io/teaching-ir-with-shared-tasks#tutorials
Archival	Summer semester browser	qualitative analysis	3 to 4	Static★	https://tira-io.github.io/ir-lab-sose-23
	Summer semester dataset	quantitative analysis	3 to 4	Zenodo	https://doi.org/10.5281/zenodo.10628640
	Winter semester browser	qualitative analysis	3 to 4	Static★	https://tira-io.github.io/ir-lab-ws-23
	Winter semester dataset	quantitative analysis	3 to 4	Zenodo	https://doi.org/10.5281/zenodo.10628882

★We use the free tiers of GitHub (alternatives: GitLab, Cloudflare, Gitpod, etc.).

Hosting and Storage. All teaching material is ideally distributed with excellent availability, enabling students to work independently on their preferred schedules. Therefore, we host all our resources on hyperscaling infrastructure, specifically, GitHub Pages[6] for web applications and Zenodo[7] for datasets, and maintain standardized data access via `ir_datasets` (Section 3.5). Our web applications are implemented as client-side apps using Vuetify.[8] To allow for random access to documents, topics, and relevance judgments, we adopt an approach inspired by the indxr [7] Python library.[9] That library indexes document collections by byte offsets so that a document can efficiently be accessed by seeking the file to its corresponding start offset stored in an index. We transfer this principle to static files hosted via HTTP (indxr only supports local files) using HTTP range requests[10] to cover all types of data involved in experimental IR research, i.e., topics, relevance judgments, runs, and document collections. As a result, our web applications provide dynamic random access to those records of data while only requiring static file hosting (e.g., an Nginx or Apache web server[11]). To enable collaborative maintenance of those web applications while providing high reliability and uptime, we host them on the free tier of GitHub Pages.[12] The client-side web applications are automatically built and re-deployed upon each new commit with GitHub Actions.[13] Similarly, storing both the data required for the tutorials and the artifacts created in the course on Zenodo allows for safe, trusted, and reliable long-term data access.[14] Using `ir_datasets` as a unified data access layer hides technical details like HTTP requests and thus facilitates a seamless experience for students.

Containerized Development. We provide tutorials in Python notebooks that cover all four stages of the course concept. All tutorials follow the same pattern, using a micro collection to outline a small problem that is subsequently solved. The tutorials cover a range of concepts quickly so students can apply the concepts they find most interesting in their later coursework. In a teaching environment, however, it is often particularly challenging (1) to set up and maintain the hardware and development environment (especially for research-oriented software) and (2) to bridge the gap from explorative experimentation to reproducible shared task submissions. First, to lower the barrier of entry for students, all tutorials can be run in Dev containers,[15] an industry standard for running development environments as containers with all dependencies pre-installed. Because all data used in our tutorials is loaded from Zenodo using `ir_datasets`, no additional setup is required for students, and even developing using just a web browser is possible, e.g., with GitHub Codespaces.[16] Second, we integrate TIREx, which allows the execution of the containerized retrieval approaches implemented against `ir_datasets` in a sandbox (i.e., no internet access for improved reproducibility). To simplify the submission process for students (so that they do not have to learn Docker), we implemented a new code submission feature in TIREx that is compatible with all tutorials and Dev containers. Hence, all tutorials already work as code submissions. This integration of our resources with TIREx and `ir_datasets` enables the same submission to later be executed on different datasets, e.g., on a test collection constructed by a different university's course (like we describe in Section 4) or to participate in a forthcoming shared task. Furthermore, we use TIREx for blinded evaluation, i.e., teachers make the results available only after the statistical analysis (including hypotheses) was finalized, supporting result-blind grading [9].

In the following, we describe and motivate each resource, by following the typical path of students in an IR course: from brainstorming (Sections 3.1 and 3.2) and tutorials (Section 3.4) over accessing data and software components (Sections 3.5 and 3.6) to, finally, developing and analyzing solutions (Section 3.7).

3.1 A Curated Dashboard to Find Datasets, Approaches, and Evaluation Tools

Given that the solution space of a shared task is huge by design (to encourage diverse submissions and to obtain a reusable judgment pool [14]), we aim to guide students in their brainstorming. Therefore, we provide a dashboard that covers the main aspects of shared tasks: (1) test collections, (2) inspiration for system development, and (3) evaluation methodology. Each component in the dashboard links to corresponding resources, tutorials, or code snippets. We manually tag the components to enable search and filtering. For

[6]https://pages.github.com
[7]https://zenodo.org
[8]https://vuetifyjs.com
[9]https://github.com/amenra/indxr
[10]https://rfc-editor.org/rfc/rfc9110#field.range
[11]https://nginx.com or https://httpd.apache.org
[12]Alternative hosters with a free tier: Cloudflare Pages (https://pages.cloudflare.com), GitLab Pages (https://about.gitlab.com/stages-devops-lifecycle/pages)
[13]https://github.com/features/actions; alternative continuous integration service with a free tier: GitLab CI (https://about.gitlab.com/solutions/continuous-integration)
[14]Zenodo claims to operate for the next 20+ years: https://help.zenodo.org/faq
[15]https://containers.dev
[16]https://github.com/features/codespaces

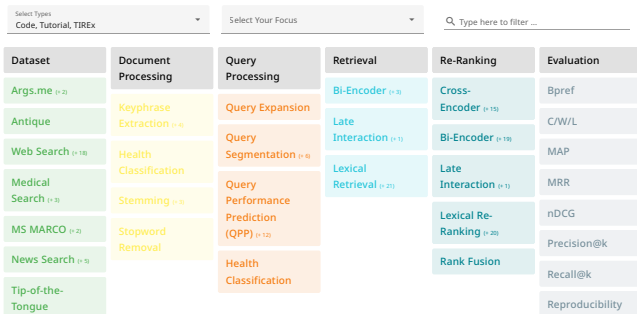


Figure 3: The dashboard with typical components of experimental IR research (test collections, systems, evaluation). Components can be expanded and have links to resources, code snippets, tutorials, and demos where applicable.

instance, components of retrieval pipelines are tagged as precision-oriented or recall-oriented if applicable so that students who aim to improve the precision using some query processing technique can find corresponding approaches (e.g., query segmentation [28]).

Figure 3 shows a preview of the dashboard available online.[17] The dashboard components and their links are rendered from a manually curated YAML file in the repository to simplify collaborative modification (each commit updates the dashboard using GitHub Actions). We intend to maintain this dashboard continuously, keeping it as simple as possible and focussing on breadth, e.g., linking contrary opinions on MRR [26, 47] instead of linking multiple homogenous resources. We link to existing web demos where applicable, e.g., the ir-measures [40] demo that allows understanding evaluation measures using small curated examples,[18] or our `ir_datasets` browser, presented in the next section.

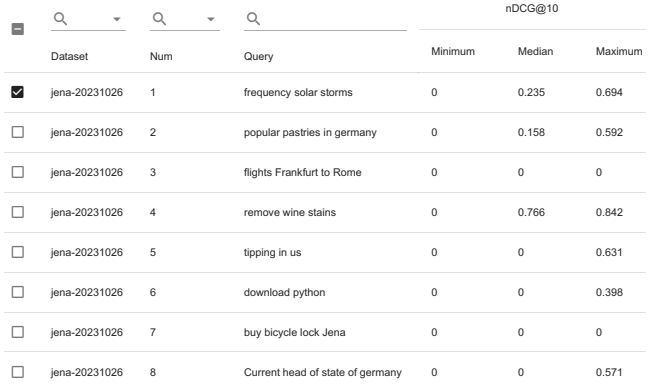
3.2 Exploring Data with an `ir_datasets` Browser

Once having grasped a sense of the task at hand (e.g., finding similar datasets, prior approaches, and evaluation measures) with the help of our curated dashboard, the next step for students is to explore the task's document collection and topics or runs from similar tasks.

The `ir_datasets` [42] framework is well suited for practical IR exercises because students can focus on IR concepts instead of data wrangling. Furthermore, `ir_datasets` highlights that the Cranfield paradigm [15, 16] transfers across diverse retrieval scenarios. Still, `ir_datasets` is used via an API, so it requires non-negligible effort to show interactive examples in a lecture, and it does not integrate runs. We close both gaps by complementing `ir_datasets` with an deep-linkable interactive browser and public runs in TIREx.

Since students create their test collections in `ir_datasets` during the first two stages of the course, their results can then also be browsed with our `ir_datasets` browser (we ensure that the document corpora allow this). Consequently, the course results are accessible, contributing to the portfolio of students' projects and advertising the course to potentially interested student peers who can quickly grasp what was done in previous years. We believe that this accessible browsing of past results increases students' motivation.

[17] https://tira-io.github.io/teaching-ir-with-shared-tasks
[18] https://demo.ir-measur.es/explore



	Dataset	Num	Query	nDCG@10 Minimum	nDCG@10 Median	nDCG@10 Maximum
☑	jena-20231026	1	frequency solar storms	0	0.235	0.694
☐	jena-20231026	2	popular pastries in germany	0	0.158	0.592
☐	jena-20231026	3	flights Frankfurt to Rome	0	0	0
☐	jena-20231026	4	remove wine stains	0	0.766	0.842
☐	jena-20231026	5	tipping in us	0	0	0.631
☐	jena-20231026	6	download python	0	0	0.398
☐	jena-20231026	7	buy bicycle lock Jena	0	0	0
☐	jena-20231026	8	Current head of state of germany	0	0	0.571

Figure 4: Overview of topics in the `ir_datasets` browser displaying topics of selected datasets with additional metadata.

Covered Test Collections. We include all public test collections integrated in TIREx, i.e., covering args.me[11, 12], ANTIQUE [29], CORD-19 [66, 67], Cranfield [15, 16], MEDLINE [31, 32, 51, 52], MS MARCO [17, 18], NFCorpus [13], and Vaswani. We excluded some frequently used test collections, like TREC Robust [63, 64], due to license requirements, sometimes involving payments. Still, being able to browse even only public test collections from TIREx with a convenient web application substantially increases their accessibility. Integrating a diverse set of test collections and systems submitted to TIREx in a unified web application further supports the students' brainstorming, as they can now browse through previous topics, find shortcomings of existing models, or get inspiration on potential research directions to focus on in their coursework.

Browsing Topics. Figure 4 shows how topics can be browsed within the web interface featuring a table with customizable column selection, sorting, and filter criteria. For each topic and measure, the minimum, median, and maximum scores and the score variance of all runs submitted to TIREx are shown. Comparing scores measured on different topics helps to assess topic difficulty. After selecting a topic, our dataset browser further shows the topic description, an overview of the relevance judgments, and all included runs.

Browsing Relevance Judgments. We make the relevance judgments (i.e., topic, document, and relevance label) browsable. Additionally, we show statistics derived from the TIREx runs, such as the median rank of each document and how many systems have retrieved the document within its top-10 (or top-100) results. Those retrievability statistics help to identify edge cases, e.g., non-relevant documents retrieved on top positions by many systems or, vice versa, relevant documents retrieved by only a few systems.

Browsing and Rendering Submitted Runs. We render the submitted runs for each dataset with DiffIR [35]. After selecting a dataset and topic, our `ir_datasets` browser first displays an overview of all runs that were submitted for that topic, as shown in Figure 5, where we show the effectiveness of each system with respect to selected effectiveness measures (e.g., nDCG@10), the proportion of judged documents in the top-10, and a visualization of the individual relevance labels of the top-10 documents. When one or more

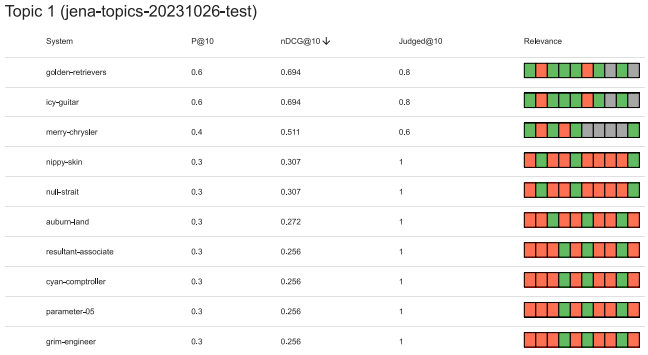


Figure 5: Overview of systems evaluated on a topic in the `ir_datasets` browser with evaluation measures and the relevance labels of the top-10 retrieved results in colored boxes.

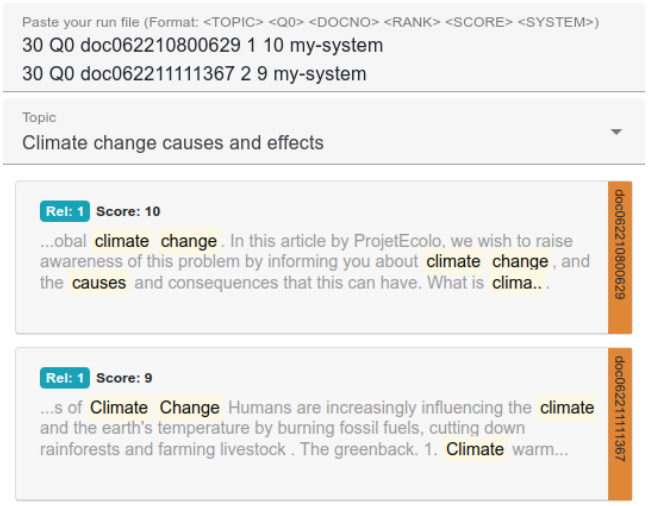


Figure 6: A search engine result page for a selected topic rendered with DiffIR from a run file pasted into a text field.

systems are selected to be compared, we show their detailed rankings, rendered with DiffIR. Furthermore, custom run files can be uploaded or just pasted into a text field of the `ir_datasets` browser and subsequently can be compared to existing runs (see Figure 6). Originally, DiffIR was intended for local usage and embeds all topics and documents into a single HTML file. As this behavior causes efficiency issues on corpora with large documents, we modify DiffIR to load only documents and snippets when needed.

3.3 Assessing Relevance with Doccano

After their initial data exploration and after devising topics (Stage 1), students assess the relevance of a pool of documents retrieved for their topics in Stage 2. In contrast to shared tasks, students have not yet developed retrieval approaches at this stage. Instead, we use retrieval approaches submitted to TIREx (covering different paradigms) to build a judgment pool. To mitigate the effect of potentially ambiguous topic descriptions on the quality of relevance judgments, each student team assesses the relevance of the documents retrieved for their own proposed topics. We use Doccano[19] as online annotation tool. We publish the scripts to create user accounts and importing and exporting the data to Doccano.

Listing 1: Accessing a test collection created in the course with `ir_datasets` via TIRA to load datasets hosted on Zenodo.

```
# Patch ir_dataset to load datasets from TIRA.
from tira.third_party_integrations import ir_datasets
dataset = ir_datasets.load(
  'ir-lab-<university>-wise-2023/test-topics'
)
```

3.4 Learning Concepts with Hands-On Tutorials

We provide tutorials that cover `ir_datasets` for data loading, standard document and query processing (e.g., stopword removal, stemming, lemmatization), query expansion, hyperparameter tuning, learning to rank, query segmentation, query performance prediction, and hypothesis testing. All tutorials are implemented in a Dev container with all dependencies pre-installed to ensure that the tutorials can be used reliably by students with different operating systems (e.g., Windows, Linux, or macOS). Each tutorial covers a single IR concept where the problem is demonstrated and subsequently solved on a small cherry-picked example test collection. We designed each tutorial to be completed in roughly 15 minutes.

Because the tutorials are implemented inside a Dev container, students can run the tutorials online in GitHub Codespaces, and instantly start with the tutorial without installing dependencies first. We prefer Github Codespaces over Google Colab,[20] for three reasons: (1) GitHub Codespaces explicitly documents the computational resources included in the free tier (e.g., 120 CPU hours per month for anyone, 180 hours for education accounts; in our experience so far, no students have ever exceeded this limit), whereas Google Colab is unclear about resource constraints. (2) As GitHub Codespaces is based on Dev containers, students can mirror the same development environment locally using open-source tools like Visual Studio Code[21] and Docker/Podman.[22] (3) Dev containers are better integrated with Git to support teamwork code versioning.

3.5 Reliably Accessing Data with Zenodo

Access to datasets used in the tutorials and for system development or statistical analysis should be standardized, stable, and reliable so that our teaching concept can scale to multiple larger courses run in parallel. While `ir_datasets` offers a standardized and scalable interface, it officially only includes mature and finalized test collections. Conversely, the test collections that students create as part of their coursework are of unclear quality, less mature, and prototypical even after finishing the course. Therefore, directly integrating these test collections created during the course into the official `ir_datasets` repository is unsuitable. More appropriately, we implement a dynamic patch for `ir_datasets` in the TIRA [24]

[19] https://github.com/doccano/doccano
[20] https://colab.research.google.com
[21] https://code.visualstudio.com
[22] https://docker.com or https://podman.io

Listing 2: Example on how to reuse an query processing approach (here: a query segmentation) in a PyTerrier pipeline.

```
dataset = pt.get_dataset("irds:<tirex-dataset-id>")
query_segmentation = tira.pt.transform_queries(
  '<team>/query-segmentation',
  dataset
)
# Apply the query segmentation to the topics.
query_segmentation.transform(dataset.get_topics())
```

client library that loads datasets from Zenodo, enabling stable and scalable data access for emerging datasets while maintaining standardization. Listing 1 shows how to load a test collection created in a course from Zenodo via TIRA's `ir_datasets` integration. The patch falls back to the standard behavior of `ir_datasets` if the dataset identifier already exists in the `ir_datasets` repository but loads the dataset from TIRA/Zenodo otherwise. When executed in a TIRA sandbox without internet access, this patch loads the dataset from a read-only mount specified via environment variables so that the same code can run on different datasets.

Zenodo supports versioned datasets, allowing test collections to evolve throughout the course according to the four stages. For Stage 1, only the document collection must be available for exploration during topic creation (e.g., to ensure that relevant documents exist for a potential topic). If a training and validation dataset is available at this stage or can be constructed (e.g., with relevance judgments created with a large language model [20]) teachers can integrate these datasets before the start of Stage 3. After Stage 3 has ended and the teachers unblinded all student runs of the course, the dataset on Zenodo can again be updated with the complete test collection, including topics, relevance judgments, and submitted runs, to support the statistical analysis in Stage 4.

3.6 Reusing Previously Submitted Components

Incorporating a research-oriented focus into the course concept introduces the difficulty that the software of existing approaches is often difficult to install and run within the time budget of the course. In TIREx, however, submissions are immutable, which allows for caching their outputs [24]. As our course concept is integrated with TIREx, students can thus directly use cached outputs of previously submitted software components. To encourage such software reuse, we extend TIREx with a Python API that provides access to cached outputs from within the TIREx sandbox and outside of it as declarative PyTerrier [43] pipelines. Besides simplifying the prototyping process, this also supports green IR [54], as each component must be executed only once on a dataset.

With a special call for retrieval components, the recent Workshop on Open Web Search [22] at ECIR 2024[23] aimed to gather a critical mass of reusable components. The workshop specified standardized input and output formats for different types of retrieval components: (1) query processing, (2) document processing, and (3) query–document processing. Our tutorials cover all three such types. By extending the TIREx Python API, the tutorials include already submitted components like query segmentation (i.e., query processing), keyphrase extraction (i.e., document processing), and post-retrieval query-performance predictors [21] (i.e., query–document processing). Listing 2 shows how to reuse a query segmentation approach [28] in a declarative PyTerrier pipeline.

[23] https://opensearchfoundation.org/wows2024

3.7 Submitting Software as Source Code

TIREx initially called for submissions of retrieval software as Docker images [24]. However, in a volatile and time-constrained teaching environment, forcing students to learn and use Docker is a disadvantage due to its steep learning curve. Instead, their time is better invested in learning core IR concepts. Hence, we developed a new source code submission feature for TIREx that reduces the interaction with Docker as much as possible. When a student registers for the course's task on TIREx, it creates a pre-configured private GitHub repository from a template, makes the student the repository owner, and grants teachers read/write access. The automatically created repository comes with a prepared Python notebook for a baseline submission very similar to the tutorials. The baseline notebook can be adapted and tested by students in GitHub Codespaces or locally using Dev containers. After a student pushes a change to the repository, the submission is automatically packaged in a Docker image, tested, and uploaded to TIRA using GitHub Actions. This process allows students to submit their retrieval approaches as Docker images without learning Docker themselves. As the underlying Docker image of this GitHub Actions workflow is the same as the image used in the tutorials, students can use code from the tutorials without modification. More experienced students are still free to adapt the Docker images. When assisting students, teachers are equally supported by our workflow, as they can easily open student repositories online and debug their approaches without installation, facilitating fast switching of mentoring different teams which contributes to the scalability of our course concept.

Since Docker images submitted to TIREx via this source code submission feature are built within GitHub Actions, we can collect all metadata for provenance and improved reproducibility. This additional metadata uploaded to TIREx includes the Git repository, its version (i.e., the commit hash and branch), and the Python notebook to be executed in the Docker image. As a result, for any source code submission, TIREx contains a deep link to the exact same version of the Python Notebook on GitHub that was used to produce the submission. For public repositories, TIREx displays this link in the leaderboard and in the dataset browser, allowing others to reproduce the students' approaches easily. Especially this enables student teams to learn and inspire each other because students can directly jump to the code that produced some results, further highlighting also the importance of easy browsing of results with our `ir_datasets` Browser.

4 CASE STUDY

We perform a case study over two semesters of IR courses across two universities with our resources to run our course concept. All data of both semesters is publicly available (links in Table 1).

In the first semester we had 66 students and 16 groups (64 in university A and 2 in university B); in the second semester we had 68 students and 16 groups (55 in university A and 13 in university B). Figure 7 illustrates the number of students in each group.

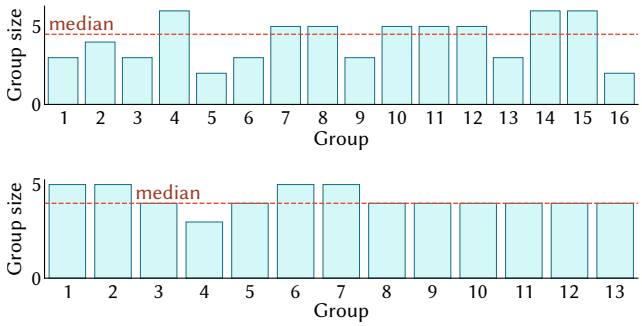


Figure 7: Distributions of group sizes for the summer semester 2023 (top) and winter semester 2023/24 (bottom). The red line indicates the median group size.

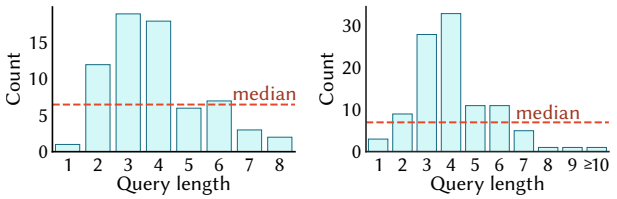


Figure 8: Distributions of query lengths for the summer semester 2023 (left) and winter semester 2023/24 (right). The red line indicates the median query length.

Since most groups were randomly assigned and a small number of students dropped out, there were two groups with only two students across both semesters. We allocated the students one week to formulate topics (Stage 1; one topic per student), one week for relevance judgments (Stage 2), four weeks to develop a retrieval system (Stage 3), and five weeks for the statistical analyses including writing a short report (Stage 4). In the first semester, we used the IR Anthology [48] as the document collection[24] (no forthcoming shared task); in the second semester, the document collection was the LongEval [1, 2, 27] corpus (i.e., aligning the course with the then upcoming LongEval 2024 [3] task[25]).

4.1 Stage 1: Topic Formulation

In the first stage, students created topics in groups. In the lecture and the subsequent tutorial, we introduced the concept of topics and their purpose. To support topic creation, we dedicated a practical exercise to provide students with examples of good topics guided by the tutorials provided by our new resources, including example topics for the document collection created by the teachers. Once the student groups started creating topics independently (each student created one topic), a basic search engine with the document collection was provided to help familiarise themselves with it.

Figure 8 shows that, in both semesters, most queries are short (median query length of 6.5 terms in the first semester and 7.0 terms in the second). Table 2 shows a sample of queries created by students.

[24]https://ir.webis.de/anthology
[25]https://clef-longeval.github.io

Table 2: Sample of student topics from the two semesters. We show the proportion of relevant to non-relevant documents (Prop.) and the number of judged documents (Docs.).

Query	Prop.	Docs.
Summer semester 2023		
fake news detection	94%	31
inclusion of text-mining	93%	41
natural language processing	80%	35
actual experiments that strengthen theoretical knowl…	69%	45
retrieval system improving effectiveness	65%	46
information retrieval on different language sources	64%	50
sentiment analysis	62%	37
the university of amsterdam	57%	23
at least three authors	53%	45
document scoping formula	7%	43
Winter semester 2023/2024		
what is my ip address	67%	36
reduce risk cardiovascular disease	60%	45
climate change causes and effects	60%	60
recognize phishing attack	52%	44
artificial intelligence applications	50%	50
ai ethical downsides	40%	43
vegan alternatives to common dairy products	30%	40
the frequency of solar storms with impact on electric…	27%	33
artificial intelligence in healthcare	23%	52
3d print warping	2%	49

In the first semester, query terms are mostly about concepts related to IR, with the terms 'information' and 'retrieval' appearing most frequently and second most frequently. Topics of semester two had substantially fewer overlapping terms, caused by its much more open scenario of LongEval on general Web search compared to the very focused scenario of the second semester on the IR Anthology.

4.2 Stage 2: Relevance Judgments

In the second stage, students created relevance judgments in groups on the same topics they created in the previous stage. In the lecture and the tutorial, students are introduced to the concept of relevance judgment and its purpose. We also did another practical exercise to show the students how to assess documents. As a class activity, students and the teaching assistant(s) worked together to create relevance judgments for an exemplary topic from the previous stage. After this practical session, we provided the students with the topic pools, which they assessed via Doccano. The judgment pools were prepared by creating a top 10 pool for retrieval models used in PyTerrier [43]. In our course, we used ten retrieval models, five lexical models (BM25, TF-IDF, DPH, INL2, PL2, DirichletLM, and InExpB2), two lexical models with query rewriting (BM25 with RM3 and BM25 with SDM), and three neural models (ColBERT [36], MonoT5 [49], and ANCE [69]) re-ranking the top 1000 results from BM25. All models used default settings.

Table 2 shows the statistics for relevance judgments for a sample of topics across both semesters. Overall, students created 66 topics with a total of 2635 relevance judgments in the first semester, and 101 topics with a total of 4997 relevance judgments in the second semester. Qualitatively, based on the proportion of relevant and

non-relevant documents per topic, most topics were assessed well for both semesters. Still, for some topics across both semesters, the topics were too broad or too specific, so almost all documents were either relevant or non-relevant. We also found several topics in both semesters that did not have any relevant documents, although such results were to be expected for prototyping new test collections, as filtering out unsuitable topics help to iteratively make the test collection more mature if the initial results obtained throughout the course reveal interesting research potential.

To validate the students' judgements, five IR PhD students additionally created expert judgements for a subset of topics (26 topics for the first semester, 10 topics for the second). We made the judgments in the same depth as the students rather than judging all topics shallowly. Judgments in the same depth allow us to better to distinguish the systems' effectiveness from the students and compare agreement between annotators. In the first semester, we obtained an annotator agreement of 0.76 and a Cohen's κ of 0.52 (using 959 overlapping judgments between students and experts); in the second, we obtained an annotator agreement of 0.73 and a Cohen's κ of 0.42 (using 346 overlapping judgments).

4.3 Stages 3 and 4: Development and Analyses

In the third and fourth stages, groups of students developed a retrieval system for the document collection and performed their self-guided analyses of their systems. In the lecture, students learnt about the general architecture of a search engine, such as retrieval models and query expansion, and in practical sessions, we used our tutorials to show how to develop a baseline system in PyTerrier, create a run file, and submit this to TIREx. As an extra layer of competition, we unblinded a leaderboard on a small open training dataset that automatically updated after each submission, whereas the final evaluation was blinded until all submissions were made (we clarified that the results do not affect their grade). For the final leaderboards after all submissions were finalized, each system was evaluated using all groups' combined topic sets and relevance judgments, but we only revealed the leaderboard in the fourth stage without accepting new submissions to allow students to conduct statistical analyses on previously blinded runs.

In the first semester, of the 16 groups, 14 made a valid submission. In semester two, of the 13 groups, 11 made a valid submission. The students' approaches to retrieval included steps to pre-process the documents to improve statistical ranking methods such as tf-idf, BM25 and DPH, relevance feedback with RM3 and neural ranking models such as monoT5. Using the two sets of relevance judgments described above, we analysed how assessors with different levels of expertise affect system ranking. Using both sets of relevance judgments, we ranked students' systems according to mean average precision (MAP) to study the rank correlation using Kendall's τ.

For the first semester, the top system measured using student judgments achieved a MAP of 0.52 and 0.42 for expert judgments. System rankings are highly correlated for student and expert judgments, reaching Kendall's τ of 0.8, indicating that students can produce usable test collections. For semester two, the top system using student judgments achieved a MAP of 0.57 and 0.53 for expert judgments, although the correlation of system rankings achieved a lower Kendall's τ of 0.2. One reason the inter-annotator agreement was high but the ranking correlation was low may be due to students purposefully designing difficult topics where smaller inconsistencies in judgments have a larger effect on system rankings.

Using the combined set of all topics for both semesters, we observe that the best student submissions outperform our BM25 baseline, which is a strong contribution given that students operated in a low data regime with blinded evaluation. In the first semester for the IR Anthology, the best student system achieved an MAP of 0.43 and an nDCG@10 of 0.628 compared to 0.34 and 0.49 of our baseline. In the second semester for LongEval, the best student system achieved an MAP of 0.33 and an nDCG@10 of 0.40 compared to 0.31 and 0.35 of our baseline. Those results highlight that the first semester's focused scenario helped students achieve substantial improvements, whereas even smaller improvements are meaningful for the diverse retrieval scenario of the second semester.

5 ETHICAL CONSIDERATIONS

In the SharKI project[26] that partially supported our research, we study whether shared tasks support computer science education and received ethical clearance. All the formulated topics and constructed datasets of the students are published on Zenodo only with their consent. Students usually are the co-authors of the respective Zenodo record (except when they opt out) and Zenodo creates a citeable DOI and tracks downloads, views, and citations so that the students get recognition for their work.

6 CONCLUSION

We have presented resources to combine teaching and research in IR, supporting a course workflow in which students collaboratively "organize" a shared task by creating test collections, developing systems, and conducting statistical analyses. Our resources and course concept are strongly aligned with student learning objectives in IR courses and scale to large student cohorts who can create test collections from scratch, extend existing test collections, or increase participation in forthcoming shared tasks. We analyzed the student-generated topics and relevance judgments and found them to be consistent with those of experts. All our resources (incl. the topics, relevance judgments, and retrieval approaches created by students in several courses) are publicly available.[27] Interesting directions for the future are instantiating the course concept at more universities and also targeting interactive retrieval scenarios.

ACKNOWLEDGMENTS

This work has been partially supported by the SharKI project (funded by the BMBF; FKZ: 16DHB4021) and by the OpenWebSearch.eu project (funded by the EU; GA 101070014).

REFERENCES

[1] Rabab Alkhalifa, Iman Munire Bilal, Hsuvas Borkakoty, José Camacho-Collados, Romain Deveaud, Alaa El-Ebshihy, Luis Espinosa Anke, Gabriela González Sáez, Petra Galuscáková, Lorraine Goeuriot, Elena Kochkina, Maria Liakata, Daniel Loureiro, Philippe Mulhem, Florina Piroi, Martin Popel, Christophe Servan, Harish Tayyar Madabushi, and Arkaitz Zubiaga. 2023. Overview of the CLEF-2023 LongEval Lab on Longitudinal Evaluation of Model Performance. In

[26] https://sharki-project.github.io/

[27] http://github.com/tira-io/teaching-ir-with-shared-tasks

Proceedings of CLEF 2023 (LNCS, Vol. 14163). Springer, Berlin, 440–458. https://doi.org/10.1007/978-3-031-42448-9_28

[2] Rabab Alkhalifa, Iman Munire Bilal, Hsuvas Borkakoty, José Camacho-Collados, Romain Deveaud, Alaa El-Ebshihy, Luis Espinosa Anke, Gabriela Nicole González Sáez, Petra Galuscáková, Lorraine Goeuriot, Elena Kochkina, Maria Liakata, Daniel Loureiro, Philippe Mulhem, Florina Piroi, Martin Popel, Christophe Servan, Harish Tayyar Madabushi, and Arkaitz Zubiaga. 2023. Extended Overview of the CLEF-2023 LongEval Lab on Longitudinal Evaluation of Model Performance. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023 (CEUR Workshop Proceedings, Vol. 3497)*, Mohammad Aliannejadi, Guglielmo Faggioli, Nicola Ferro, and Michalis Vlachos (Eds.). CEUR-WS.org, 2181–2203. https://ceur-ws.org/Vol-3497/paper-184.pdf

[3] Rabab Alkhalifa, Hsuvas Borkakoty, Romain Deveaud, Alaa El-Ebshihy, Luis Espinosa Anke, Tobias Fink, Gabriela González Sáez, Petra Galuscáková, Lorraine Goeuriot, David Iommi, Maria Liakata, Harish Tayyar Madabushi, Pablo Medina-Alias, Philippe Mulhem, Florina Piroi, Martin Popel, Christophe Servan, and Arkaitz Zubiaga. 2024. LongEval: Longitudinal Evaluation of Model Performance at CLEF 2024. In *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24-28, 2024, Proceedings, Part VI (Lecture Notes in Computer Science, Vol. 14613)*, Nazli Goharian, Nicola Tonellotto, Yulan He, Aldo Lipani, Graham McDonald, Craig Macdonald, and Iadh Ounis (Eds.). Springer, 60–66. https://doi.org/10.1007/978-3-031-56072-9_8

[4] Sophia Althammer, Sebastian Hofstätter, Suzan Verberne, and Allan Hanbury. 2022. TripJudge: A Relevance Judgement Test Collection for TripClick Health Retrieval. In *Proceedings of CIKM 2022*. ACM, New York, 3801–3805. https://doi.org/10.1145/3511808.3557714

[5] Leif Azzopardi, Matt Crane, Hui Fang, Grant Ingersoll, Jimmy Lin, Yashar Moshfeghi, Harrisen Scells, Peilin Yang, and Guido Zuccon. 2017. The Lucene for Information Access and Retrieval Research (LIARR) Workshop at SIGIR 2017. In *Proceedings of SIGIR 2017*. ACM, New York, 1429–1430. https://doi.org/10.1145/3077136.3084374

[6] Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2016. UQV100: A Test Collection with Query Variability. In *Proceedings of SIGIR 2016*. ACM, New York, 725–728. https://doi.org/10.1145/2911451.2914671

[7] Elias Bassani and Nicola Tonellotto. 2024. indxr: A Python Library for Indexing File Lines. In *Proceedings of ECIR 2024 (LNCS)*. Springer, Berlin.

[8] Christine Bauer, Ben Carterette, Nicola Ferro, Norbert Fuhr, Joeran Beel, Timo Breuer, Charles L. A. Clarke, Anita Crescenzi, Gianluca Demartini, Giorgio Maria Di Nunzio, Laura Dietz, Guglielmo Faggioli, Bruce Ferwerda, Maik Fröbe, Matthias Hagen, Allan Hanbury, Claudia Hauff, Dietmar Jannach, Noriko Kando, Evangelos Kanoulas, Bart P. Knijnenburg, Udo Kruschwitz, Meijie Li, Maria Maistro, Lien Michiels, Andrea Papenmeier, Martin Potthast, Paolo Rosso, Alan Said, Philipp Schaer, Christin Seifert, Damiano Spina, Benno Stein, Nava Tintarev, Julián Urbano, Henning Wachsmuth, Martijn C. Willemsen, and Justin Zobel. 2023. Report on the Dagstuhl Seminar on Frontiers of Information Access Experimentation for Research and Education. *SIGIR Forum* 57, 1 (2023), 7:1–7:28. https://doi.org/10.1145/3636341.3636351

[9] Christine Bauer, Maik Fröbe, Dietmar Jannach, Udo Kruschwitz, Paolo Rosso, Damiano Spina, and Nava Tintarev. 2023. Overcoming Methodological Challenges in Information Retrieval and Recommender Systems through Awareness and Education. arXiv 2305.01509. https://doi.org/10.48550/arXiv.2305.01509

[10] Daniel Blank, Norbert Fuhr, Andreas Henrich, Thomas Mandl, Thomas Rölleke, Hinrich Schütze, and Benno Stein. 2011. Teaching IR: Curricular Considerations. In *Teaching and Learning in Information Retrieval*. INRE, Vol. 31. Springer, Berlin, 31–46. https://doi.org/10.1007/978-3-642-22511-6_3

[11] Alexander Bondarenko, Maik Fröbe, Meriem Beloucif, Lukas Gienapp, Yamen Ajjour, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and Matthias Hagen. 2020. Overview of Touché 2020: Argument Retrieval. In *Proceedings of CLEF 2020 (LNCS, Vol. 12260)*. Springer, Berlin, 384–395. https://doi.org/10.1007/978-3-030-58219-7_26

[12] Alexander Bondarenko, Lukas Gienapp, Maik Fröbe, Meriem Beloucif, Yamen Ajjour, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and Matthias Hagen. 2021. Overview of Touché 2021: Argument Retrieval. In *Proceedings of CLEF 2021 (LNCS, Vol. 12880)*. Springer, Berlin, 450–467. https://doi.org/10.1007/978-3-030-85251-1_28

[13] Vera Boteva, Demian Gholipour Ghalandari, Artem Sokolov, and Stefan Riezler. 2016. A Full-Text Learning to Rank Dataset for Medical Information Retrieval. In *Proceedings of ECIR 2016 (LNCS, Vol. 9626)*. Springer, Berlin, 716–722. https://doi.org/10.1007/978-3-319-30671-1_58

[14] Chris Buckley, Darrin Dimmick, Ian Soboroff, and Ellen M. Voorhees. 2007. Bias and the Limits of Pooling for Large Collections. *Inf. Retr.* 10, 6 (2007), 491–508. https://doi.org/10.1007/S10791-007-9032-X

[15] Cyril W. Cleverdon. 1967. The Cranfield Tests on Index Language Devices. In *ASLIB Proceedings*, Vol. 19. Emerald, Leeds, 173–192. https://doi.org/10.1108/eb050097

[16] Cyril W. Cleverdon. 1991. The Significance of the Cranfield Tests on Index Languages. In *Proceedings of SIGIR 1991*. ACM, New York, 3–12. https://doi.org/10.1145/122860.122861

[17] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2020. Overview of the TREC 2020 Deep Learning Track. In *Proceedings of TREC 2020 (NIST Special Publication, Vol. 1266)*. NIST, Gaithersburg, 13 pages. https://trec.nist.gov/pubs/trec29/papers/OVERVIEW.DL.pdf

[18] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2019. Overview of the TREC 2019 Deep Learning Track. In *Proceedings of TREC 2019 (NIST Special Publication, Vol. 1250)*. NIST, Gaithersburg, 22 pages. https://trec.nist.gov/pubs/trec28/papers/OVERVIEW.DL.pdf

[19] Theresa Elstner, Frank Loebe, Yamen Ajjour, Christopher Akiki, Alexander Bondarenko, Maik Fröbe, Lukas Gienapp, Nikolay Kolyada, Janis Mohr, Stephan Sandfuchs, Matti Wiegmann, Jörg Frochte, Nicola Ferro, Sven Hofmann, Benno Stein, Matthias Hagen, and Martin Potthast. 2023. Shared Tasks as Tutorials: A Methodical Approach. In *Proceedings of EAAI 2023*. AAAI Press, Washington, DC, 15807–15815. https://doi.org/10.1609/AAAI.V37I13.26877

[20] Guglielmo Faggioli, Laura Dietz, Charles L. A. Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, and Henning Wachsmuth. 2023. Perspectives on Large Language Models for Relevance Judgment. In *Proceedings of ICTIR 2023*. ACM, New York, 39–50. https://doi.org/10.1145/3578337.3605136

[21] Guglielmo Faggioli, Oleg Zendel, J. Shane Culpepper, Nicola Ferro, and Falk Scholer. 2021. An Enhanced Evaluation Framework for Query Performance Prediction. In *Proceedings of ECIR 2021 (LNCS, Vol. 12656)*. Springer, Berlin, 115–129. https://doi.org/10.1007/978-3-030-72113-8_8

[22] Sheikh Farzana, Maik Fröbe, Michael Granitzer, Gijs Hendriksen, Djoerd Hiemstra, Martin Potthast, and Saber Zerhoudi. 2024. The First International Workshop on Open Web Search (WOWS). In *Proceedings of ECIR 2024 (LNCS)*. Springer, Berlin.

[23] Juan M. Fernández-Luna, Juan F. Huete, Andrew MacFarlane, and Efthimis N. Efthimiadis. 2009. Teaching and Learning in Information Retrieval. *Inf. Retr.* 12, 2 (2009), 201–226. https://doi.org/10.1007/S10791-009-9089-9

[24] Maik Fröbe, Jan Heinrich Reimer, Sean MacAvaney, Niklas Deckers, Simon Reich, Janek Bevendorff, Benno Stein, Matthias Hagen, and Martin Potthast. 2023. The Information Retrieval Experiment Platform. In *Proceedings of SIGIR 2023*. ACM, New York, 2826–2836. https://doi.org/10.1145/3539618.3591888

[25] Maik Fröbe, Matti Wiegmann, Nikolay Kolyada, Bastian Grahm, Theresa Elstner, Frank Loebe, Matthias Hagen, Benno Stein, and Martin Potthast. 2023. Continuous Integration for Reproducible Shared Tasks with TIRA.io. In *Proceedings of ECIR 2023 (LNCS, Vol. 13982)*. Springer, Berlin, 236–241. https://doi.org/10.1007/978-3-031-28241-6_20

[26] Norbert Fuhr. 2017. Some Common Mistakes In IR Evaluation, And How They Can Be Avoided. *SIGIR Forum* 51, 3 (2017), 32–41. https://doi.org/10.1145/3190580.3190586

[27] Petra Galuscáková, Romain Deveaud, Gabriela González Sáez, Philippe Mulhem, Lorraine Goeuriot, Florina Piroi, and Martin Popel. 2023. LongEval-Retrieval: French-English Dynamic Test Collection for Continuous Web Search Evaluation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete (Eds.). ACM, 3086–3094. https://doi.org/10.1145/3539618.3591921

[28] Matthias Hagen, Martin Potthast, Benno Stein, and Christof Bräutigam. 2011. Query Segmentation Revisited. In *Proceedings of WWW 2011*. ACM, New York, 97–106. https://doi.org/10.1145/1963405.1963423

[29] Helia Hashemi, Mohammad Aliannejadi, Hamed Zamani, and W. Bruce Croft. 2020. ANTIQUE: A Non-factoid Question Answering Benchmark. In *Proceedings of ECIR 2020 (LNCS, Vol. 12036)*. Springer, 166–173.

[30] Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. 2017. DBpedia-Entity v2: A Test Collection for Entity Search. In *Proceedings of SIGIR 2017*. ACM, New York, 1265–1268. https://doi.org/10.1145/3077136.3080751

[31] William R. Hersh, Ravi Teja Bhupatiraju, L. Ross, Aaron M. Cohen, Dale Kraemer, and Phoebe Johnson. 2004. TREC 2004 Genomics Track Overview. In *Proceedings of TREC 2004 (NIST Special Publication, Vol. 500-261)*. NIST, Gaithersburg, 19 pages. http://trec.nist.gov/pubs/trec13/papers/GEO.OVERVIEW.pdf

[32] William R. Hersh, Aaron M. Cohen, Jianji Yang, Ravi Teja Bhupatiraju, Phoebe M. Roberts, and Marti A. Hearst. 2005. TREC 2005 Genomics Track Overview. In *Proceedings of TREC 2005 (NIST Special Publication, Vol. 500-266)*. NIST, Gaithersburg, 26 pages. http://trec.nist.gov/pubs/trec14/papers/GEO.OVERVIEW.pdf

[33] Konrad Hinsen. 2015. ActivePapers: A Platform for Publishing and Archiving Computer-aided Research. *F1000Research* 3, 289 (2015). https://doi.org/10.12688/f1000research.5773.3

[34] Oana Inel, Giannis Haralabopoulos, Dan Li, Christophe Van Gysel, Zoltán Szlávik, Elena Simperl, Evangelos Kanoulas, and Lora Aroyo. 2018. Studying Topical Relevance with Evidence-based Crowdsourcing. In *Proceedings of CIKM 2018*. ACM, New York, 1253–1262. https://doi.org/10.1145/3269206.3271779

[35] Kevin Martin Jose, Thong Nguyen, Sean MacAvaney, Jeffrey Dalton, and Andrew Yates. 2021. DiffIR: Exploring Differences in Ranking Models' Behavior. In *Proceedings of SIGIR 2021*. ACM, 2595–2599. https://doi.org/10.1145/3404835.3462784

[36] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of SIGIR 2020*. ACM, New York, 39–48. https://doi.org/10.1145/3397271.3401075

[37] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Frassetto Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of SIGIR 2021*. ACM, New York, 2356–2362. https://doi.org/10.1145/3404835.3463238

[38] Rafael López-García and Fidel Cacheda. 2011. A Technical Approach to Information Retrieval Pedagogy. In *Teaching and Learning in Information Retrieval*. INRE, Vol. 31. Springer, Berlin, 89–105. https://doi.org/10.1007/978-3-642-22511-6_7

[39] Anke Lüdeling, Seanna Doolittle, Hagen Hirschmann, Karin Schmidt, and Maik Walter. 2008. Das Lernerkorpus Falko. *Deutsch als Fremdsprache* 45, 2 (2008), 67. https://doi.org/10.37307/j.2198-2430.2008.02.02

[40] Sean MacAvaney, Craig Macdonald, and Iadh Ounis. 2022. Streamlining Evaluation with ir-measures. In *Proceedings of ECIR 2022 (LNCS, Vol. 13186)*. Springer, 305–310. https://doi.org/10.1007/978-3-030-99739-7_38

[41] Sean MacAvaney and Luca Soldaini. 2023. One-Shot Labeling for Automatic Relevance Estimation. In *Proceedings of SIGIR 2023*. ACM, New York, 2230–2235. https://doi.org/10.1145/3539618.3592032

[42] Sean MacAvaney, Andrew Yates, Sergey Feldman, Doug Downey, Arman Cohan, and Nazli Goharian. 2021. Simplified Data Wrangling with ir_datasets. In *Proceedings of SIGIR 2021*. ACM, 2429–2436. https://doi.org/10.1145/3404835.3463254

[43] Craig Macdonald and Nicola Tonellotto. 2020. Declarative Experimentation in Information Retrieval using PyTerrier. In *Proceedings of ICTIR 2020*. ACM, New York, 161–168. https://doi.org/10.1145/3409256.3409829

[44] Craig Macdonald, Nicola Tonellotto, and Sean MacAvaney. 2021. IR From Bag-of-words to BERT and Beyond through Practical Experiments.. In *Proceedings of CIKM 2021*. 4861.

[45] Craig Macdonald, Nicola Tonellotto, Sean MacAvaney, and Iadh Ounis. 2021. PyTerrier: Declarative Experimentation in Python from BM25 to Dense Retrieval. In *Proceedings of CIKM 2021*. 4526–4533.

[46] Ilya Markov and Maarten de Rijke. 2018. What Should We Teach in Information Retrieval? *SIGIR Forum* 52, 2 (2018), 19–39. https://doi.org/10.1145/3308774.3308780

[47] Alistair Moffat. 2023. Categorical, Ratio, and Professorial Data: The Case for Reciprocal Rank. arXiv 2312.12672. https://doi.org/10.48550/arXiv.2312.12672

[48] Martin Potthast, Sebastian Günther, Janek Bevendorff, Jan Philipp Bittner, Alexander Bondarenko, Maik Fröbe, Christian Kahmann, Andreas Niekler, Michael Völske, Benno Stein, and Matthias Hagen. 2021. The Information Retrieval Anthology. In *Proceedings of SIGIR 2021*. ACM, New York, 2550–2555. https://doi.org/10.1145/3404835.3462798

[49] Ronak Pradeep, Rodrigo Frassetto Nogueira, and Jimmy Lin. 2021. The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models. arXiv 2101.05667. https://doi.org/10.48550/arXiv.2101.05667

[50] Navid Rekabsaz, Oleg Lesota, Markus Schedl, Jon Brassey, and Carsten Eickhoff. 2021. TripClick: The Log Files of a Large Health Web Search Engine. In *Proceedings of SIGIR 2021*. 2507–2513. https://doi.org/10.1145/3404835.3463242

[51] Kirk Roberts, Dina Demner-Fushman, Ellen M. Voorhees, William R. Hersh, Steven Bedrick, and Alexander J. Lazar. 2018. Overview of the TREC 2018 Precision Medicine Track. In *Proceedings of TREC 2018 (NIST Special Publication, Vol. 500-331)*. NIST.

[52] Kirk Roberts, Dina Demner-Fushman, Ellen M. Voorhees, William R. Hersh, Steven Bedrick, Alexander J. Lazar, and Shubham Pant. 2017. Overview of the TREC 2017 Precision Medicine Track. In *Proceedings of TREC 2017 (NIST Special Publication, Vol. 500-324)*. NIST.

[53] Tetsuya Sakai, Sijie Tao, Nuo Chen, Yujing Li, Maria Maistro, Zhumin Chu, and Nicola Ferro. 2024. On the Ordering of Pooled Web Pages, Gold Assessments, and Bronze Assessments. *ACM Trans. Inf. Syst.* 42, 1 (2024), 23:1–23:31. https://doi.org/10.1145/3600227

[54] Harrisen Scells, Shengyao Zhuang, and Guido Zuccon. 2022. Reduce, Reuse, Recycle: Green Information Retrieval Research. In *Proceedings of SIGIR 2022*. ACM, New York, 2825–2837. https://doi.org/10.1145/3477495.3531766

[55] Philipp Schaer. 2012. Better than Their Reputation? On the Reliability of Relevance Assessments with Students. In *Proceedings of CLEF 2012*. Springer, Berlin, 124–135. https://doi.org/10.1007/978-3-642-33247-0_14

[56] Alan F. Smeaton and Donna K. Harman. 1997. The TREC Experiments and their Impact on Europe. *J. Inf. Sci.* 23, 2 (1997), 169–174. https://doi.org/10.1177/016555159702300208

[57] Manuel Steiner, Damiano Spina, Falk Scholer, and Lawrence Cavedon. 2021. Crowdsourcing Backstories for Complex Task-Based Search. In *Proceedings of ADCS 2021*. ACM, New York, 5:1–5:6. https://doi.org/10.1145/3503516.3503526

[58] Rudolf Stichweh. 1994. The Unity of Teaching and Research. In *Romanticism in Science: Science in Europe, 1790–1840*. BSPS, Vol. 152. Springer, Berlin, 189–202. https://doi.org/10.1007/978-94-017-2921-5_12

[59] Clare Thornley. 2011. Teaching Information Retrieval Through Problem-Based Learning. In *Teaching and Learning in Information Retrieval*. INRE, Vol. 31. Springer, Berlin, 183–198. https://doi.org/10.1007/978-3-642-22511-6_13

[60] Margaret Thornton. 2009. Academic Un-Freedom in the New Knowledge Economy. *Australian National University College of Law Legal Studies Research Paper Series* 10-47 (2009), 19–34. https://ssrn.com/abstract=1599365

[61] Andrew Trotman and Kat Lilly. 2020. JASSjr: The Minimalistic BM25 Search Engine for Teaching and Learning Information Retrieval. In *Proceedings of SIGIR 2020*. ACM, New York, 2185–2188. https://doi.org/10.1145/3397271.3401413

[62] Julián Urbano, Mónica Marrero, Diego Martín, and Jorge Morato. 2011. Bringing Undergraduate Students Closer to a Real-world Information Retrieval Setting: Methodology and Resources. In *Proceedings of ITiCSE 2011*. ACM, New York, 293–297. https://doi.org/10.1145/1999747.1999829

[63] Ellen Voorhees. 2004. Overview of the TREC 2004 Robust Retrieval Track. In *Proceedings of TREC 2004 (NIST Special Publication)*. NIST.

[64] Ellen M. Voorhees. 1996. NIST TREC Disks 4 and 5: Retrieval Test Collections Document Set.

[65] Ellen M. Voorhees. 2001. Philosophy of IR Evaluation. In *Working Notes of CLEF 2001 (CEUR Workshop Proceedings, Vol. 1167)*. CEUR-WS.org. https://ceur-ws.org/Vol-1167/CLEF2001wn-other-Voorhees2001.pdf

[66] Ellen M. Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R. Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. 2020. TREC-COVID: Constructing a Pandemic Information Retrieval Test Collection. *SIGIR Forum* 54, 1 (2020), 1:1–1:12.

[67] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, Kathryn Funk, Rodney Kinney, Ziyang Liu, William Merrill, Paul Mooney, Dewey A. Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex D. Wade, Kuansan Wang, Chris Wilhelm, Boya Xie, Douglas Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. 2020. CORD-19: The Covid-19 Open Research Dataset. arXiv 2004.10706. https://doi.org/10.48550/arXiv.2004.10706

[68] Thomas Wilhelm-Stein, Stefan Kahl, and Maximilian Eibl. 2017. Teaching the Information Retrieval Process Using a Web-Based Environment and Game Mechanics. In *Proceedings of SIGIR 2017*. ACM, New York, 1293–1296. https://doi.org/10.1145/3077136.3084143

[69] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *Proceedings of ICLR 2021*. OpenReview.net. https://openreview.net/forum?id=zeFrfgyZln

[70] Zeynep Akkalyoncu Yilmaz, Charles L. A. Clarke, and Jimmy Lin. 2020. A Lightweight Environment for Learning Experimental IR Research Practices. In *Proceedings of SIGIR 2020*. ACM, New York, 2113–2116. https://doi.org/10.1145/3397271.3401395

[71] Amir Zeldes. 2017. The GUM corpus: Creating Multilayer Resources in the Classroom. *Language Resources and Evaluation* 51, 3 (2017), 581–612. https://doi.org/10.1007/s10579-016-9343-x